SmartX 超融合技术原理与特性解析合集（一）

– 虚拟化与存储

深入解读快照、缓存、I/O 路径、弹性恢复、Vhost、Boost 等关键技术与特性，包含与 VMware 和 Nutanix 的详细对比。

smartx
MAKE IT SIMPLE

关于 SmartX

北京志凌海纳科技有限公司（SmartX） 是专业的现代化 IT 基础设施产品与方案提供商，提供超融合基础设施、 分布式存储、Kubernetes 原生存储等多样化产品组合，助力富士康、交通银行、泰康保险、国泰君安证券、京东方、中山一院等 1000+ 海内外客户构建简单、弹性、可靠、开放的现代化 IT 基础设施。SmartX 是国家级“专精特新”小巨人企业与Gartner 全球全栈超融合软件推荐厂商，在中国超融合软件市场份额统计中位列第一，并连续三年获评 Gartner Peer Insights 亚太区客户之选。

了解更多有关 SmartX 的信息，请访问官方网站。

www.smartx.com

联系销售了解产品与服务，请在工作日 9:00 – 18:00 给我们来电。

400-116-5559

发送邮件向我们咨询产品或市场的更多信息。

info@smartx.com

获取最新技术资讯与行业客户实践，扫码关注微信公众号。

SmartX HCI 是中国独立超融合软件市场份额排名第一的超融合基础设施产品组合。它以弹性、精简的架构一站式提供虚拟化、分布式存储、软件定义网络与安全、容器管理与服务、数据保护与容灾等云基础架构核心组件，关键业务支撑能力经行业头部客户大规模验证，让您以更低的初始投资，逐步、平稳实现云化、国产化替代和容器化转型目标。

SmartX HCI 具备领先的全栈能力、核心自主研发、承载关键业务、方案开放解耦的核心优势。这些优势如何通过具体的技术与特性实现的？能为用户带来哪些好处？与 VMware、Nutanix 等国际厂商对比如何？

为解答以上问题，本文档挑选了 SmartX 超融合所涉及的部分技术原理与特性解析，基于博客内容整理而成，希望能对读者深入了解该产品有所帮助。《虚拟化与存储》部分包含：快照、缓存、I/O 路径、VMware 性能对比、Nutanix 全面对比、弹性恢复、Vhost、Boost、GPU 直通 & vGPU、DRS、网络 I/O 虚拟化、临时副本机制。

目录

快照 | VMware 与 SmartX 快照原理浅析与 I/O 性能对比

点击链接阅读原文：VMware 与 SmartX 快照原理浅析与 I/O 性能对比

要点总结

VMware vSphere 共有 4 种快照模式：VMFSsparse 基础快照、Sesparse、vSANSparse 和 vVols/native snapshots。

除了 vVols/native snapshots，VMware vSphere 其他三种快照的性能取决于多种因素，包括 I/O 类型、数据位置、快照深度、redo-log 大小以及 VMDK 的类型等。

其中 VMFSsparse->SEsparse->vSANSparse 这几种快照的演变实际上是快照性能优化的过程：相比 VMFSsparse，SEsparse 通过 4KB 对齐场景优化降低了写放大和磁盘置零开销，vSANSparse 在此基础上又优化了内存缓存元数据，提高了快照 I/O 性能。但 vSANSparse 依旧无法完全避免快照遍历开销，同时快照深度增大、快照合并/删除还是会带来大幅的性能降低。

不同于基于 redo-log 文件和快照链结构的 VMware 快照技术，SMTX OS 快照拥有独立的元数据，避免遍历快照；快照元数据除了利用内存加速，同时做了持久化存储；使用更大的数据块进行存储，有效规避了多种影响快照性能的因素，降低时延、提升快照性能可恢复性。

同时，SMTX OS 内多组快照之间相互独立，删除快照无需合并操作，更快捷、简单。

相信使用过虚拟机的朋友，对快照功能肯定不陌生。在安装软件、变更系统配置等场景都会使用到快照，它可以帮助我们轻松地将虚拟机恢复到特定时刻的状态。快照功能无疑很方便，但使用 VMware vSphere 执行快照后经常会出现虚拟机性能下降、快照管理复杂等问题，十分影响业务效率。

针对这一现象，本文浅析 VMware vSphere 中快照工作原理，并通过对比 VMware vSphere 和 SMTX OS（SmartX 超融合软件）内的快照机制和实测数据，说明快照执行对虚拟机 I/O 性能的影响。

VMware vSphere 中的快照技术浅析

VMware 对于快照的定义

快照可保存虚拟机在特定时刻的**状态和数据**。

- 状态包括虚拟机的**电源状态**（例如，打开电源、关闭电源、挂起）。
- 数据包括组成**虚拟机的所有文件**。这包括磁盘、内存和其他设备（例如虚拟网卡）。

虚拟机提供了多个用于创建和管理快照及快照链的操作。通过这些操作，用户可以创建快照、还原到链中的任意快照以及移除快照。

快照种类

目前 VMware vSphere 的虚拟机快照共有 4 种模式：

- **VMFSsparse**

VMFSsparse 是 VMware 传统/基础的虚拟机快照模式，其快照运行原理类似 redo-log。在VMFS5 文件系统下，虚拟磁盘默认使用 VMFSsparse 格式（.vmdk 文件小于 2TB）。

- **SEsparse**

SEsparse 运行原理与 VMFSsparse 类似，主要为了改进 VMware Horizon View（虚拟桌面场景）性能推出的快照类型，并且 SEsparse 支持空间回收技术。SEsparse 是 VMFS6 数据存储上所有增量磁盘的默认格式。在 VMFS5 上，SEsparse 用于大小为 2TB 及更大的虚拟磁盘。

- **vSANSparse**

vSANSparse 格式利用新的 VirstoFS 文件系统 (v2) 磁盘格式的底层稀疏性和用于跟踪更新的新内存缓存机制，在保留现有的 redo-log 机制的同时提高了快照性能。vSANSparse 只用于 vSAN 集群，并要求虚

拟机不包含 VMFSsparse 快照。

- **vVols/native snapshots**

这种快照实现并不是由 VMware 层面实现的，而是需要依赖外部存储的快照功能，VMware 通过 VAAI 或者 vVols 将快照操作 Offload 到存储端执行。

本文将针对前三种快照模式进行分析。

快照原理

VMFSsparse 基础快照

VMFSsparse 是在创建虚拟机快照或从虚拟机创建链接克隆时使用的虚拟磁盘格式。VMFSsparse 在 VMFS（VMware 专属的文件系统）之上实现，其本质上是一个重做日志（redo-log）文件，创建快照初时它是空的，当有数据变化就记录到该文件之上，直至文件增长到跟原来的虚拟磁盘一样的大小（当虚拟磁盘上的所有数据都发生了变化）。VMFSsparse 快照实质上是 VMFS 命名空间中的另一个文件，它随着虚拟机快照创建而产生，它与 VM 的虚拟磁盘文件（VMDK）一一对应，并记录虚拟磁盘执行快照后的数据变化。

快照文件组成

- **.vmdk 和 –delta.vmdk**

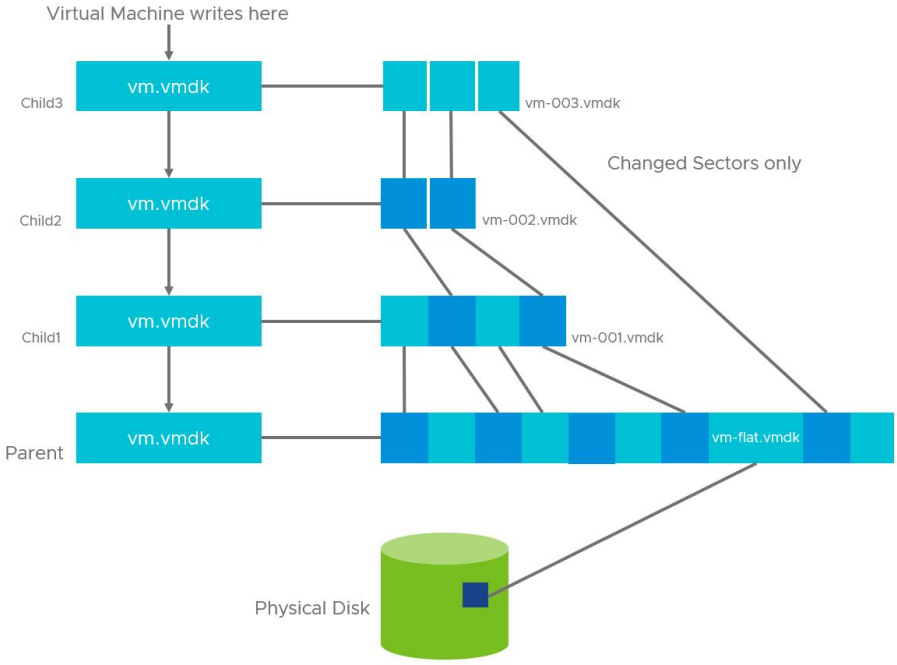
  VMware 虚拟机上的每个虚拟磁盘都是以 .vmdk 命名的，在执行快照后，虚拟磁盘 .vmdk 文件会对应生成 –delta.vmdk 文件。而 .vmdk 和 –delta.vmdk 文件的集合都会连接到虚拟机。–delta.vmdk 文件可称为子磁盘文件。当虚拟机再次执行快照时，这些子磁盘可以被视为未来的子磁盘的父磁盘。在原始父磁盘中，每个子磁盘都将构建一个还原点：提供从虚拟磁盘的当前状态回退到原始状态的服务。

- **.vmsd**

  .vmsd 文件是虚拟机快照信息的数据库（也可以理解为快照的元数据），并且是快照管理器信息的主要来源。该文件包含一些行条目，这些条目定义了快照之间以及每个快照的子磁盘之间的关系。

快照链

如下图，原始虚拟磁盘（parent）在示意图的最下方，它包含未执行快照之前完整的数据块。第一次执行快照后（示意图下方起第二层）生成子磁盘（child1）文件，该快照文件只会记录执行快照后修改过的数据，未被修改过的数据块不会记录在子磁盘文件，而是访问父磁盘对应的数据块，因此它是一个稀疏的磁盘文件。当第二次执行快照时（示意图下方起第三层）生成子磁盘（child2）文件，原理跟首次快照类似，只是 child2 的父磁盘变为 child1，child2 将记录第二次快照后的数据变化，如此类推。



图片来源：了解 vSphere 中的虚拟机快照 (1015180)

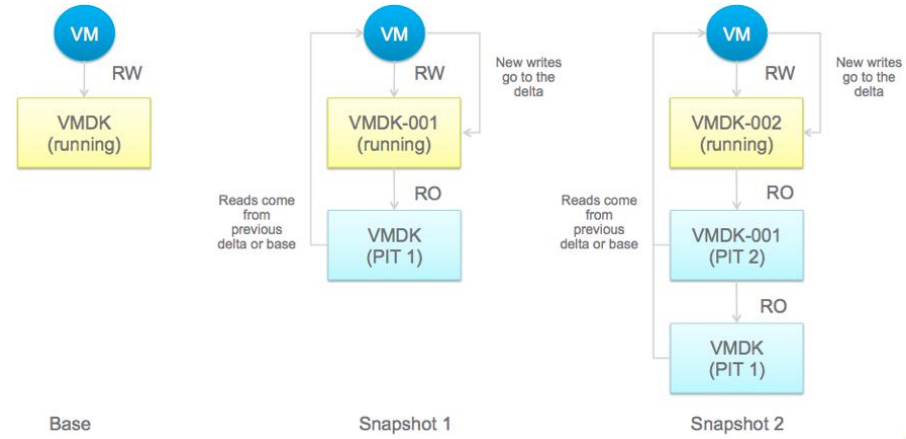
快照 I/O 原理

如前面提到的，VMFSsparse 快照是在 VMFS 文件系统之上实现的，其中快照重做日志（-delta.vmdk 文件）除了记录了已变化的数据，还同时维护自身的元数据，以便实现重做日志上的数据块的寻址。重做日志的块大小是 512 字节（刚好是一个扇区大小），使得其读写粒度可以小到一个扇区。当从一台带快照的虚拟机发出 I/O 时，VMware 需要通过元数据信息确定数据是在基础虚拟磁盘（vmdk）上，还是在快照重做日志（-delta.vmdk）上，使得 I/O 能从正确的位置进行服务。快照的性能取决于多种因素，包括 I/O 类型、数据位置、快照深度、redo-log 大小以及 VMDK 的类型等。

VMFSsparse 快照对 I/O 性能影响

1. I/O 类型

当虚拟机执行快照后，读、写两种 I/O 类型的性能变化是明显不同的：



图片来源：vsanSparse Snapshots

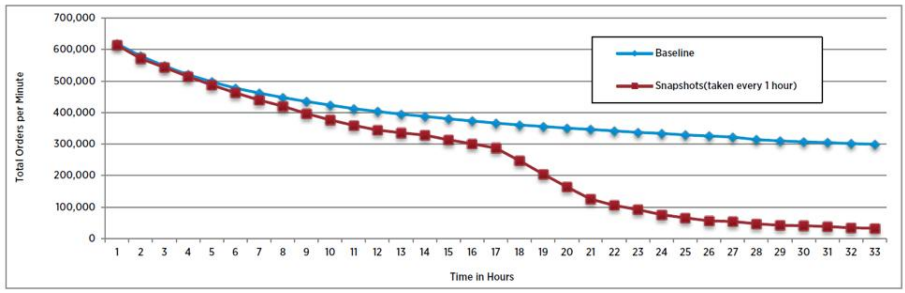
其中，读 I/O 由快照文件和原始磁盘文件共同提供服务；执行快照后修改过的数据将从 redo-log 上读取，未修改过的数据则从原始 VMDK 上读取，这种机制使得部分顺序读取的 I/O 变成随机读取，这种情况对机械磁盘并不友好。

对于写 I/O，如果是快照后首次写入的数据块，它将直接写入 redo-log，并需要同时更新 redo-log 上的元数据以标记该数据块的物理位置；已存在 redo-log 的数据则会直接覆盖。

2. 快照深度

当虚拟机拥有多个快照时，读取数据的时候可能需要遍历每一层快照文件，查询多个快照文件中的元数据，并造成 I/O 性能明显下降。

下图是 VMware vSAN 官方给出的快照深度性能测试示意，可以看到性能随着快照数量增加而递减，执行 32 个快照后性能下降至接近 0，而且性能并不会恢复。



数据来源：VMware Virtual SAN Snapshots in VMware vSphere 6.0

3. VMDK 格式

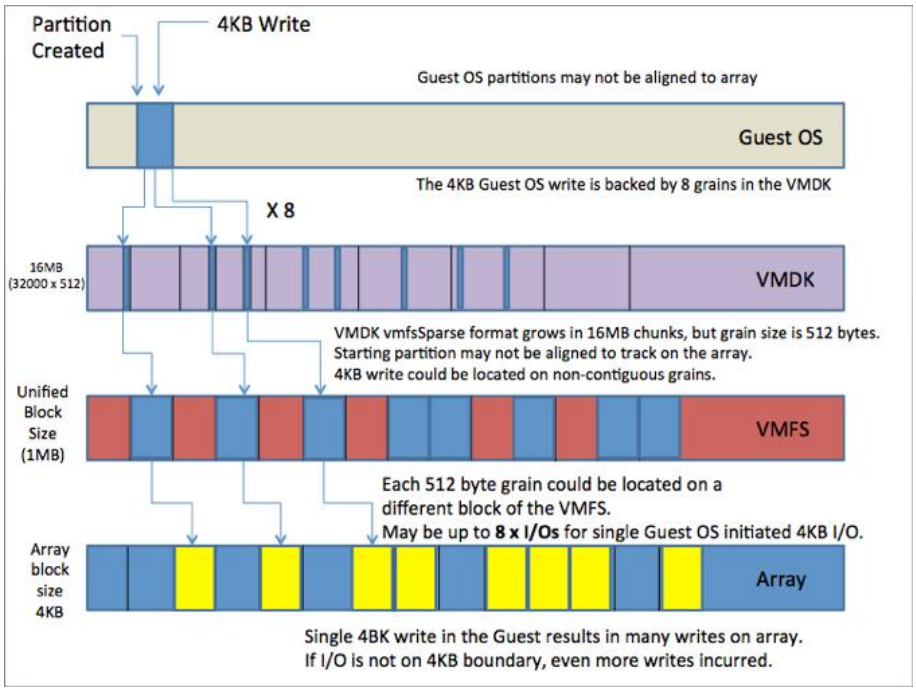
基础虚拟磁盘（.vmdk）格式也会影响 I/O 的性能。在基础虚拟磁盘（.vmdk）的格式为 thin（精简磁盘）且空间未完全分配的情况下，在执行快照后，写入基础精简 VMDK 中的未分配块将导致两个操作：1）对基础 thin 虚拟磁盘（.vmdk）分配空间以及数据块进行置零操作（VMware 避免出现残留数据的机制）；2）将真实数据写入快照文件（-delta.vmdk）。这种场景下 I/O 性能将明显下降。

SEsparse 快照

SEsparse 是一种类似 VMFSsparse（redo-log）的虚拟磁盘格式，并提供一些新功能以及特定场景下的性能优化。SEsparse 与 VMFSsparse 的区别之一是 SEsparse 的块大小为 **4KB**，而 VMFSsparse 的块大小为 **512 字节**。上面讨论的关于 VMFSsparse 的大多数性能影响因素——I/O 类型、快照深度、数据的物理位置、基本 VMDK 类型等也适用于 SEsparse 格式。除了块大小的变化，SEsparse 虚拟磁盘格式的主要变化在于空间效率。SEsparse 虚拟磁盘在 VMTools 的配合下（开启 umap 功能），客户端的文件系统删除数据后，自动通知 SEsparse 删除数据块的映射并回收空间，使得膨胀后的 VMDK 再次收缩，以达到节省存储空间的目标。

4K 对齐改善写放大问题

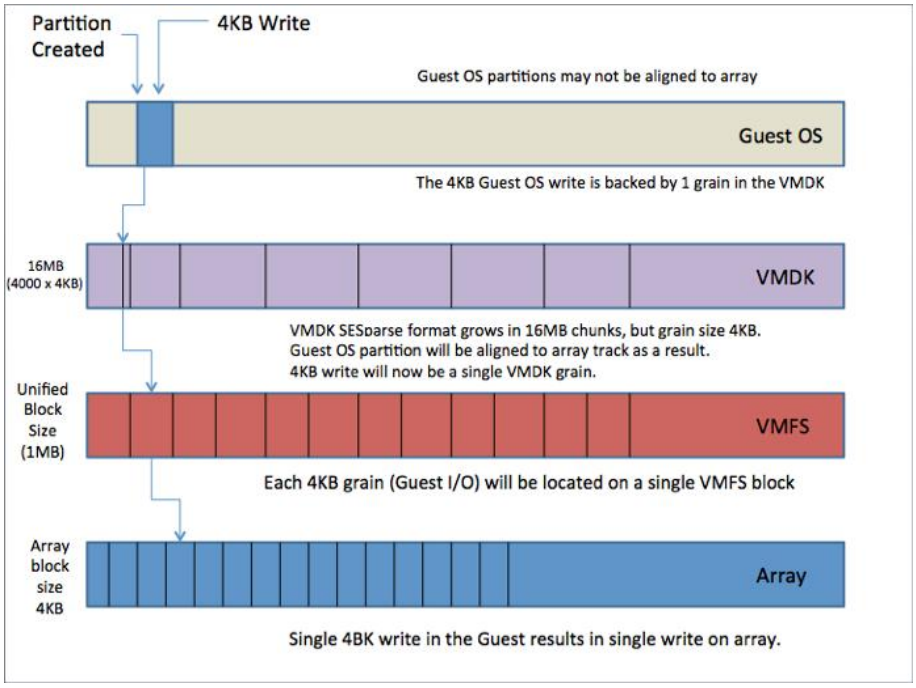
前面提到过 VMFSsparse 的块大小为 512 字节，而实际 I/O 经过多层文件系统后，写操作放大问题是比较显著的。下面以从虚拟机操作系统（Guest OS）发出一个 4KB 的 I/O 作为例子，展示其经过 VMDK、VMFS 以及后端存储的过程中写放大的情况。



图片来源：vsanSparse Snapshots

当虚拟机发出一个 4KB I/O，由于虚拟磁盘（VMFSsparse VMDK）的块大小是 512 字节，那么 4KB I/O 需要被拆成 8 个 512B I/O，写到 VMDK 文件的 8 个不同的数据块当中，因为不对齐的原因，4KB 的数据有可能打散到多个不连续的块当中；而 VMDK 文件又是存放在 VMFS 文件系统之上（VMFS 的块大小是 1MB），这些 VMDK 上的数据块分别映射到 VMFS 上的 8 个不同的数据块当中；而最终 VMFS 的 I/O 会写到存储阵列（或其他外部存储设备），使得 I/O 操作至少放大了 8 倍（仅当外部存储设备块大小为 4KB 时；如果不是 4KB，有可能放大的情况更严重）。

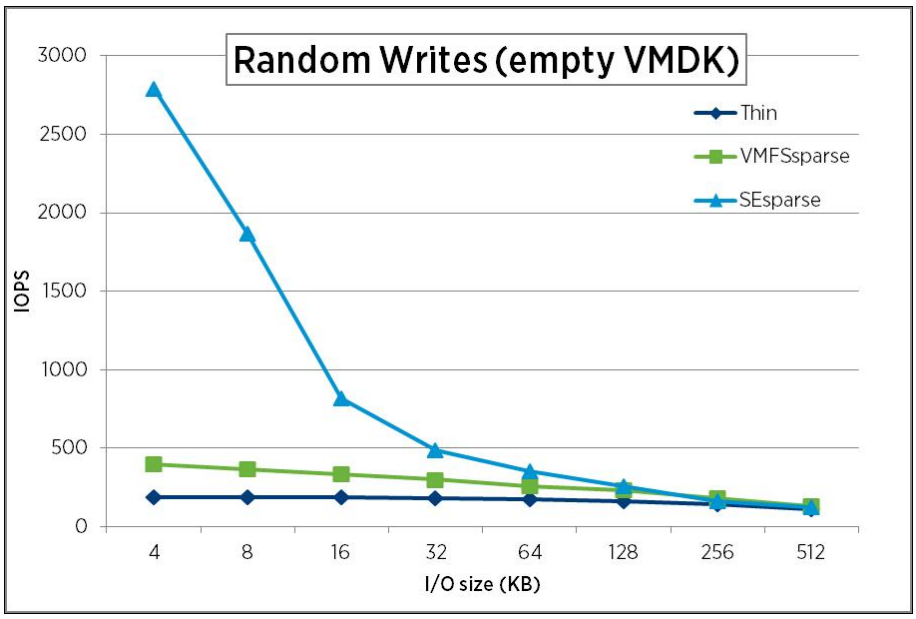
SEsparse 为了改善上述写放大的问题，将块大小调整为 4KB，那么从虚拟机发出的 4KB I/O 将对齐写入单个 VMDK 数据块，由于 VMFS 的块更大（1MB），因此最终也只会写入单个 VMFS 的数据块当中，最后写入外部存储设备时，只需要一次 I/O 操作就能完成（4KB 对齐），避免了写放大的情况。



图片来源：vsanSparse Snapshots

4KB 对齐的优化效果

为证明 SEsparse 对于减少写放大的效果，针对三组对象执行快照，并使用 IOMeter 执行不同 I/O 块大小的测试：



图片来源：SEsparse in VMware vSphere 5.5

Thin：原 VMDK 设置为精简置备。 VMFSsparse：原 VMDK 设置为厚置备置零。 SEsparse：原 VMDK 设置为厚置备置零。

从测试结果上可以看到：精简置备（Thin）随机写入性能在所有测试场景都是最低的，主要原因在于，精简置备场景下，需要首先将块置零，然后再写入实际数据。这是因为 VMFS 以 1MB 的粒度分配块，而该区域的一部分可能会被真实数据填充。置零可防止应用程序从分配的 1MB 物理介质中读取了残留数据。相反，当使用 SEsparse 和 VMFSsparse 格式时，空间分配发生在更小的块大小中，分别为 4KB 和 512 字节，因此当 I/O 大于或等于 4KB 并且是 4KB 对齐的，则无需将块置零（对于非对齐情况，需要执行“读–修改–写”操作），避免了置零的性能开销。

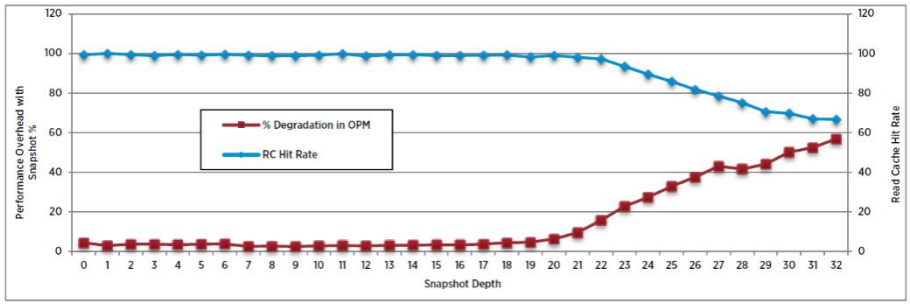
在随机写入测试中，SEsparse 的性能也明显优于 VMFSsparse 格式。这是因为 SEsparse 实现了智能 I/O 合并逻辑，避免写放大以获得更好的性能。（需要注意的一点：SEsparse 仅在 I/O 与 4KB 边界对齐的情况下执行，能获得与 VMFSsparse 相当或更好的性能。这是因为当 I/O 没有 4KB 对齐，写入操作可能会导致“读取–修改–写入”多次 I/O 操作，从而增加开销。但现实中几乎所有文件系统和应用程序都是 4KB 对齐的，因此 SEsparse 在常见场景中表现要比 VMFSsparse 更好。

vSANSparse 快照

vSANSparse 是在 vSAN 6.0 中引入的一种新的快照格式，它利用内存缓存的快照的元数据提升性能；与 VMFSsparse 和 SEsparse 相比，vSANSparse 在多数情况下性能更好。

当读 I/O 请求到达 vSAN 时，vSANSparse 快照逻辑会遍历该虚拟机的快照树的各个级别，并自动组合 I/O 请求相关的 vSAN 对象和偏移量。然后，这个寻址信息会缓存在 vSAN 快照元数据缓存中（内存中）。快照元数据缓存在于内存中，对快照的读性能至关重要。因为一旦快照元数据缓存未命中，就必须通过遍历多级快照来获取地址信息，这将大幅增加 I/O 访问延迟（这与原来的 VMFSsparse 和 SEsparse 快照是类似的）。元数据缓存的大小是有限制的，并且缓存空间是 VMware 系统中所有打开虚拟机的全部 VMDK 之间共享。因此，当缓存已满时，会淘汰一部分已有的缓存信息。

下图是关于快照缓存命中率与快照性能下降比例的对照图：



数据来源：VMware Virtual SAN Snapshots in VMware vSphere 6.0

从测试结果观察到，当 vSAN 快照数量低于 19 个时，快照缓存的命中率维持在 98% 以上，这个时候快照的性能损失低于 5%，证明快照缓存空间充足的时候，vSANSparse 对于读操作的优化十分明显。但随着快照深度增大，缓存命中率进一步降低，到 32 个快照的时候，性能下降比例增至 56%。另外由于快照元数据位于内存当中，一旦主机重启，缓存会被清空，含有快照的虚拟机性能将明显下降。

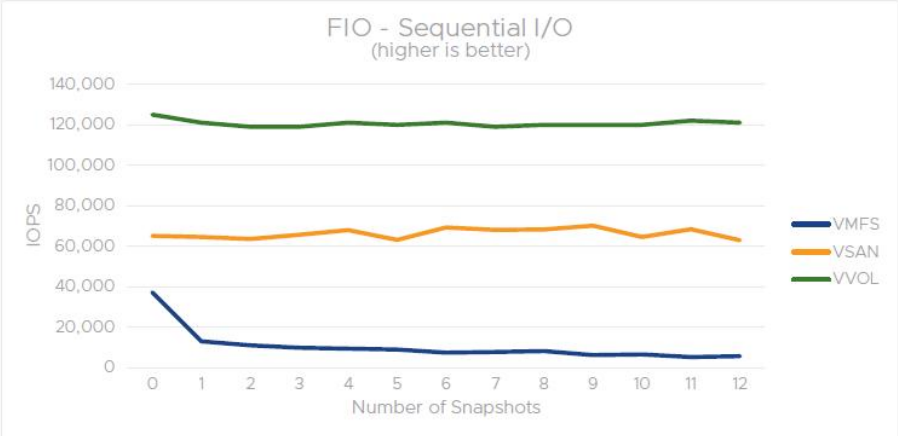
当 VMDK 只包含一个快照的时候，VMFSsparse 与 vSANSparse 混合读写的性能对比如下：

性能下降因素	VMFSsparse	vSANSparse
4K FIO (顺序 I/O)	性能下降 65%	几乎没有影响
FIO (随机 I/O)	性能下降 65%	性能下降 35%
FIO (512k 顺序 I/O)	性能下降 70%	几乎没有影响

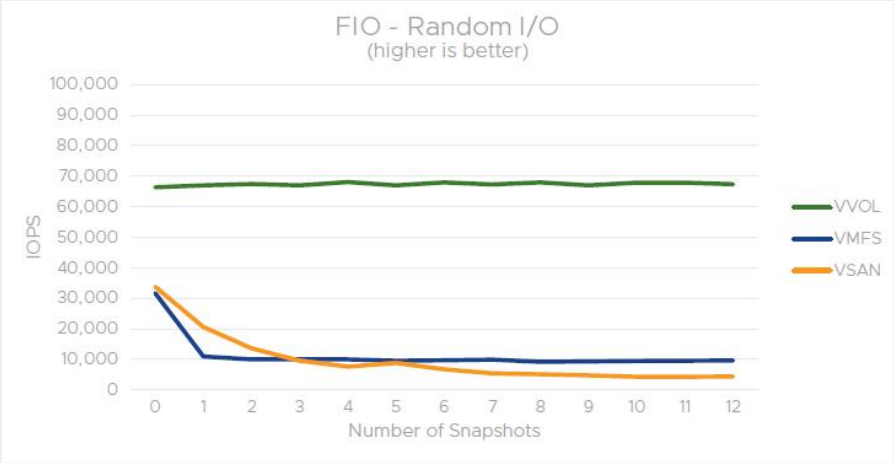
数据来源：VMware vSphere Snapshots: Performance and Best Practices

可以看到 vSANSparse 在快照深度等于 1 的场景下，其性能优化效果是比较明显的。

以下是 vSANSparse 混合读写在不同的快照深度下性能测试结果：



4KB 顺序混合读写（50% 读，50% 写）测试



4KB 随机混合读写（50% 读，50% 写）测试

数据来源：VMware vSphere Snapshots: Performance and Best Practices

从测试结果中观察到 vSANSparse 快照对顺序读写 I/O 的工作负载的性能影响比较小。而在随机读写 I/O 测试的场景下，结果与 VMFSsparse 是类似的，性能有较大幅度的下降。可以了解到当快照深度加大，vSANSparse 快照对于随机读写的优化效果并不明显。

VMware 快照的演进情况汇总

性能下降因素	VMFSsparse	SEsparse	vSANSparse
快照遍历开销	未解决	未解决	内存缓存元数据优化，但当缓存不足或缓存清空，依然存在遍历问题
写放大开销	未解决	4KB 对齐场景优化	4KB 对齐场景优化
磁盘置零开销	未解决	4KB 对齐场景优化	4KB 对齐场景优化

vSANSparse 快照存在的性能问题：

- **随着快照数量和深度增加，元数据缓存无法避免快照的性能下降。**
- **主机重启后，元数据缓存无法自动加载，快照遍历的情况依然存在，性能下降明显。**
- **快照链结构导致删除快照时，可能需要进行多次快照合并操作，带来较大的性能损耗。**

SMTX OS 中的快照技术浅析

SMTX OS 对于快照的定义

虚拟机快照可保存其特定时刻数据和配置信息。且虚拟机和虚拟机快照是独立的存在，它们并不互相依赖。

虚拟机快照包含下面信息：

- 虚拟机包含所有虚拟卷的快照（共享虚拟卷除外）。
- 虚拟机的配置信息，例如 vCPU 数量、内存大小、磁盘启动顺序、网络配置等。

SMTX OS 中虚拟机快照支持崩溃一致性快照[1]以及文件系统一致性快照[2]，但目前并不支持内存快照[3]。此外，虚拟机快照支持诸如：创建快照、还原任意快照以及单独移除任意时刻快照等快照管理操作。（VMware vSphere 中虚拟机快照无法单独删除位于快照链条中间的快照，必需完成此快照之后的多个快

照合并操作，才能实现快照的删除）；SMTX OS 中虚拟机快照还可以支持通过某个时刻的快照实现重建（克隆）虚拟机的操作，这些都属于 SMTX OS 的虚拟机快照的特点之一。

快照原理

SMTX OS 的虚拟机快照与 VMware vSphere 的快照的运行原理并不相同。SMTX OS 的快照不是基于 redo-log 文件实现的，因此也不存在快照链条的结构，且多个快照之间没有依赖关系。

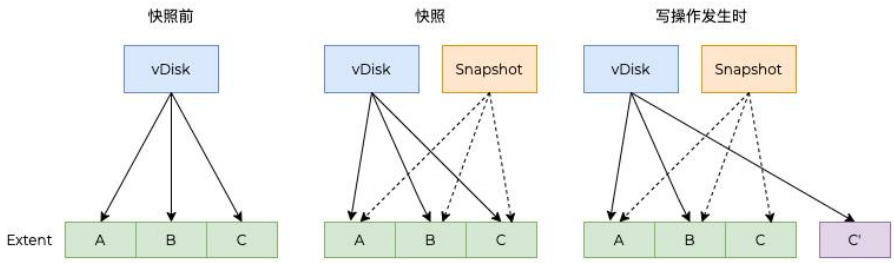
快照组成

由于 SMTX OS 的虚拟机快照并没有类似 VMFS 文件系统或者 VMDK 文件这一层，它的组成更加简单。它主要由两部分组成：

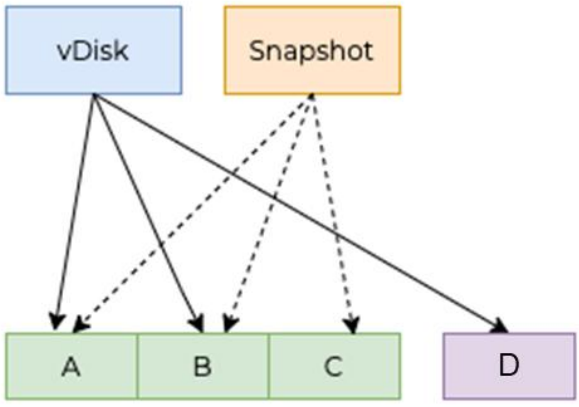
- **元数据**
  虚拟磁盘（vDisk）和快照都有类似结构的元数据信息，而且同一个 vDisk 的多个快照分别拥有独立的元数据信息（包含快照相关的所有数据块的物理位置信息），该信息记录在 zbs-meta。
- **数据块（extent）**
  快照的真实数据存储在数据块（extent，每个 extent 块大小为 256MB），快照数据一般由多个 extent 组成，快照后发生变化的数据将存放在独立的数据块（extent）中，而没有被修改的 extent 是快照和原 vDisk 共享的。

快照 I/O 原理

vDisk 拥有自身的元数据信息并记录了原始的数据块（extent）映射关系，当执行快照后，系统将生成独立的快照的元数据信息并记录快照相关的数据块（extent）映射关系（如下图）。当数据未发生任何变化时，快照与 vDisk 对应的 extent 是完全一致的，也就是快照与 vDisk 共享所有数据块，因此，这个时候快照并不额外占据任何存储空间。

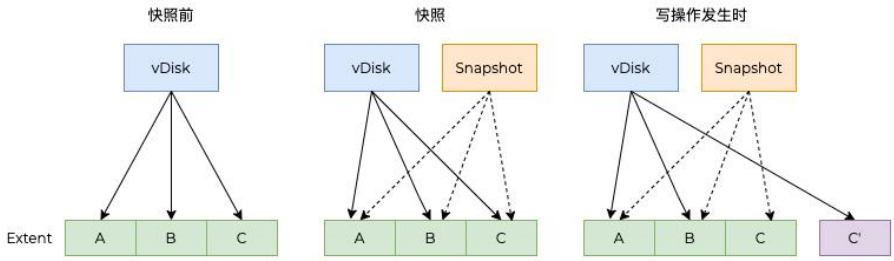


写 I/O 发生时，如果是快照后首次写入的数据块（该 extent 未被分配），它将被分配新数据块（new extent），并将数据直接写入新分配的 extent 中，并更新 vDisk 的元数据信息，将其映射关系指向新分配 extent。

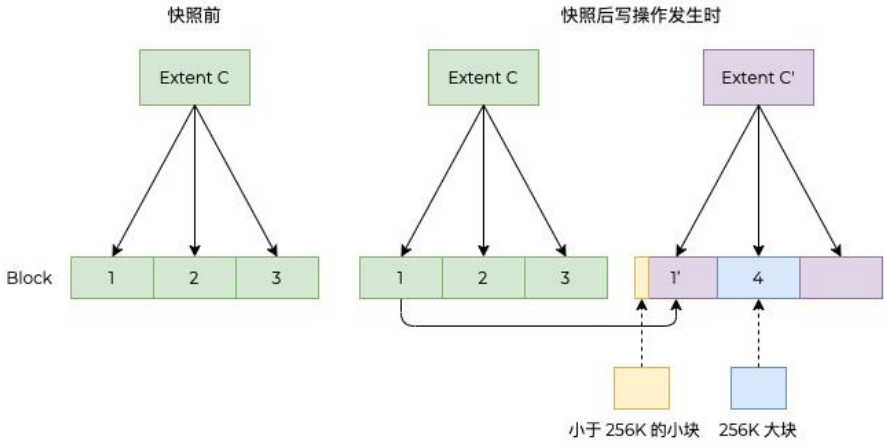


如果写入的数据块（extent）在原 vDisk 已经被分配，同样地，系统也将新分配数据块（extent C'与被修

改的 extent C 对应），并更新 vDisk 的元数据信息，将其映射关系指向新分配 extent C’。由于原 vDisk 的 extent 上已经有数据，写入操作可能会导致“读取–修改–写入”多次 I/O 操作。



其中 extent 块大小是 256MB，而 block 是 extent 下面更小的数据块单位，block 的大小为 256KB，每个 extent 包含了 1024 个 block。



当写入 I/O 小于 256KB，例如需要写入 4KB 数据，那么需先从原 extent C 上读取对应 block 的数据，修改数据后，将数据最终写入新创建 extent C’上对应的 block。当对齐写入 256K I/O 时，则无需读取原 vDisk 上 block，直接写入新位置。

读 I/O 发生时，底层存储快照数据块和 vDisk 数据块共同提供服务；由于 vDisk 的元数据映射关系已经被更新，它包含当状态所有数据块的最新映射关系，因此读取访问是无需遍历快照的，读取寻址的延时是比较低的。

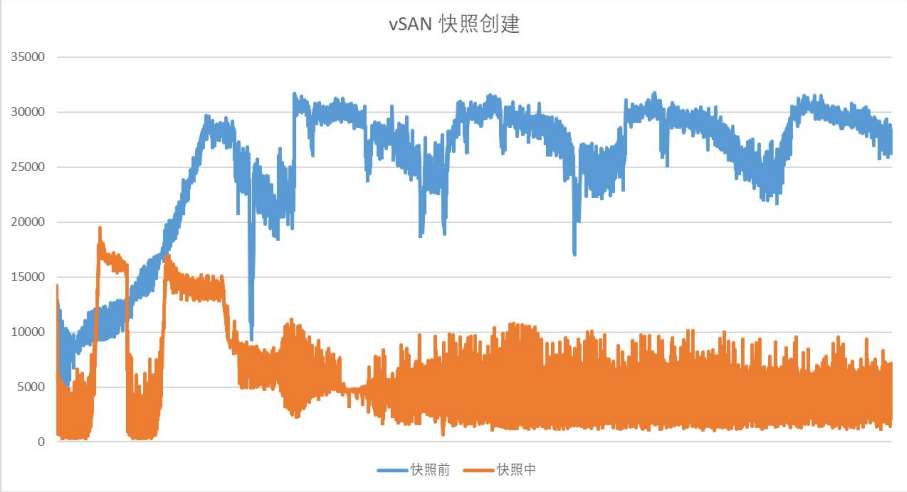
SMTX OS 快照的元数据是存储在 ZBS 分布式存储元数据服务集群内，元数据位于内存中有更好的响应速度，同时元数据也会持久化同步到 SSD 介质上，这样即使是主机重启后，也可以通过 SSD 快速加载元数据到内存当中，不会因为主机重启而降低快照性能。

SMTX OS 的虚拟机快照与 vSANSparse 的优劣对比

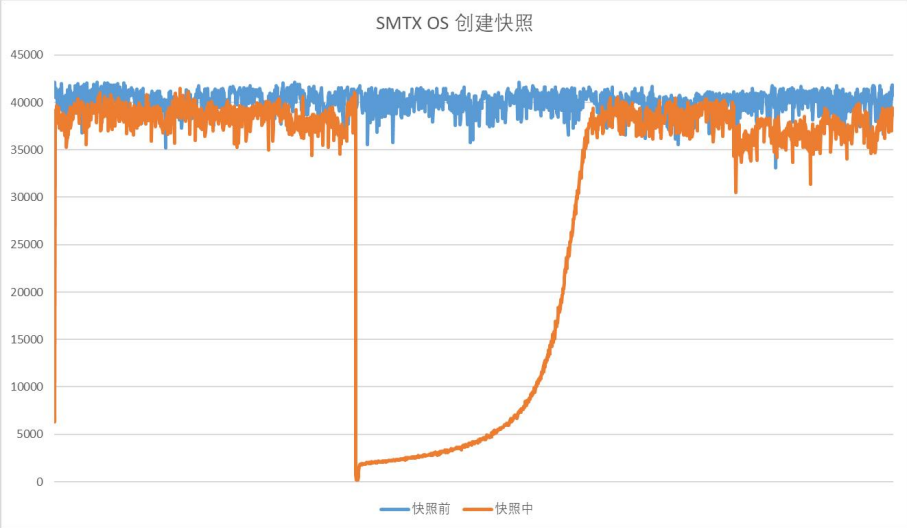
对比项目	SMTX OS 的虚拟机快照	vSANSparse 快照
I/O 对齐块大小	256 KB	4 KB
快照空间增长	较快	较慢
多个快照之间的关系	快照互相独立	形成快照链，快照间互相关联
创建快照的性能影响	性能影响时间短，可恢复	性能影响是持续的，无法恢复
元数据缓存加速	持久化，主机重启自动加载	非持久化， 主机重启后缓存消失
删除快照的性能影响	只涉及元数据操作，无影响	需要合并快照，性能下降明显

实测对比快照性能

在同一硬件配置场景下，分别测试 SMTX OS 和 vSAN 在快照前后的性能表现。



在 4k 随机写测试中，可以看到 vSAN 创建快照后，虚拟机性能下降接近 60%，并无法恢复。



在同样的 4k 随机写测试中，可以看到 SMTX OS 创建快照后，虚拟机在短时间内有明显的性能下降，但可逐步恢复到快照前的性能水平（恢复时间约为 20 分钟）。

总结

通过对 VMware vSphere 和 SMTX OS 实现快照的原理的解读以及实测快照性能的对比验证，可以看出，SMTX OS 的快照技术规避了 I/O 类型、VMDK 格式等会对 I/O 性能产生影响的因素，显著改善了快照运行后 I/O 性能下降的时间和可恢复性；同时，SMTX OS 不存在快照链条的结构，能够保证多个快照间的独立性，从而方便运维人员对快照进行删除等操作和管理。

另外，当 SMTX OS 与 VMware vSphere 集成部署时，可支持文章前面提到过的 vVols/native snapshots 快照模式，使得 vSphere 虚拟机可以通过专用的 VAAI 插件将快照操作 Offload 到 SMTX OS 中完成，获得 SMTX OS 的快照特性。

崩溃一致性快照[1]：崩溃一致性快照仅记录已写入虚拟硬盘的数据。快照中不会捕获内存或待处理 I/O 操作中的任何数据。因此，此类型的快照无法保证文件系统或应用程序的一致性，您可能无法还原具有崩溃一致性快照的虚拟机。

文件系统一致性快照[2]：除了虚拟硬盘上的数据之外，文件系统一致性快照还会记录内存和待处理 I/O 操作中的所有数据。在拍摄文件系统一致性快照之前，访客操作系统上的文件系统会进入静默状态，内存中的所有文件系统缓存数据和待处理I/O 操作都会刷新到硬盘。

内存快照[3]：内存快照是指虚拟机执行快照时，除了对硬盘数据执行快照之外，虚拟机会进入静默状态，内存也会同时执行快照，并持久化保存内存数据；当执行虚拟机快照恢复时，可加载内存快照数据。

缓存 | VMware 与 SmartX 分布式存储缓存机制浅析与性能对比

点击链接阅读原文：VMware 与 SmartX 分布式存储缓存机制浅析与性能对比

要点总结

vSAN 7 采用划分读写缓存空间的机制，将缓存磁盘按照容量占比划分为写缓冲区（30%）和读缓存区（70%）。这种方式可能出现缓存利用率低、在访问数据量过大时导致缓存击穿，进而引起性能下降等问题。

ZBS 采用统一缓存空间的机制，并通过 2 级 LRU 算法对冷热数据进行管理，在充分利用缓存容量的同时避免了因访问量激增导致虚拟机性能下降的情况。

本文基于相同的硬件配置和 I/O 读写场景，分别测试 VMware 超融合（vSphere 虚拟化 + vSAN 分布式存储）写入 300 GB 数据、SMTX OS（ELF + ZBS）写入 500 GB 数据时虚拟机的性能表现。结果显示，vSAN 7 难以充分利用缓存介质，发生缓存击穿，导致存储性能下降；而 SMTX OS 即便在写入更多数据的情况下也未发生缓存击穿，虚拟机性能保持稳定。

VMware 发布的 vSAN 8 对存储架构进行了重大更新。其中最主要的变化，即引入了新的 Express Storage Architecture（ESA）架构：用“存储池”替代了原存储架构（OSA）中的“磁盘组”，并不再需要专用 SSD 承担缓存加速功能，一定程度上避免了 8.0 之前版本中的专用缓存盘利用率低、易发生缓存击穿等问题。

而值得一提的是，在 vSAN 大版本更新之前，SmartX 即通过**统一缓存空间**和**智能冷热数据管理**优化了分布式存储缓存机制，有效规避了上述问题。本文将通过重点解读 vSAN（以 vSAN 7 为例）和 SmartX 分布式块存储组件 ZBS* 缓存机制的原理，并测试对比两种缓存机制下虚拟机性能表现，让读者更好地了解两种技术实现机制的区别对业务可能带来的实际影响。

** ZBS 内置于 SmartX 超融合软件 SMTX OS，可与 SmartX 原生虚拟化 ELF 搭配提供服务。*

场景问题

混闪配置是超融合或分布式存储现阶段的主流落地模式。混闪配置是指机器中的磁盘使用 SSD + HDD 混合组成，其中 SSD 磁盘作为数据缓存层，而 HDD 磁盘作为数据容量层。以该模式构建的分布式存储池通过软件算法进行冷热数据自动判断，在提供高性能的同时，还可获得较大的存储容量，进而提升资源利用率，获得相对全闪存储更高的性价比。

在将 SSD 磁盘用作数据缓存层时，部分超融合产品会将缓存容量（Cache）划分为读和写各自独立的两部分。例如，vSAN 7（以下简称“vSAN”）及更早版本会将每个磁盘组（Disk Group）中的缓存磁盘，按照容量占比划分为写缓冲区（30%）和读缓存区（70%），当读取数据未命中缓存或者写缓存已满，将会直接从容量层进行读写1。

这种划分读写的方式，虽然可以保障读写 I/O 的缓存击穿空间隔离，但经常导致无法充分利用高速存储介质的缓存空间。例如，在业务虚拟机写数据较多、读数据较少的场景，可能作为写入数据的缓存容量已经被占满，但是读缓存空间还有很多容量没有被使用，反之亦然。

以医疗客户的集成平台建设为例，集成平台通过将各个系统产生的数据集中存储并重新组织，形成医院的数据仓库，帮助医院挖掘数据价值、形成智能化决策，进而加快数字化转型。集成平台通过 ETL 工具，从现有医疗业务系统（如 HIS、EMR 和 LIS 等）的数据库直接抽取数据并进行转换、加载。该过程都发生在中间数据库中，最大程度降低对生产数据库的影响。此时中间数据库会有大量的数据进行写入，使得

缓存空间容易被填满，而如果读写缓存采用固定容量分配，就可能会发生写入数据量 > 写缓存空间容量，进而导致缓存击穿、业务访问性能下降。大型三甲医院集成平台平均每天需要处理 900 万条消息，要求峰值处理能力需达到 1000 TPS，存储性能不足易导致整个业务系统卡顿，严重情况下甚至会宕机，因此非常考验基础架构 I/O 吞吐能力。

针对这一问题，ZBS 使用统一的缓存空间，不划分读写，所有缓存层容量均被使用，不易出现缓存空间不足从而影响存储性能的情况。同时，通过冷热数据分层技术，依据数据的访问频率，将频繁读写的热数据放置在 SSD 中、长时间无读写的冷数据放置在 HDD 中，有效提升数据缓存层利用率，保证业务高性能稳定运行。

为了让读者更直观地感受到不同的缓存机制对性能的影响，本文将分别介绍 VMware 和 SmartX 分布式存储缓存机制的原理，并测试对比数据写入场景中两种缓存机制下虚拟机性能表现。

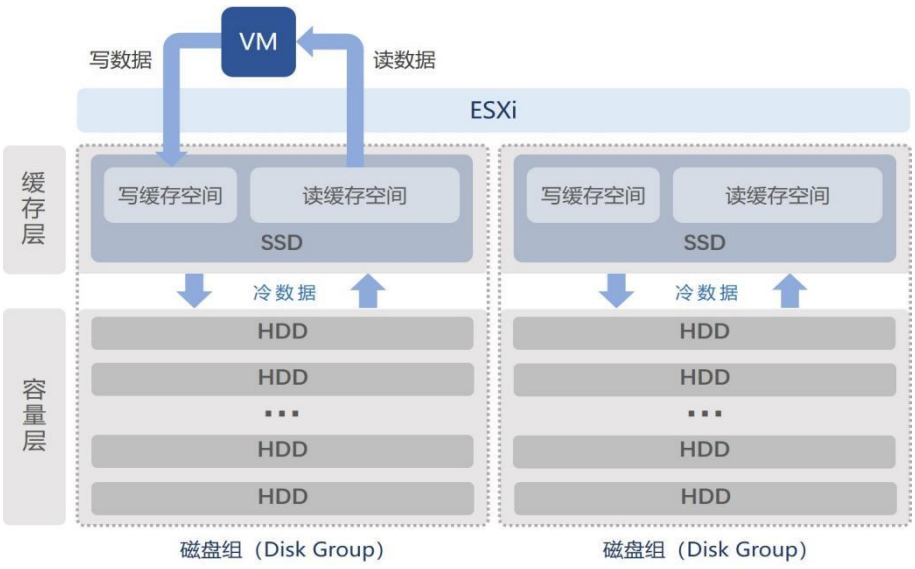
技术实现

vSAN

vSAN 7 使用磁盘组（Disk Group）将高性能存储介质（如 NVMe / SATA SSD）与低性能存储介质（如 SATA / SAS HDD）组成逻辑存储空间，并通过网络 RAID 功能保障数据可靠性。

每台 ESXi 主机可创建 5 个磁盘组，每个磁盘组中至多仅支持 1 块高性能存储介质与 1 ~ 7 块低性能存储介质组合构成。其中高性能存储介质作为缓存层，并以 7 : 3 容量比例划分为读和写的空间使用，虚拟机在进行数据 I/O 访问时，可使用到其中单个磁盘组的缓存空间。低性能存储介质作为容量层，保存因访问频率较低从缓存层回写（Write Back）的冷数据。

当数据写入写缓存空间后，会基于电梯算法（Elevator Algorithm），周期性地将数据回写到容量层，从而保障缓存层有充足的容量支持后续 I/O 的数据请求。当写缓存空间不足时，会直接写入到容量层。冷数据被读取访问时，会将其加载进入读缓存空间中，缓存空间不足时将直接访问持久化缓存层。
另外，vSAN 7 的缓存数据仅能用于本磁盘组，且缓存数据没有冗余。



ZBS

ZBS 在以混闪模式部署的时候会要求选择至少 2 块 SSD 作为缓存容量（Cache），并通过 2 级 LRU

（Least Recently Used）算法进行判定、管理数据冷热程度。

在 Cache 中，数据会被划分为 4 种状态，分别是 Active、Inactive、Clean 和 Free。

- Active：用来记录访问最频繁和“冷转热”的数据，是 Cache 中“最热”的数据。
- Inactive：用来记录首次写入和短时间未访问的数据，是 Cache 中“次热级”的数据。
- Clean：用来记录长时间未访问的数据，是 Cache 中的“冷”数据，且数据已经完成了 HDD 落盘。
- Free：用来记录未使用或被回收的数据，是 Cache 中闲置的数据空间。

通过 2 级 LRU 链表来管理数据冷热程度

首次进行数据写入时，由于 Cache 中没有数据，会从 Free 中请求未使用的数据空间，数据写入完成后将被记录为 Inactive。

Active 中记录了被访问过多次的数据块，当 Active 的数据超过 Inactive 后，将 Active 数据按照被访问的先后顺序进行排序，不经常被访问的数据会转移记录到 Inactive 中，直到两者的容量相同。

同时当 Active 和 Inactive 的数据容量超过 Cache 空间的 20% 之后， Inactive 数据将开始触发落盘操作，按照时间先后顺序进行排序，排名靠前的早期数据会被写入到 HDD 并在缓存层中被标记为 Clean（此时 Clean 数据在 HDD 和 Cache 中各有一份），后续当 Clean 数据有访问请求时会被重新标记为 Active，否则将被回收为可用空间供写入新数据使用。

不同数据读写场景的处理机制

数据写入场景

1. Cache 命中

a. 写入 Cache 空间并根据当前数据状态发生“冷热”变化。

2. Cache 未命中

a. Cache 未满，写入 Cache 中的闲置空间。

b. Cache 已满，写入容量层中。

数据读取场景

1. Cache 命中

系统收到读请求，通过元数据优先查找本地节点 Cache 是否命中请求数据，如果数据在 Cache（Active、Inactive、Clean）中，数据将直接从 Cache 中读取。

2. Cache 未命中

a. Cache 未满

系统会查询本地 Data（容量层）中是否有请求数据，如果有则转向本地容量层请求数据的副本。向本地容量层请求数据的副本，并不是直接读取容量层上的数据，而是首先查询本地 Cache 是否有富余空间，如果 Cache 空间充足，请求数据副本会先从本地容量层中载入到 Cache 中，然后通过 Cache 读取数据，并返回数据，读取成功。

b. Cache 已满

系统收到读请求，当发现请求数据副本没有在本地 Cache，并且这个时候，本地 Cache 空间已经没有富余空间，请求数据副本会直接从容量层中读取。

测试性能验证

测试环境

硬件环境

对比测试时采用相同的硬件环境，均使用 3 台 Dell PowerEdge C6420 服务器 ，每台硬件配置如下：

部件	型号	数量
CPU	Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz	2
MEM	16GB DDR-4	8
SSD	Intel D3-S4610 960GB	2
HDD	TOSHIBA 2.4TB SAS HDD	4

软件版本

软件	版本	用途
SMTX OS	5.0.3	SmartX 超融合软件
vSphere	7.0u1	VMware 虚拟化平台
vSAN	7.0u1	VMware 分布式存储软件
FIO	2.15	通用性能测试工具

测试方法

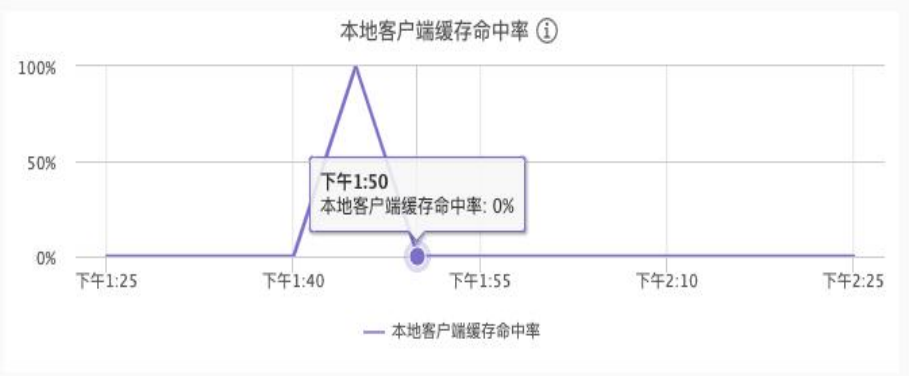
通过 FIO 测试工具，分别在 vSAN 中写入 300 GB 数据、在 SMTX OS 中写入 500 GB 数据，观察虚拟机性能监控视图的数据表现和变化。

测试结果

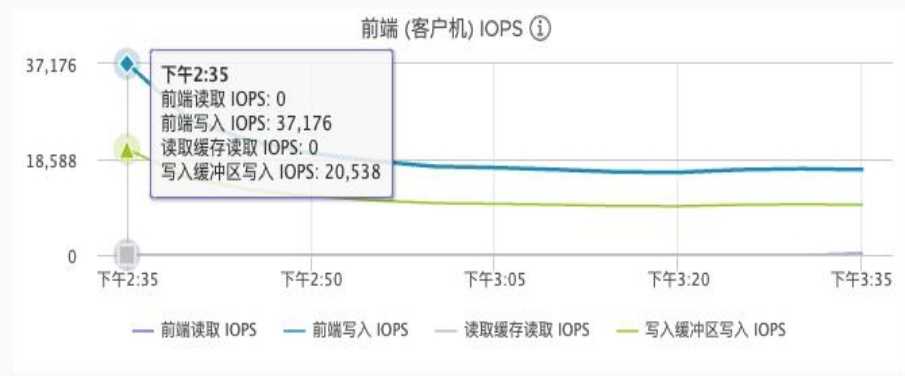
vSAN

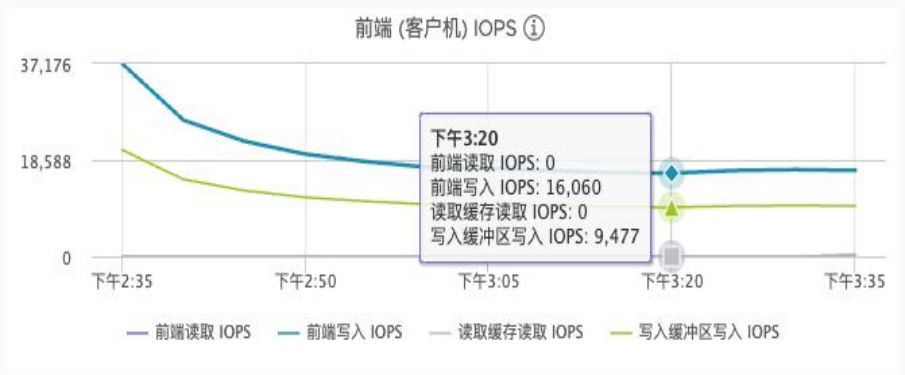
SSD 单盘容量 894.25 GB，读缓存 620.61 GB，写缓存 268.28 GB，读写缓存容量比例为 7:3，符合《VMware vSAN Design Guide》中的描述。

在 256K 顺序写 300 GB 数据测试中，由于缓存空间不足，发生写缓存击穿，性能发生下降，从 280 MB/s 降低到 75 MB/s 左右。



在 4K 随机写 300 GB 测试中，同样由于缓存击穿导致了很大的性能下降，IOPS 从 37176 跌落至 16060。

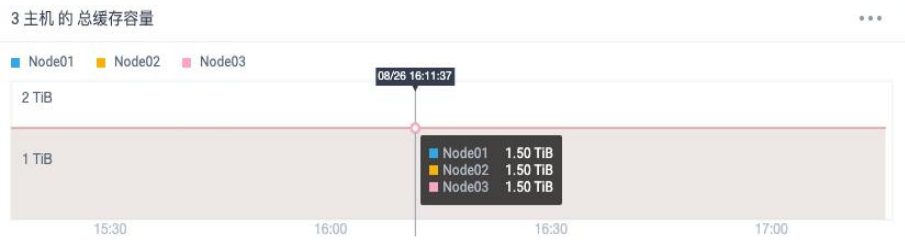




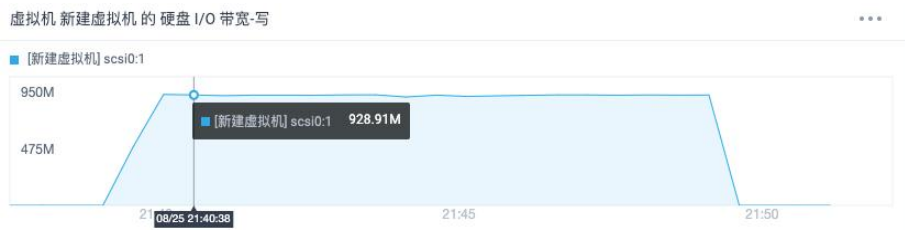
通过上面的测试可以发现，vSAN 采用读写缓存空间各自独立，在容量较大的数据请求场景中存在缓存介质无法充分利用的情况，容易发生缓存击穿导致存储性能下降。

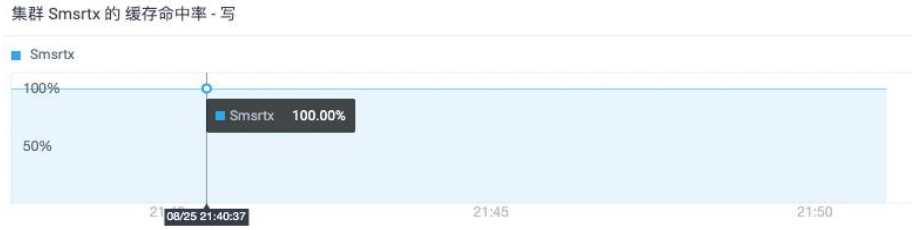
ZBS

从缓存容量监控视图可以看出每个节点均有 1.5 TiB 的缓存容量可使用（不区分读写缓存），即 2 块 ~900GB SSD 的可用缓存容量。



首先使用 FIO 在测试虚拟机中 256K 顺序写入 500 GB 数据。通过虚拟机性能监控视图可以发现，FIO 测试虚拟机无性能波动，一直处于缓存 100% 命中状态。此时首次写入的 500 GB 数据保存在 Inactive。





从上述对比可以发现，在相同的数据容量和 I/O 读写场景中，ZBS 的统一缓存机制能够更好的利用缓存容量空间，即使发生大容量的数据突发请求也不易发生缓存空间击穿导致虚拟机性能下降，进而更好的保障业务稳定运行。

总结

与 vSAN 7 划分读写的缓存机制相比而言，ZBS 在处理数据请求时，通过统一缓存空间的方式，提升了缓存利用率，即使面对业务突发性数据写入和访问激增场景，也能很好地保障业务性能需求。同时，ZBS 采用 2 级 LRU 算法将数据划分为多种热度级别，可对冷热数据进行生命周期管理，缓存层也支持使用多种存储介质（SATA SSD、NVMe SSD 等），用户可根据不同业务的性能需求灵活选择，节省硬件投入成本，获得更高的性价比。

1 对于全闪配置的磁盘组，可以将全部缓存盘容量（100%）用于写缓存，不再设置读缓存区。

I/O 路径丨浅析 VMware 与 SmartX 超融合 I/O 路径差异及其影响

点击链接阅读原文：浅析 VMware 与 SmartX 超融合 I/O 路径差异及其影响

要点总结

基于 SDS 的超融合架构中，I/O 既可能发生在主机内部的本地磁盘上，也有可能发生在外部的网络主机上。相同软硬件配置下，本地 I/O 操作时延更短。

在写入场景下，vSAN 至少有 1 个副本需要经过网络写入。在读取场景下，读 I/O 路径在正常状态下不会出现 100% 本地读取情况；在故障状态下仅有 25% 概率出现 100% 本地读取，其余情况均为 100% 远程读取，且易因故障恢复导致性能瓶颈。

在写入场景下，ZBS* 在正常状态下不会发生 100% 远程写入的情况，并针对虚拟机迁移后的 I/O 路径进行了优化，最大限度避免 100% 远程写入的情况。在读取场景下，正常状态时可确保 100% 本地读；数据恢复时优先进行本地恢复，并通过弹性副本恢复策略平衡恢复速度与业务 I/O。

在正常和数据恢复状态下，ZBS 的本地 I/O 访问概率比 vSAN 更高，理论上时延会更低；而在需要频繁迁移的场景下，vSAN 本地 I/O 访问概率更高。即使超融合的整体 I/O 性能有富余，更多的远程 I/O 也可能引起网络资源争抢等问题。

*ZBS 是 SmartX 超融合软件 SMTX OS 中与 vSAN 对应的分布式块存储组件。

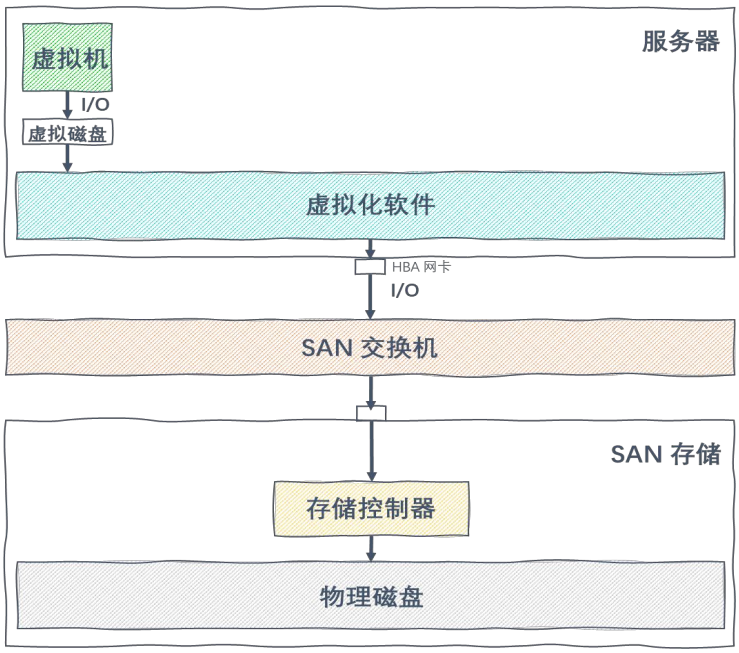
不同的超融合软件，其读写机制有一定的差异性，I/O 路径也不尽相同，这将使得他们在 I/O 读写效率以及资源占用上都有不同的表现。有兴趣着手构建超融合基础架构的用户，可能会希望了解更多关于 I/O 路径的细节，从而在实施之前，进行更充足的准备和更合理的规划（例如针对不同 I/O 路径选择更合适的网络带宽）。为了帮助读者更好地理解超融合架构 I/O 路径对集群性能的影响，本文将对比 VMware 和 SmartX 超融合 I/O 路径，分析不同场景下 I/O 读写的方式、发生概率及其对存储性能和集群的相关影响。

什么是 I/O 路径

业务系统经过运算后生成了数据，并通过系统的 I/O 路径将数据传输到存储介质上（一般是磁盘）进行持久化存储。通常情况下， I/O 的持久化存储过程会经历不同的硬件设备和软件逻辑，而经过每一个硬件/软件都会占用一定的系统资源并增加处理时间（产生时延），因此 I/O 路径的设计优劣跟业务系统的整体运行效率是息息相关的。

传统虚拟化架构下的 I/O 路径

在传统三层式基础架构中， I/O 从虚拟机端发起，需要经过 Hypervisor（虚拟化软件），然后通过主机的 FC HBA 卡，发送至 SAN 交换机，然后到 SAN 存储控制器，并最终将数据存储到物理磁盘当中。

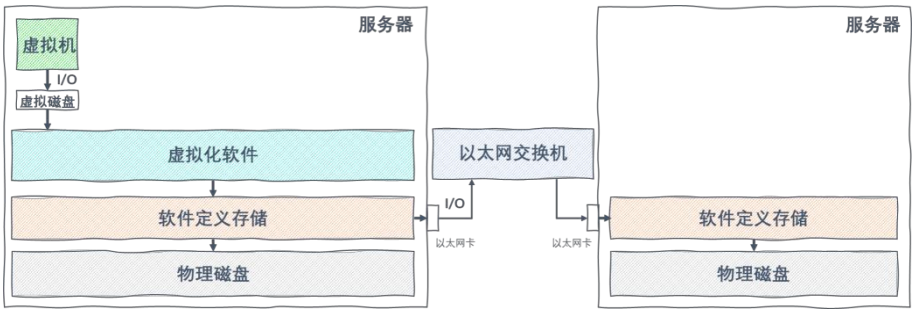


I/O 路径中涉及的硬件设备和软件逻辑包括：

- **硬件设备**：服务器主机 / SAN 存储网络交换机 / SAN 存储设备……
- **软件逻辑**：虚拟机操作系统 / 虚拟磁盘 / 虚拟化软件 ……

超融合环境下 I/O 路径

超融合架构将计算、网络和存储进行了融合，I/O 从虚拟机发起，同样需要经过 Hypervisor（虚拟化软件），然后发送到软件定义存储（SDS），最终在本地或通过以太网络将数据存储到物理磁盘当中。



I/O 路径中涉及的硬件设备和软件逻辑包括：

- **硬件设备**：服务器主机 / 以太网交换机……
- **软件逻辑**：虚拟机操作系统 / 虚拟磁盘 / 虚拟化软件 / 软件定义存储……

使用 SAN 存储的场景中，虚拟机数据必须通过 SAN 网络存储到服务器外部的 SAN 存储设备上，也就是数据读写 100% 需要经过网络。而与此不同的是，超融合架构里面虚拟机的 I/O 是通过内置的 SDS 存储软件写入磁盘的，而 SDS 属于分布式架构，I/O 既可能发生在主机内部的本地磁盘上，也有可能发生在外部的网络主机之上。

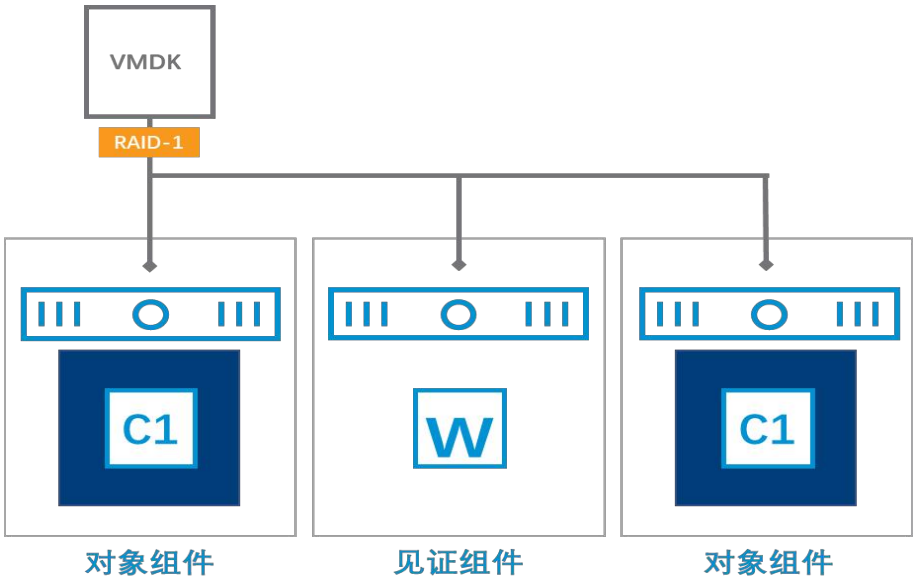
其中不难理解的一点：在同样的软、硬件条件下，本地 I/O 操作显然要比远程主机上的 I/O 操作的响应时间更短，毕竟 I/O 需要经过网络传输到远程节点执行，势必会增加 I/O 操作的时延。即使网络交换机的速度越来越快，依然无法完全避免时延的增加。

I/O 路径分析

vSAN 的 I/O 路径

VMware 超融合中的存储软件 vSAN 本质上是对象存储，它将虚拟机磁盘文件（.vmdk 文件）以对象（object）的形式进行存储，并提供了包括 FTT=1（RAID1 Mirror / RAID5），FTT=2（RAID1 Mirror / RAID6）等多种数据冗余机制。下面以较为常用的 FTT=1（RAID1 Mirror）为例展开 I/O 路径的讨论（RAID5/6 只适用于全闪集群，实际上混合存储在超融合环境更为常见）。

在 FTT=1 的存储策略下，虚拟磁盘（.vmdk）的副本数量是 2，两个副本会分别放置在 2 台不同的服务器主机上。而 vSAN 中 object 默认大小是 255GB，条带数为 1。举个例子：当虚拟机创建了一个 200GB 的虚拟磁盘，vSAN 会创建一组镜像组件，它包含 2 个 object 组件（实际上还有 1 个见证组件，但不包含业务数据，暂不讨论），分别放置在 2 台不同的主机上。如果虚拟磁盘的容量大于 255GB，则以 255GB 为单位拆分为多个 object。

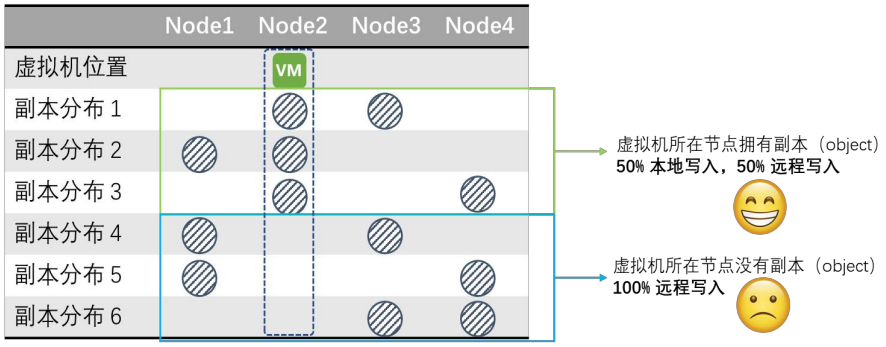


写 I/O 路径

正常状态下的 I/O 路径

前面提到在 SDS 当中，本地读写相比远程读写而言是一个更优的选择，因为它的时延更低，网络开销更少。vSAN 对于副本（object）的放置并没有优先写入本地的策略，而是随机写入两个节点。下面将分析 vSAN 在不同情况下的的写 I/O 路径。

以 4 节点的集群为例，2 个副本的放置节点位置共有 6 种可能性，当中有 3 种情况（½ 的概率），虚拟机写入的两个 object 均不在虚拟机运行所在服务器主机，需要 100% 远程写入（两个副本都需要经过网络进行写入），其余 3 种情况都是有一个副本是本地写入（另外一个副本经过网络进行写入），显然后者是更优的路径选择（2 个副本，必然导致至少有 1 个副本需要经过网络写入）。



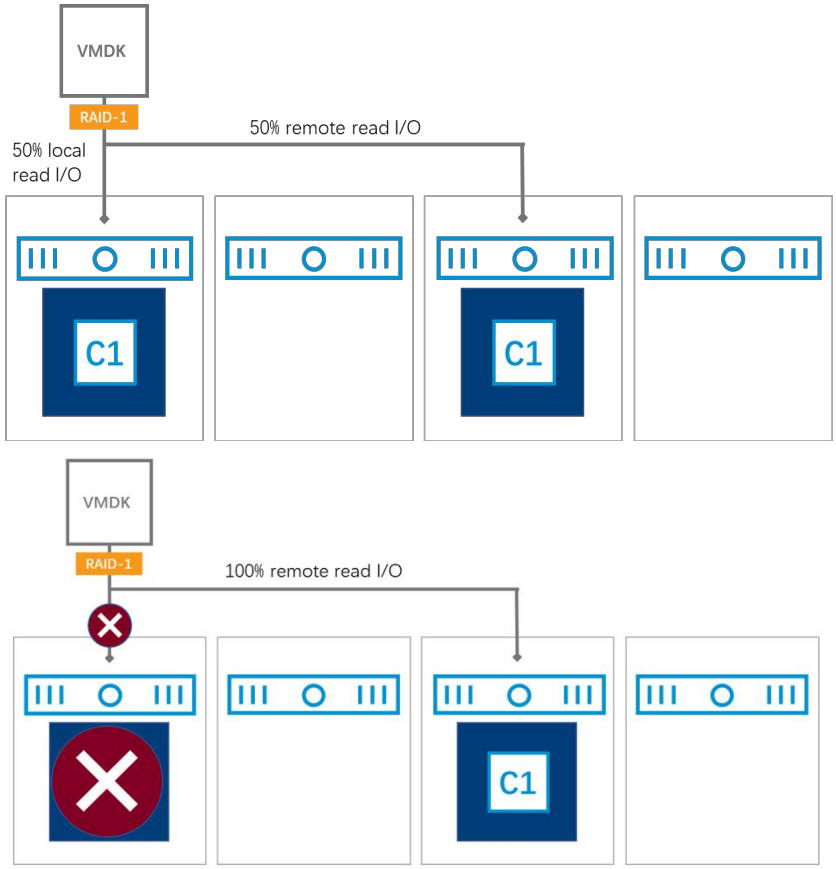
读 I/O 路径

正常状态下的 I/O 路径

根据 VMware World1 的技术资料透露，vSAN 的 I/O 读取会遵循 3 个原则：

- 副本间负载均衡读。
- 非必然发生的本地读（如果只剩下一个副本）。
- 确保同一数据块从同一个副本中读取。

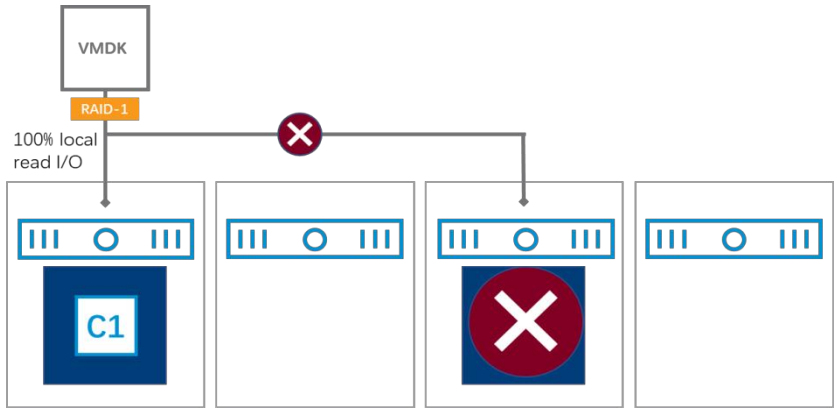
vSAN 的平衡读机制，意味着即使虚拟机所在的主机本地有数据副本（图 4 中虚拟机本地拥有副本的概率为 ½），也将会有 50% 的读取是通过网络进行的。另外，有 ½ 概率虚拟机所在的节点没有任何一个本地副本，需要 100% 远程读取。总而言之，vSAN 在正常状态下是不会发生 100% 本地读。



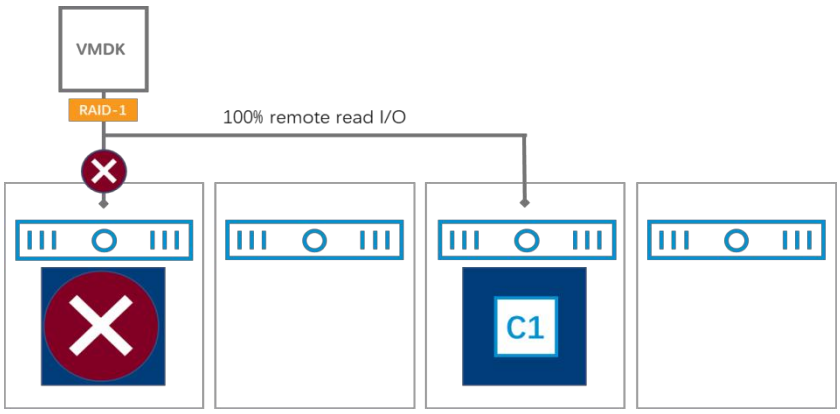
故障状态下的 I/O 路径

当集群中发生硬盘故障时，由于副本降级（其中 1 个副本由于硬盘故障而损失），无法再执行平衡读，所有读 I/O 将发生在同一个副本内。

其中，故障场景下将有 ¼ 概率发生 100% 本地读，这是相比正常状态更优的 I/O 路径。



剩余 ¾ 概率都是 100% 远程读。

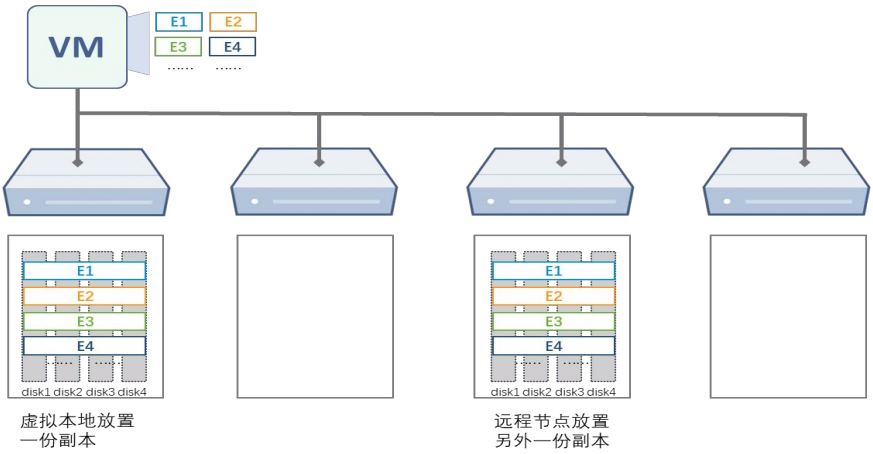


当硬盘遭遇故障时，需通过读取可用副本（唯一）进行数据恢复。由于 vSAN 中 object 的默认大小为 255GB，条带为 1，这种设置使得虚拟机数据副本很容易集中到单一、两块硬盘当中。在数据恢复时触发的读取操作容易受限于单块硬盘的性能瓶颈，难以利用多块硬盘执行并发恢复。因此，VMware 会建议在存储策略中通过增加“条带数”配置，以便尽量利用多块硬盘的读能力。

ZBS 的 I/O 路径

SmartX 分布式块存储 ZBS 将虚拟磁盘切分为多个数据块（extent），并为数据块提供 2 副本或 3 副本的数据冗余保护。其中 2 副本在数据冗余保护级别与 vSAN 的 FTT=1（RAID1 Mirror）是相对应的，下面将以 2 副本策略分析 ZBS 的 I/O 路径。

在 2 副本存储策略下，虚拟磁盘由多个数据块（extent 大小为 256MB）组成，而 extent 以一组镜像（Mirror）的方式存在，默认条带数为 4。ZBS 支持数据本地化功能，可精准控制副本的位置：虚拟机运行所在主机放置一份虚拟磁盘的完整副本，另外一份副本则放置在远程主机。



写 I/O 路径

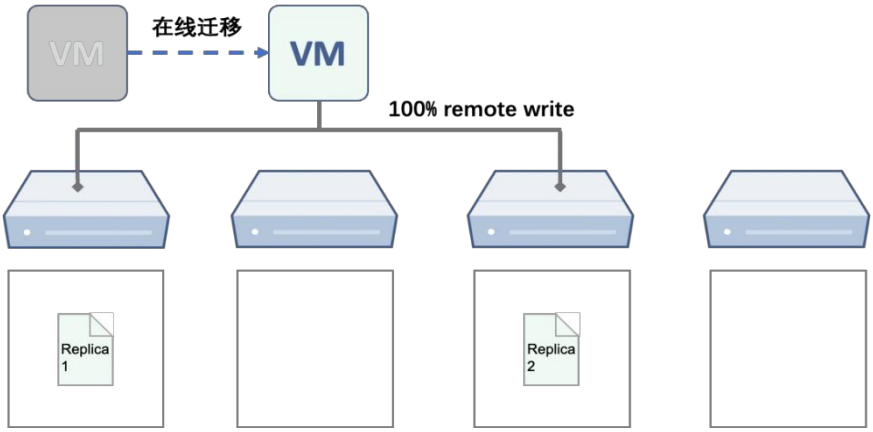
正常状态下的 I/O 路径

同样以 4 节点集群为例，由于 ZBS 可以保证虚拟机所在节点一定会有 1 个数据副本，写 I/O 操作可以一直保证 50% 写入发生在本地，50% 写入发生在远程，无论另外一个副本如何放置，都不会受到影响。因此，ZBS 在正常状态下不会发生 100% 远程写入的情况。

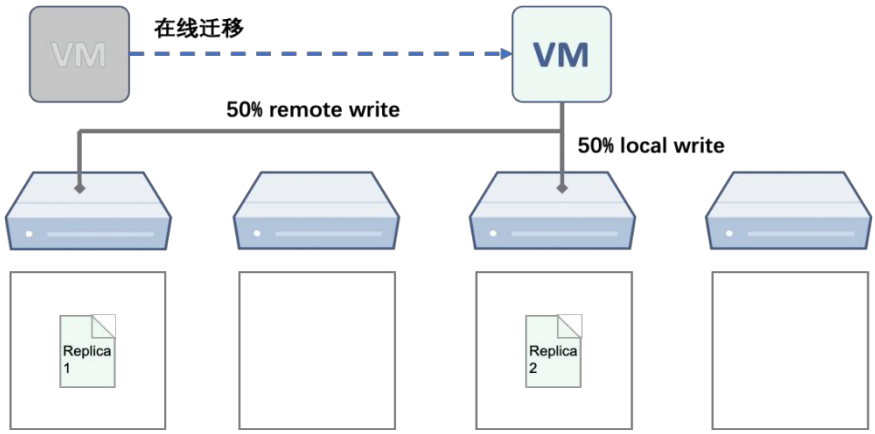
虚拟机迁移后的 I/O 路径

数据本地化功能可确保虚拟机的一份数据副本完整存放在本地主机，从而降低 I/O 访问的时延。但大家可能会思考一个问题：如果虚拟机发生了在线迁移，离开了原有主机，那么数据本地化是否失效了？通常来讲，迁移后的虚拟机可能遭遇到以下 2 种情况：

情况 1：迁移后，两个副本都不在本地，100% 远程写入（触发概率 66.6%）



情况 2：迁移后，移动到对应的副本位置，50% 远程写入（触发概率 33.3%）

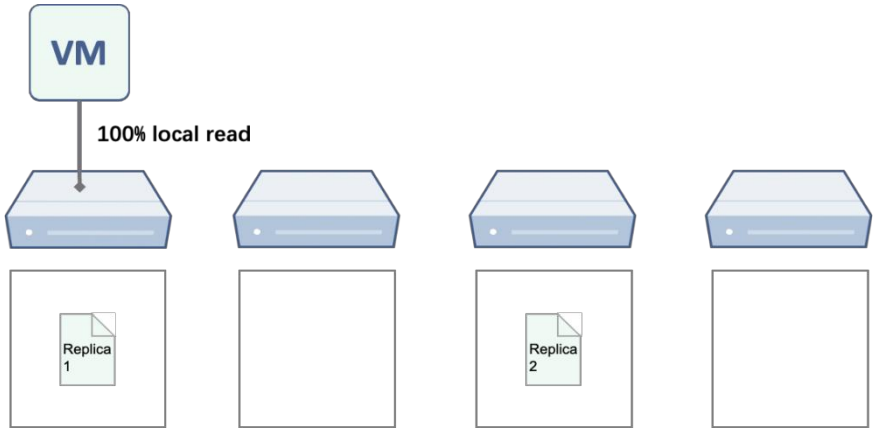


从分析中可以看到，当虚拟机迁移后，发生 100% 远程写入的概率比较高。ZBS 针对这类场景提供了专门的 I/O 路径优化：当虚拟机迁移后，新写入的数据将直接存放在本地新节点，并且会在 6 小时后，将虚拟机原有节点上对应的数据副本移动到新节点，重新形成数据本地化，同时可以解决迁移后远程读的问题。

读 I/O 路径

正常状态下的 I/O 路径

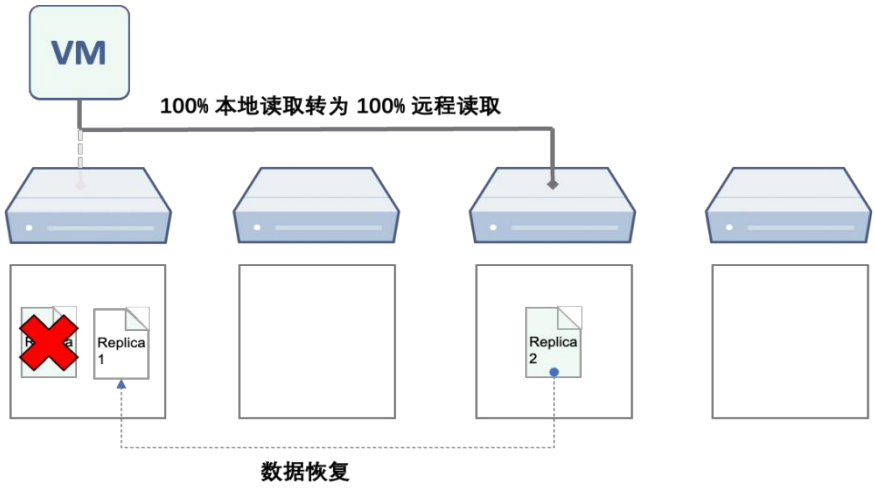
虚拟机运行所在的主机总是拥有一份完整的副本，可以一直确保 100% 本地读。



数据恢复状态下的 I/O 路径

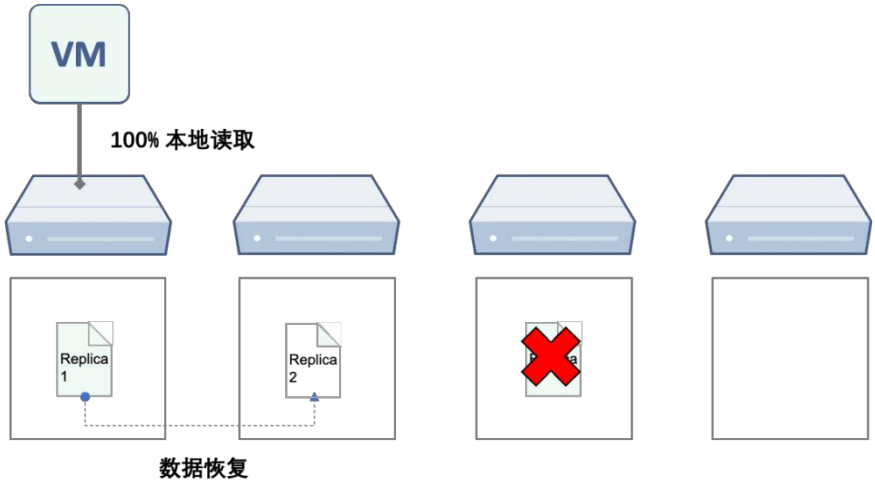
场景 1：虚拟机所在节点发生硬盘故障

I/O 访问从本地快速切换到远程节点维持正常业务的运行，同时触发数据恢复，优先在本地的可用空间进行恢复，并重新形成数据本地化。



场景 2：虚拟机远程节点发生硬盘故障

I/O 读取依然保持本地访问，并同时触发数据恢复。数据恢复的 I/O 流是从本地可用副本读取，然后向远程节点写入恢复数据。



大家可能会意识到：在数据恢复过程中，唯一可用的副本需要同时响应业务的正常 I/O 读写和数据恢复读访问，是否会给存储系统造成较大的压力？

ZBS 针对故障恢复场景也有专门的优化方案：

- ZBS 的数据块划分相比 vSAN 更小（vSAN object 大小为 255GB，ZBS extent 大小为 256MB），加上条带化策略，使得虚拟机的数据更容易分散在多个硬盘，利用多个硬盘的读取能力以及多个硬盘的写入能力，有效提升数据恢复速度，避免单个硬盘性能成为瓶颈。
- 内置弹性副本恢复策略：ZBS 支持自动感知当前业务压力并根据压力调整数据恢复速度。当节点 I/O 繁忙，恢复速度下降至最低水平；节点负载下降，系统会逐步提升恢复速度，以求平稳、快速地恢复数据副本级别。

小结

根据上面的 I/O 路径分析，我们汇总和对比 vSAN 与 ZBS 在不同状态的 I/O 路径情况，其中以 100% 本地访问为最优，50% 远程访问为次之，100% 远程访问为再次之。

场景	I/O 类型	vSAN			ZBS		
		100% 本地	50% 远程	100% 远程	100% 本地	50% 远程	100% 远程
正常状态	读	N/A	50% 概率	50% 概率	100% 概率	N/A	N/A
	写	N/A	50% 概率	50% 概率	N/A	100% 概率	N/A
在线迁移	读	N/A	50% 概率	50% 概率	33% 概率	N/A	66% 概率
	写	N/A	50% 概率	50% 概率	N/A	33% 概率	66% 概率
数据恢复	读	25% 概率	N/A	75% 概率	75% 概率	N/A	25% 概率
	写	N/A	48% 概率	52% 概率	N/A	100% 概率	N/A

从对比表格上看到，无论在**正常**还是**数据恢复**的状态下，ZBS 的本地 I/O 访问的概率都要比 vSAN 更高，理论上时延会更低；而 vSAN 在绝大部分情况下都会有远程 I/O 发生，换言之，对网络的占用要更高一些。在虚拟机发生**在线迁移**的场景下，ZBS 的本地 I/O 的概率有所下降，而 vSAN 的 I/O 路径显得要更好一些，随着 ZBS 重新完成数据本地化，这种情况会有所改善（至少需要几个小时）。

基于以上的分析，ZBS 在不频繁发生虚拟机在线迁移（几小时就发生一次迁移）的环境下具有明显的优势，反之，vSAN 会有一定优势。而在实际生产环境下，频繁的在线迁移并不常见。

I/O 路径对于集群的其他影响

文章上面提到过，在超融合场景中，本地 I/O 发生概率越大，理论上存储的性能也会更好。但同时也会有另外一种声音：如果超融合的整体 I/O 性能是有富余的，那么是否就不需要考虑本地读写的优势呢？答案依然是否定的，因为更多的远程 I/O，除了增加了时延，还额外占用了网络资源。在集群正常状态下，或许这些网络的开销未必会造成很大的压力，但针对部分特定场景，这部分影响还是无法忽略的，如：

- 虚拟机高密度部署

  当虚拟机发生远程 I/O 的比例比较高的时候，虚拟机部署密度增大，远程 I/O 的流量也会剧增，有可能最终无法满足虚拟机的 SLA 要求。

- 数据平衡

  当集群容量使用率较高时，系统一般会触发数据平衡（执行数据迁移），平衡过程中通常都会涉及主机之间的数据复制，这时候网络带宽的压力会较大。此时，数据平衡和虚拟机的远程 I/O 容易造成网络资源争抢的情况，使得数据平衡完成的时间延长，甚至带来不必要的风险。

- 数据恢复

  数据恢复与数据平衡类似，需要依赖网络完成数据复制，但通常情况下，数据恢复需要更多的网络带宽，对业务的影响更大。一旦出现资源争抢的情况，将延长数据恢复时间，引入更大的风险。

最终，为了避免网络资源争抢的问题，用户可能需要付出额外的成本（如切换至 25G 网络），也许这并不是用户所期望的。

写在最后

超融合在选型和规划上有许多值得关注的细节，充分了解和重视细节可以帮助用户最大程度发挥超融合架构的优势。后续我们将与读者探讨更多关于超融合基础架构的技术话题。

1 VMworld 2016: STO7875 – A Day in the Life of a vSAN I/O.
https://www.youtube.com/watch?v=oxi1_Eb7vxA

VMware 性能对比 | VMware 超融合国产替代之性能对比篇

点击链接阅读原文：VMware 超融合国产替代之性能对比篇

要点总结

信创趋势要求国内企业寻找 VMware 超融合的可替代方案，防止因国际形势急剧变化带来的业务风险。

与传统基础架构相比，超融合架构具有软件定义存储的性能和良好的扩展能力，合理分配每台虚拟机的 IOPS，提高资源利用效率，节省成本。

从存储性能来看，SmartX 超融合方案无论是基于传统架构还是超融合架构都可以替代 VMware 超融合方案。

从数据库场景看，SmartX 超融合方案在创建表测试和数据库运行效率测试中性能表现不俗，可替代 VMware 虚拟计算层。

SmartX 超融合产品具备生产就绪能力，能够满足生产业务的需求。

背景

作为老牌虚拟化厂商，VMware 进入中国已经有十几年，在国内累积了大量的用户群体，除了使用 vSphere 虚拟化，也有不少用户开始采用 vSphere + vSAN 构建超融合基础架构。但近年随着国际形势的急剧变化，不少国内企业开始加快“信创”步伐，并尝试寻找 VMware 超融合的可替代方案。用户在寻找替代方案时通常比较关注功能层面，却容易忽略超融合性能表现是否能真正满足生产业务的需求。市面上也不乏一些号称“功能齐全”的超融合替代方案，但由于性能跟不上而无法应用在生产环境，最终使得所有的功能幻化为“泡影”。鉴于此，本文将着重围绕性能展开讨论，并对比 VMware 和 SmartX 超融合的实测性能表现。

性能评估

性能问题会直接影响业务系统稳定性，因此在探讨超融合替代方案时需要对此重点关注。在虚拟化的场景下，随着虚拟机数量增多，有些时候用户能明显察觉到虚拟机运行变慢了，这种情况很可能是存储性能已到达了瓶颈。在传统基础架构中，存储性能取决于共享存储设备的性能；而在超融合架构则取决于软件定义存储的性能。

评估超融合平台是否具备良好的 I/O 响应速度，实际上需关注到每个虚拟机可分配的 IOPS。以一套容量为 80TB 的存储设备能提供 10000 IOPS 为例，其性能约为 0.125 IOPS/GB；假设一台 40GB 存储容量的虚拟机，它只能获得 5 IOPS，这样的性能对于维持业务正常运行是远远不够的。作为参考，Google 的标准 SSD 持久卷具有 30 IOPS/GB1 ，是该数值的 240 倍！或许大家认为以 SSD 的性能作为标准可能过于苛刻，我们可以举出另外一个例子：一台 Windows 虚拟机在启动的时候大约需要 30 IOPS；而进入系统后几乎不运行任何负载的情况下大约也需要 10–15 IOPS。如果您的虚拟机只有 5 IOPS 甚至更低，那么业务卡顿就是再正常不过的事情了。

如果存储设备能为每台虚拟机提供 50–100 IOPS，显然是一个不错的目标，可保障虚拟机较流畅地运行。但按照这个目标计算，80TB 的存储设备应该具备 100000–200000 IOPS。传统架构下，一款能支持 200000 IOPS 的存储设备，采购成本是比较高的，另外大部分用户使用虚拟机的数量是逐年增长的，当前期虚拟机数量比较少的时候，大部分投入都是闲置的，并不是十分合理的。这就不难理解，为什么越来越多的用户愿意尝试超融合基础架构；用户期望利用超融合架构良好的扩展能力，以更低的成本提升整体性能。

I/O 读写基准性能评估

在评估替代方案之前，需要对 VMware 方案的性能表现有一定的了解。下面以 VMware 超融合

（vSphere 7.0 + vSAN 7.0）为例，执行 I/O 读写性能测试，用作评估替代方案的基准。

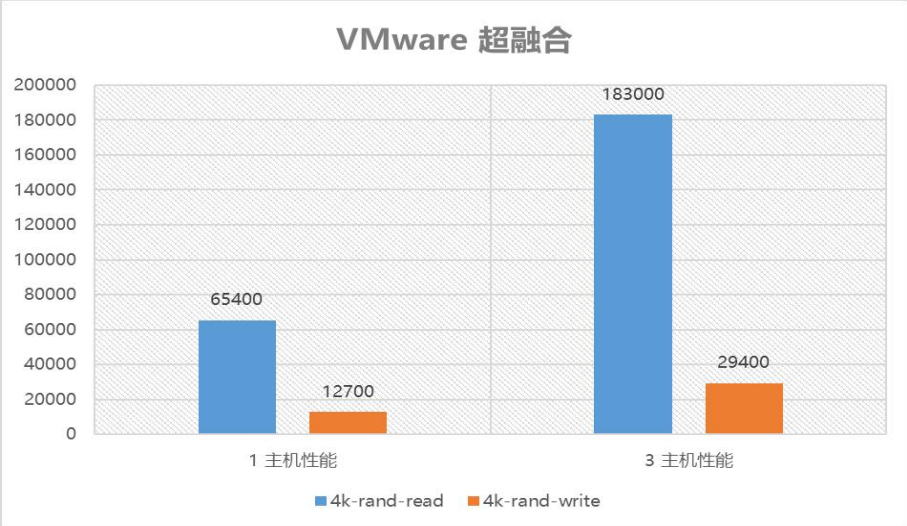
测试硬件配置

测试基于 3 台服务器组成的超融合集群，下表是单台服务器的硬件配置。存储网络采用 10GbE 交换机。

配件	型号	数量
CPU	Intel(R) Xeon(R) Gold 6128 CPU @ 3.40GHz	2
内存	16GB 2400 MHz	8
SSD（固态硬盘）	1.92TB SSD	2
HDD（机械硬盘）	2TB	4
Boot Disk（启动盘）	128GB SATA SSD	2
NIC (千兆网卡)	Intel i350-T2	1
NIC (万兆网卡)	Intel X710 2x10GbE	1

VMware 超融合测试结果

基于 FIO 测试工具执行 4k 随机读/写测试。下图为测试结果。



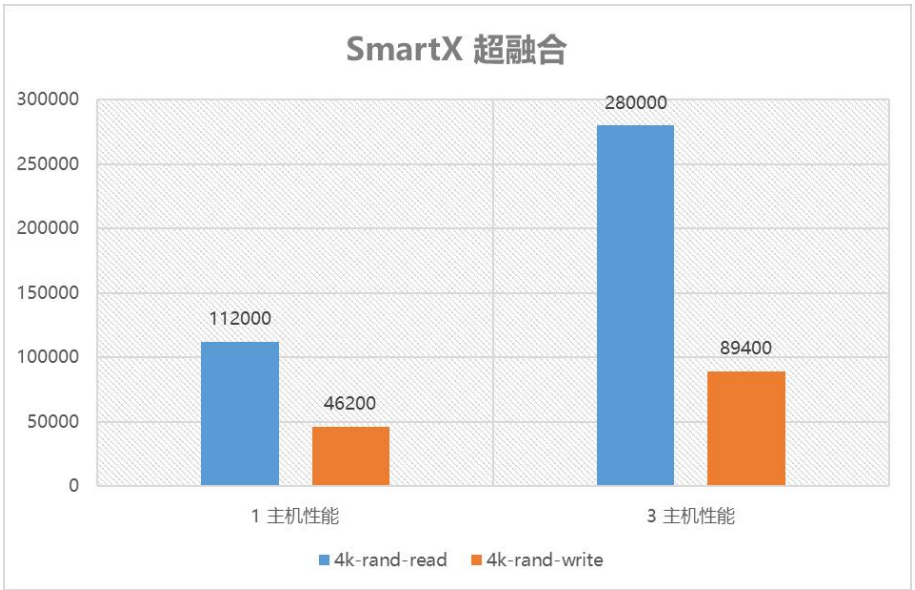
从结果可以看到：

VMware vSAN 的读写性能也是不错的，3 主机集群随机 4k 读取速度接近 20 万 IOPS（实际是 183000 IOPS）。如按读写比例 7:3 计算，混合读写的性能约为 136820 IOPS，性能指标处于目标的中游位置（目标是 80TB 存储容量可提供 10 万 – 20 万 IOPS，当前测试环境存储容量小于 80TB），从测试结果来看 VMware vSAN 性能是可以满足虚拟化场景的，在扩展节点数量后（5–6 节点集群）应该能轻松突破 20 万 IOPS。

以 VMware 超融合方案为参照，下面将在同一硬件条件下测试 SmartX 超融合方案，以评估方案性能是否能达到替代水平。

SmartX 超融合测试结果

基于 FIO 测试工具执行 4k 随机读/写测试。测试软件为 SmartX 超融合软件 SMTX OS 5.0。下图为测试结果。



从结果可以看到：

SMTX OS 在 3 主机集群下，随机 4k 读取速度达到 28 万 IOPS。如按读写比例 7:3 计算，混合读写的性能约为 22 万，已超过预期目标（目标是 10万 – 20万 IOPS）。单从存储性能方面看，SmartX 超融合方案无论是基于传统架构转型还是超融合方案替代都是完全没有问题的。但有读者可能会有疑问：除了存储性能之外，业务系统同时也需要强大、稳定的运算能力，那么替代方案是否具备同样性能？下面通过执行数据库测试验证计算与存储能力结合的情况。

MySQL 数据库性能评估

MySQL 数据库性能测试还是以 VMware 超融合集群上运行 MySQL 数据库的性能指标作为参照，观察 SmartX 超融合集群是否能达到相同水平或以上。

MySQL 测试共分两个部分：

- 通过 sysbench 创建 10 个表，每个表包含 1000 万行，从而测试表创建的效率（数据准备）。
- 通过 sysbench 插入数据条目（10 个表，每个表包含 1000 万行），模拟 OLTP 数据库运行压力。

测试硬件配置

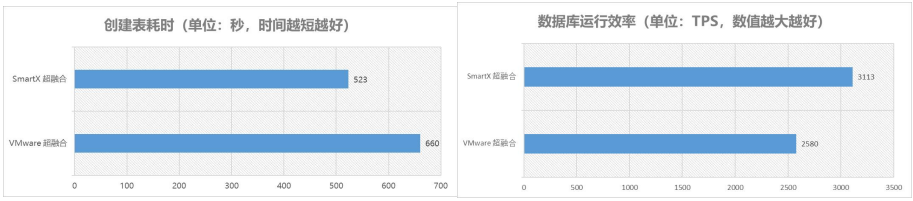
测试基于 3 台服务器组成的超融合集群，下表是单台服务器的硬件配置。存储网络采用 10GbE。

配件	型号	数量
CPU	Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz	2
内存	16GB 2400 MHz	8
SSD（固态硬盘）	1.92TB SSD	2
HDD（机械硬盘）	2TB	4
Boot Disk（启动盘）	240GB SATA SSD	2
NIC（千兆网卡）	Intel i350-T2	1
NIC（万兆网卡）	Intel X520 2x10GbE	1

测试脚本

脚本用途	脚本内容
数据准备	time sysbench --threads=100 --tables=10 --table-size=10000000 --db-driver=mysql --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=root --mysql-password='xxxxxx' oltp_read_write prepare
插入数据	sysbench --threads=100 --tables=10 --table-size=10000000 --time=100 --db-driver=mysql --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=root --mysql-password='xxxxxx' --report-interval=0 oltp_read_write run

测试结果



从 MySQL 数据库测试结果看，SmartX 超融合方案无论在创建表测试，还是数据库运行效率测试上都有一定的优势，也证实在数据库的场景下，可替代 VMware 虚拟计算层输出不俗的性能表现。

小结

超融合的性能表现是考察国产替代方案的重要一环，如果性能跟不上，方案只能一直停留于测试/开发环境，无法真正意义上实现替代。当然，考虑方案替代不会只有性能一个方面，未来我们将基于这个主题的其他方面作更多的探讨。

1 Google Cloud 磁盘性能描述：https://cloud.google.com/compute/docs/disks/performance

Nutanix 全面对比｜一文了解 SmartX 超融合替代可行性与迁移方案

点击链接阅读原文：Nutanix 国产化替代｜一文了解 SmartX 超融合替代可行性与迁移方案

要点总结

由于 Nutanix 中国市场销售模式的变化和 IT 基础架构的国产化趋势，不少 Nutanix 用户开始寻求国产超融合替代。作为独立超融合厂商，SmartX 坚持核心技术自主研发，在金融行业超融合软件市场占有率排名第一，收获良好口碑，成为 Nutanix 国产化替代的可行选择。

SmartX 超融合在产品组件与交付形式、分布式存储技术和实际性能上都不输于 Nutanix。在产品组件与交付形式上，SmartX 提供一体机、软件和订阅三种模式供用户灵活选择；在分布式存储技术上，SmartX 在群集核心组件配置、存储数据结构、数据 I/O 路径和数据冗余机制中表现更为出色；在 FIO 性能测试中，SmartX 测试结果均高于 Nutanix 超融合。

SmartX 超融合具有更强的硬件开放性，联合本土厂商推出更符合国内客户需求的联合解决方案，并提供更及时、专业和全面的服务，相比于 Nutanix 超融合更具差异化优势，能为国内客户创造更多价值。目前，越来越多的 Nutanix 用户已经转向 SmartX。

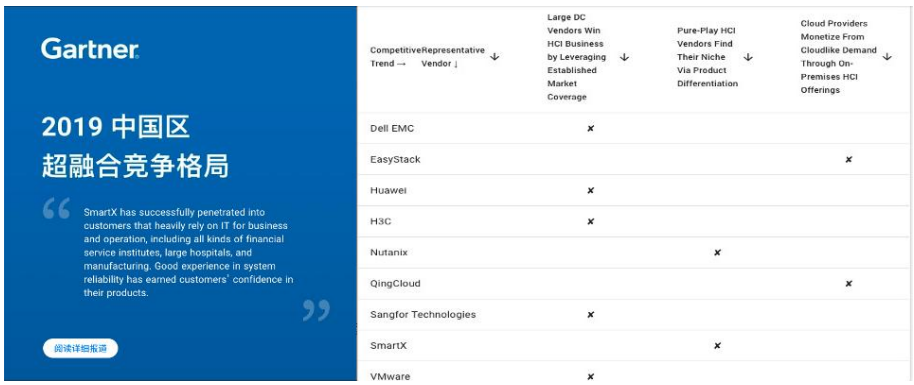
2022 年 8 月 19 日，Nutanix（路坦力）宣布中国市场自 2023 财年起将转型为合作伙伴销售主导模式，引起了广泛关注；同时结合当前 IT 基础架构的国产化趋势背景，不少正在使用和考虑使用 Nutanix 产品的企业开始寻求国产超融合替代方案。

作为国内最早投入超融合自主研发并大规模商用的团队，志凌海纳 SmartX 已经成为不少用户的云化和信创转型选择。今天，**我们将重点分析对比 Nutanix 与 SmartX 在市场格局、技术特性和性能表现上的差异，充分探讨 SmartX 超融合替代 Nutanix 的可行性**。最后，文末也给出了将业务从 Nutanix 迁移至 SmartX 超融合的具体方法，帮助有需求的用户快速实现替换。

替代可行性之市场、产品、技术、性能对比

国内超融合市场格局对比

根据 Gartner 发布的《2019 中国区超融合竞争格局》报告，SmartX 和 Nutanix 均为独立超融合厂商（Pure-play HCI Vendor），两者都专注超融合领域技术开发，完全依赖领先的核心技术、专业的产品和服务与传统大厂竞争，并赢取客户信赖。双方都聚焦在金融、医疗、制造等对 IT 高度依赖，并且对产品和服务要求苛刻的行业。



同时，SmartX 在国内金融市场份额表现突出。根据 IDC 《中国软件定义存储 (SDS) 及超融合存储 (HCI) 系统市场季度跟踪报告，2021 年第四季度》，中国超融合软件市场份额排名靠前的国产品牌分别是华为、新华三、深信服、浪潮和 SmartX。其中，SmartX 在金融行业超融合软件市场占有率排名第一，收获良

好口碑。目前，SmartX 超融合在金融行业已落地超过 200 家客户，总计部署超融合节点超过 4000 个，**其中信创相关节点超过 700 个，80% 是生产业务案例**。

下面，我们将从逐一对比 Nutanix 和 SmartX 超融合在产品组件、分布式存储技术和实际性能上的表现。

产品组件与交付形式对比

测试硬件配置

以下针对双方超融合基础架构核心产品特性与相关组件进行对比：

	Nutanix	SmartX
计算虚拟化	**原生虚拟化 AHV** 支持第三方虚拟化： VMware vSphere Microsoft Hyper-v Citrix Xenserver	**原生虚拟化 ELF** 支持第三方虚拟化： VMware vSphere Citrix Xenserver
分布式存储	**自主研发的** **AOS**	**自主研发的** **ZBS**
双活	☑	☑
异步复制	☑	☑
网络安全	Flow	Everoute
备份与恢复	Mine Integrated Backup	SMTX Backup & Recovery

结论：目前在超融合基础架构的核心产品功能和组件层面，SmartX 不仅同样具备自主研发的存储核心、原生虚拟化以及多虚拟化平台支持，同时也具备网络与安全、备份与恢复等产品。

另外，**Nutanix 目前已经全面转型到订阅交付模式，而 SmartX 目前提供一体机、软件和订阅三种模式供用户灵活选择**。

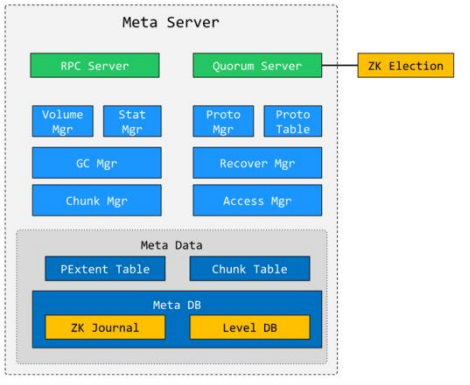
以下，针对超融合最核心的组件——分布式块存储的技术架构进行更详细的对比。

分布式存储技术对比

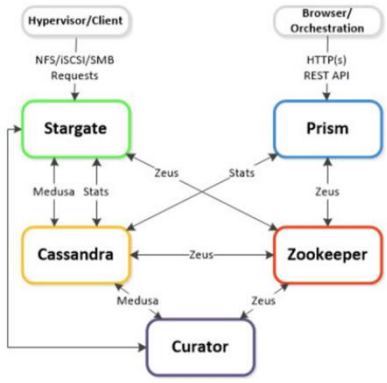
目前市场上的分布式存储技术架构基本上分两大类：一类是围绕开源技术 Ceph 提供的解决方案，一类是通过自研的方式提供解决方案。SmartX 和 Nutanix 在此都走了自研技术的路线，且都借鉴了 GFS（Google File System）的技术架构。更多关于 ZBS 的架构解析，请阅读：分布式块存储 ZBS 的自主研发之旅 | 架构篇。

从广泛意义上讲，分布式存储中通常需要解决三个问题，分别是元数据服务、数据存储引擎，以及一致性协议。不同分布式存储系统之间的区别，主要来自于这三个方面的不同选择。接下来我们通过对比 SmartX 和 Nutanix 核心关键组件最终的实现方法来看看两者之间的关系。

群集核心组件关系对比



SmartX

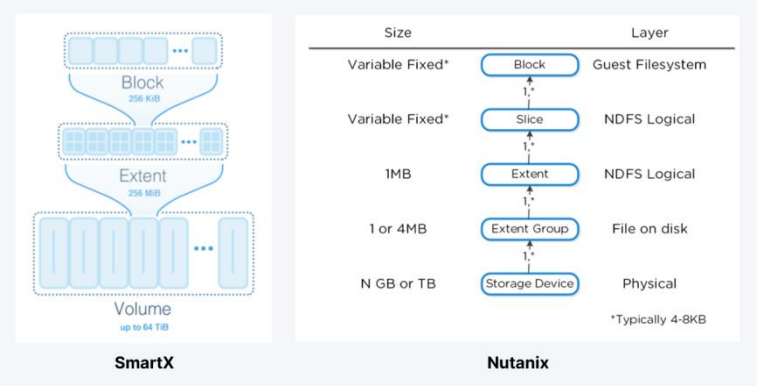


Nutanix

功能	SmartX	Nutanix	相同特征	区别
集群配置服务	Zookeeper	Zookeeper	存储群集配置信息、负责选举“leader”角色用于保证数据一致性；3 节点或 5 节点部署（取决副本数量）	SmartX：Zookeeper 承担了更多的任务，除了选“主”还包括重要的 log replication
元数据服务	MetaServer	Cassandra	集群元数据信息	SmartX：Metaserver 跟 Zookeeper 运行在相同的节点上，多个 Metaserver 构成一个群集，3 节点或 5 节点部署 Nutanix：Cassandra 运行在群集中所有节点，采用环形架构，成员之间是 Peer 的关系
数据访问控制服务	AccessServer	Stargate	用于接受和处理数据,运行在每个节点上；支持 NFS/iSCSI	Nutanix：SMB，支持 Hyper-V
磁盘设备管理服务	ChunkServer	Stargate	物理磁盘（SSD 和 HDD）控制	N/A
数据管理服务	TaskServer	Cerebro	数据复制、快照、容灾	N/A
路径重定向	iSCSI 重定向或 I/O rerouter	Data Service IP	提供的统一地址，可靠地使用 iSCSI 接入服务	N/A

通过上面群集核心组件的对比，能够看出两个方案采用了相似的架构。但另外一方面， ZBS 针对一些关键组件进行了技术优化。例如，元数据管理关键组件 Zookeeper 和 Cassandra 本身存在一些局限性：Zookeeper 存储的数据容量非常有限，且无法和数据服务混合部署在一起；Cassandra 不提供 ACID 机制，在上层实现时会比较复杂，需要额外的工作量。ZBS 选择采用 LevelDB 和 Zookeeper 结合的方式，有效避免这些问题，从而实现可靠性、高性能和轻量化的目标。

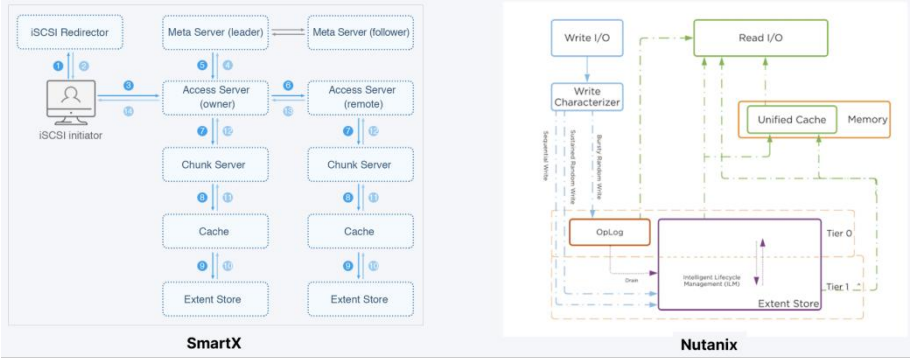
存储数据结构



功能	ZBS	NDFS	相同特点	区别对比
最小物理单位	Block	Block	Block 为最小数据单元	ZBS：256KB NDFS：通常是 4–8KB（取决于 Guest 文件系统）
数据块	Extent	Extent	由 Block 组成	ZBS：256MB NDFS：1M
条带	Striping	Extent	利用多个磁盘的 I/O 能力	ZBS：1、2、4（4KB–256KB） NDFS：1M
存储卷	Volume	Container/ Volume	逻辑单元，支持精简配置，可以是虚拟磁盘或 LUN	ZBS：支持精简或厚置备 NDFS：仅支持精简

通过上面的对比，可以看出两种方案选择了不同的块大小单位。ZBS 采用较大粒度的基本数据单元（数据块），可减少元数据消耗的内存资源，保证全部元数据都可以保存在内存中，提高访问效率。除了支持精简置备外，ZBS 也同时支持厚置备磁盘模式。

数据 I/O 路径



功能	ZBS	NDFS	相同特点	区别对比
Write/写	分全局 I/O 路径和本地 I/O 路径	按 I/O 特征（随机或顺序）进行分类，不同的 I/O 特征写到不同存储层，随机 I/O 写到 oplog（SSD），顺序 I/O 写到持久化层（HDD）	数据本地化、数据自动平衡；元数据副本（3/5）均采用 LRU（近期最少使用算法），通过两级 LRU 链表来管理数据冷热程度	ZBS: 1.采用较大粒度的数据单元(256M) 2.元数据缓存在内存中 NDFS: 1.读缓存横跨 CVM 的内存和 SSD
Read/读	数据在缓存中直接响应，数据不在缓存中从 HDD 读取并保留在缓存			

在数据访问 I/O 路径上，两者同样具备很多相似性，例如：自动分层、数据本地化、数据自动平衡等。但 ZBS 的元数据缓存在内存中，这样可以加快针对元数据的操作响应速度。同时，元数据在所有节点中都保存一份副本，保证元数据的高可靠，这样即使有部分服务器发生宕机，元数据也不会丢失。

数据冗余机制

功能	ZBS	NDSF	相同特点	区别对比
副本	1/2/3	1/2/3	数据本地化、拓扑安全、数据校验、容量均衡、数据动态移动	ZBS：数据存放策略和数据恢复策略更丰富和动态。例如：根据磁盘容量负载（低/中/高）存放 NDFS：支持在线增加或减少副本数量
快照	COW	ROW	两种快照实现的逻辑是相同的；快照之间没有依赖关系，实现秒级快照；无快照队列深度带来的读性能影响	ZBS：基于存储卷（由多个固定大小数据块组成 ） NDFS：基于 vdisk 文件（extend 组成的逻辑上连续的数据块）
克隆	同上	同上	克隆都是基于自有的快照技术	
异步复制/双活	TaskServer	Cerebro	基于快照技术、一对多、保护域、保护周期、增量复制、数据传输优化（压缩）	ZBS：15 分钟 NDFS：准同步，分钟级（NearSync）

在数据保护层面，两者也采用了类似的技术，但 ZBS 在副本分配策略上提供了更丰富的方式，例如：局部化、副本分配策略动态调整（根据存储容量空间占比大小进行动态调整）。ZBS 快照的元数据是存储在 ZBS 分布式存储元数据服务集群内的。元数据位于内存中，有更好的响应速度，同时元数据也会持久化同步到 SSD 介质上，这样即使是主机重启后，也可以通过 SSD 快速加载元数据到内存当中，不会因为主机重启而降低快照性能。

更多关于 SmartX 元数据存储与快照技术解析的内容，请阅读：VMware 与 SmartX 快照原理浅析与 I/O 性能对比。

性能对比

性能表现通常是用户比较关心的问题，因此，我们选取用户对 SmartX 超融合（SMTX OS 5.0.3）和 Nutanix 超融合（AOS 5.20.4.6）的实际评测，对比说明两家产品在测试用例中的表现差异。以下所有测试基于完全相同的物理硬件设备环境，并进行了多角度的测试。

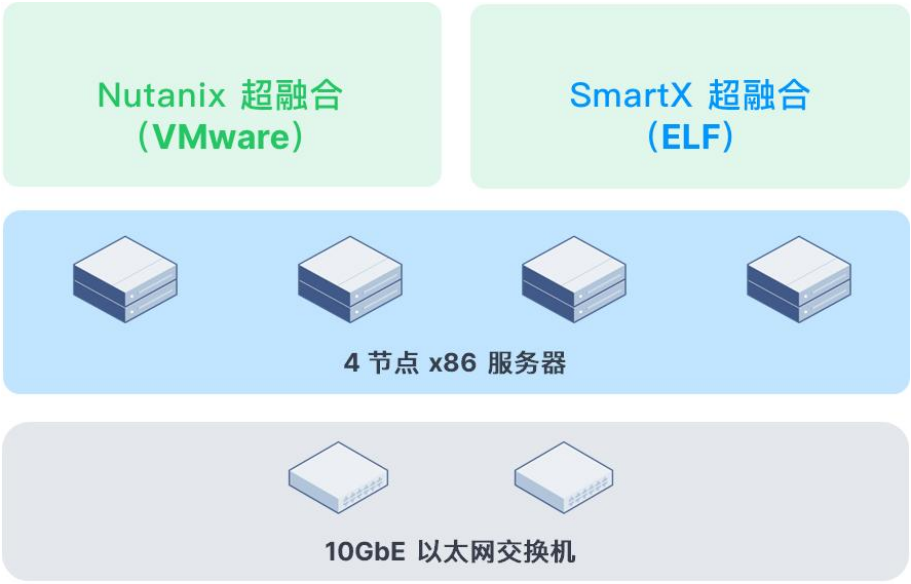
注：SmartX 超融合使用的是原生虚拟化 ELF，并开启了 Boost 模式，而 Nutanix 超融合使用的是 VMware ESXi 6.7u3B 虚拟化平台。

FIO 测试

首先针对每个虚拟机的 100GB 数据磁盘进行全盘数据写入，再对数据磁盘进行 4P1V 和 4P4V 不同 I/O 模型的性能测试。

- 4P1V：代表 4 个节点运行 1 台虚拟机，只对 1 个虚拟机进行性能测试，也就是模拟单个业务系统在集群内所能获得的性能。
- 4P4V：代表 4 个虚拟机分布在 4 个不同的节点上，对 4 个虚拟机同时进行性能测试，也就是模拟整个集群所能提供的性能总和。

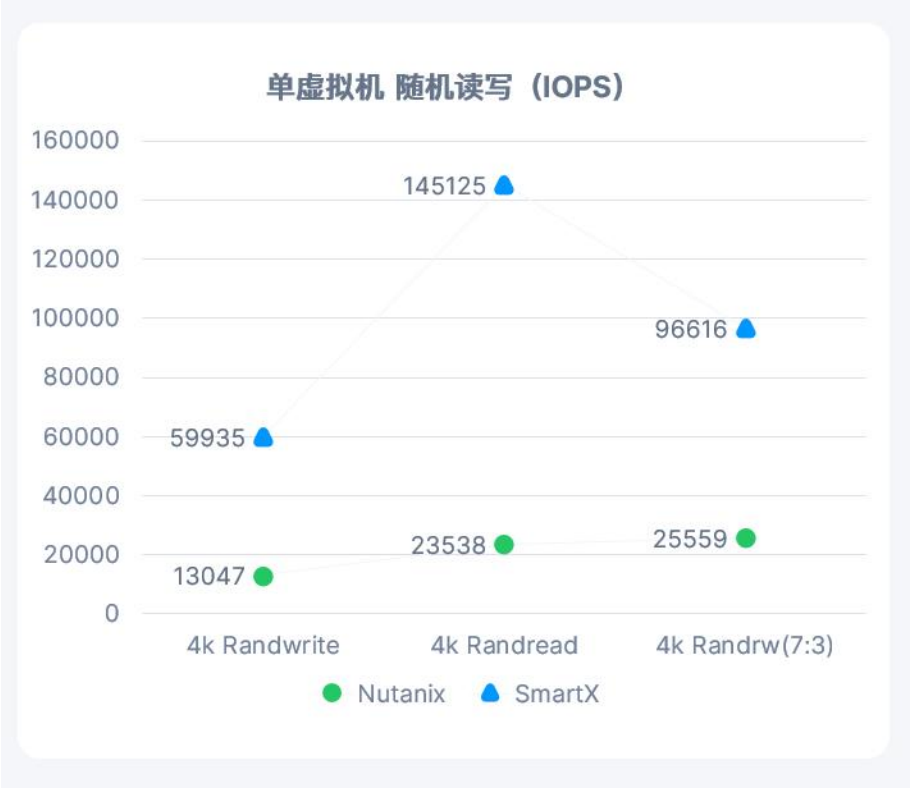
测试环境



x86 单节点服务器硬件配置

- CPU：Intel E5-2699v4 22C 2.2GHz *2
- SSD：1.92TB SATA SSD *2
- MEM：1 TB
- HDD：6TB SATA *4

场景一：随机读/写 IOPS 对比（4P1V 单虚拟机）



说明：4k Randrw 混合读写场景下，Nutanix 读 IOPS 为 17885，写 IOPS 为 7674；SmartX 读 IOPS 为 67633，写 IOPS 为 28983。

场景二：顺序读/写 带宽对比（4P1V 单虚拟机）

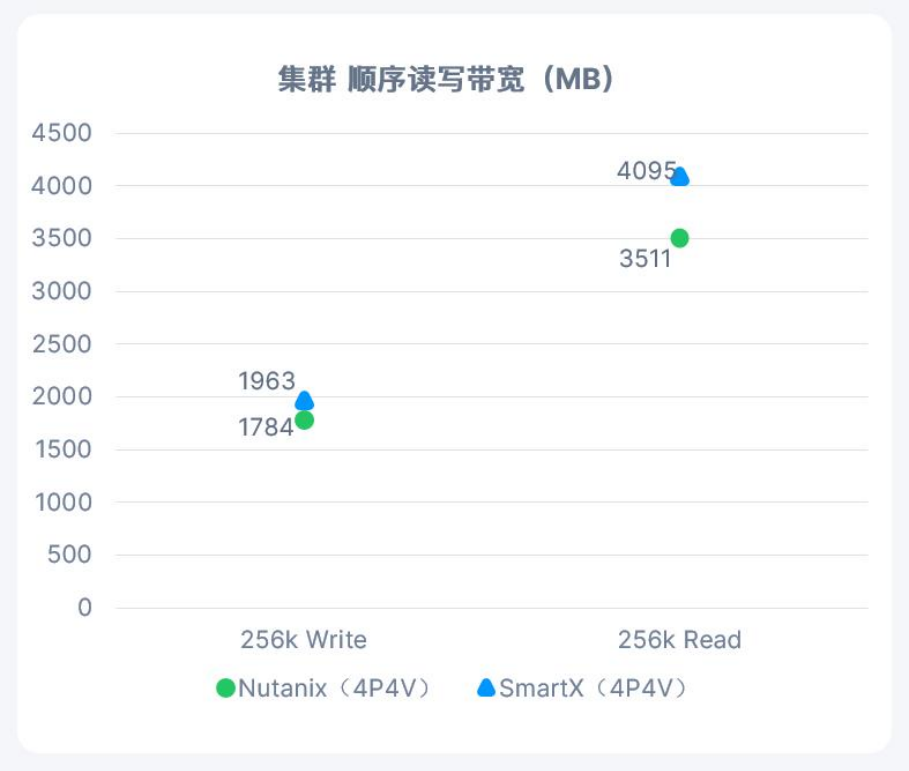

单虚拟机 顺序读写时延（ms）
70
60
50
40
30
20
10
0
60.24
34.01
38.61
31.2
256k Write
256k Read
Nutanix
SmartX


单虚拟机 顺序读写带宽（MB）
1200
1000
800
600
400
200
0
963
543
1025
848
256k Write
256k Read
Nutanix
SmartX

场景三：随机读/写 IOPS 对比（4P4V 集群）

说明：4k Randrw 混合读写场景下，Nutanix 4 个虚拟机读 IOPS 分别为 17738、18035、18426、18330，写 IOPS 分别为 7623、7718、7881、7862；SmartX 4 个虚拟机读 IOPS 分别为 48193、46668、46466、46997，写 IOPS 分别为 20655、20001、19929、20156。

场景四：顺序读/写 带宽对比（4P4V 集群）



可以看到，SmartX 超融合在以上测试用例中，测试结果均高于 Nutanix 超融合。

SmartX 超融合的差异性优势

虽然 SmartX 和 Nutanix 超融合“两者的技术基因非常类似”，但 SmartX 作为国内最早专注于自主研发超融合软件的公司，还在产品、方案和服务层面提供了差异化的能力，为国内客户创造更大价值。

产品：更强的硬件开放性

SmartX 超融合支持集群异构，可将不同品牌、不同配置（CPU、内存和磁盘）的节点统一在一个集群内部，这样可以满足不同环境的各种需求，例如不同时期的不同配置或者不同品牌服务器的利旧。而在这一点上，Nutanix 有严格的品牌一致性要求。这一特性在 SmartX 超融合客户实践中得到了充分体现：五矿期货超融合硬件平滑升级与多数据中心管理实战。

欲了解更多 SmartX 异构节点支持特性与软硬件平滑升级实践，请阅读：
如何做到 IT 基础架构软硬件升级简单又不停机?
不止弹性，更加灵活。一文了解 SmartX 超融合如何扩容

方案：联合本土厂商推出更符合国内客户需求的联合解决方案

目前，SmartX 已经在硬件服务器、操作系统、数据库、云管、备份等层面与本土厂商进行方案整合，以便满足国内客户在信创等方面的特定需求。

欲了解更多 SmartX 与国内厂商发布的联合解决方案，请阅读：
SMTX OS 成为国内首个获得鲲鹏 Validated 认证的超融合软件 | 信创生态
SmartX 携手爱数发布无代理虚拟化备份联合解决方案

服务：更及时全面的本地化服务

除了专业的产品和方案，SmartX 及时、专业和全面的服务同样得到用户的认可，这也成为 SmartX 超融合达到近 70% 复购率的重要原因：不仅源于掌控核心技术的本土团队，更来自于 SmartX 从研发、产品到一线团队的 360 度闭环服务体系。

越来越多的原 Nutanix 客户转向 SmartX

基于上述双方的相同点和 SmartX 的优势，越来越多的 Nutanix 用户在与 SmartX 进行接触后，对 SmartX 的产品能力、价值、可替代性以及优秀的本地化服务给予了充分的肯定，**并从原有 Nutanix 平台迁移到 SmartX 平台，其中包括日立电梯、东方证券、交通银行、松下万宝、国泰君安等在内的行业头部客户**。

如何从 Nutanix 超融合迁移至 SmartX 超融合

最后，我们结合 Nutanix 用户现有可能的技术架构，将业务迁移到 SmartX 平台的技术路径进行简单的汇总。

原有环境	目标环境	业务可用性	迁移过程
VMware + AOS	VMware + ZBS	**“零”中断**	利用 ESXi 本身的 vMotion 和 Storage vMotion 实现在线的业务迁移
VMware + AOS	ELF + ZBS	**分钟级**	利用 SmartX 提供的 V2V 工具，无需在原有环境安装任何插件
AHV + AOS	ELF + ZBS	**分钟/小时**	1.利用迁移工具 2.将原有虚拟机磁盘导出并上传到目标环境，虚拟机挂载上传后的虚拟机磁盘，启动虚拟机

总结

通过以上分析可以看出，SmartX 不仅在市场定位、产品组件、整体技术实现层面与 Nutanix 具有较高的相似性，更是在超融合产品性能、产品开放性和本地化方面有着更出色的表现，是 Nutanix 国产化替代的理想方案选择。

弹性恢复 | 通过弹性副本恢复策略平衡数据恢复速度与业务 I/O 性能

点击链接阅读原文：通过弹性副本恢复策略平衡数据恢复速度与业务 I/O 性能

要点总结

SmartX 分布式存储 ZBS 经过多个版本迭代，引入和优化副本弹性恢复/迁移策略，在 SMTX OS 4.0.12 及以上版本、SMTX OS 5.0.3 及以上版本、SMTX ZBS 5.1.0 及以上版本中存储系统单词下发的恢复指令数量提升 28%，集群硬件性能进一步得到合理利用。

SmartX 分布式存储的弹性副本恢复策略采用自研存储引擎，使存储系统自动识别业务 I/O 和数据 恢复/迁移 I/O，进行业务 I/O 的 IOPS 和 BW 与设定阈值对比，根据负载灵敏调节限速。

集群异常触发数据恢复时，SmartX 分布式存储优先确保业务 I/O 正常使用，并将数据恢复/迁移速度调整至合理数值，实现数据恢复速度与业务 I/O 性能的平衡。

场景问题

分布式存储集群在硬件配置确定后，集群性能的物理上限也随之确定。分布式存储集群因硬盘损坏、节点宕机等异常问题，触发数据恢复，若此时业务处于高峰期，集群的性能应当优先保证业务的使用，还是优先保证副本恢复以确保数据的安全?

本文描述了 SmartX 核心产品组件分布式存储 ZBS 的弹性恢复策略和实际应用效果。其中 SMTX OS 是包含 ZBS 组件的超融合产品软件，SMTX ZBS 是包含 ZBS 组件的存算分离产品形态——分布式存储软件。

SmartX 怎么做

3.0 / 3.5 版本

通过对副本恢复 / 迁移进行限速，单节点恢复和迁移限速分别是 100 MiB/s 和 40 MiB/s，防止副本恢复 / 迁移占用大量分布式存储性能，影响业务 I/O 性能。

4.0 版本

自 4.0 开始，SmartX 引入副本弹性恢复 / 迁移策略，提供了两种模式供用户选择，以便调整默认限速。

- **智能调节（AUTO）**：此为默认模式。以保护业务 I/O 性能为前提，每个节点根据自身当前承载的业务 I/O 负载，自动调整本节点副本恢复 / 迁移速度。在业务 I/O 压力较大时，确保业务性能不受影响。
- **静态调节（STATIC）**：用户人工设置集群最大速度限额。当用户希望最大程度保护业务 I/O 时，可以将恢复速度设置为较小的值（例如 40 MiB/s）。当用户希望加速恢复时，可以设置为较大值（例如 500 MiB/s）。静态设置在集群内全局生效，所有节点使用同一限额。

两种模式都需要保证设置的限速落于合法范围内。取值的依据是：

- 默认值为 100 MiB/s
- 上限值为 500 MiB/s
- 下限值为 1 MiB/s

5.0.0 版本

针对节点不同的硬件配置，综合考虑存储网络（10 GbE、25 GbE、是否开启 RDMA）和存储介质（SATA HDD、SATA SSD、NVMe SSD、PMem）的能力，设计不同的业务压力触发降速阈值，确保节

点自适应设置副本恢复 / 迁移速率阈值。

新版本

为了充分利用集群的数据处理能力，SmartX 分布式存储在以下新版本中再次优化策略，存储系统单次下发的恢复指令数量提升 28%，进一步合理利用集群硬件性能。

- SMTX OS 4.0.12 及以上版本
- SMTX OS 5.0.3 及以上版本
- SMTX ZBS 5.1.0 及以上版本

业务 I/O vs. 恢复 / 迁移 I/O

在 SmartX 分布式存储中，I/O 分为业务 I/O 和数据恢复 / 迁移 I/O 两种。

智能调节（AUTO）的目标是，遵循业务 I/O 优先的原则，对数据恢复 / 迁移 I/O 的速率进行调节，以加速恢复任务的完成。

SmartX 分布式存储采用自研存储引擎，存储系统可自动识别业务 I/O 和数据恢复 / 迁移 I/O。根据业务 I/O 的 IOPS 和 BW 与设定阈值的对比，判定业务 I/O 为空闲或繁忙状态。

- 当业务 I/O 的 IOPS 或 BW >= 阈值时，一次性将节点的副本恢复 / 迁移限速调整到默认值，起到保护业务 I/O 的目的；
- 当业务 I/O 的 IOPS < 阈值且 BW < 阈值，且恢复 / 迁移速度超过当前限速的 80%，说明恢复 / 迁移速度快要达到限速，系统自动提高限速值，起到加速恢复的目的（每次提升限速为原值的 1.5 倍，直至达到上限值）。

该策略采用快速增减的调节方式，每隔 4s 重新判定业务 I/O 的大小，可以灵敏地根据负载调节限速。

常见场景

以常见硬件组合为例，数据恢复 / 迁移的最小、最大速率和触发限速的业务 I/O 阈值如下图所示

SSD	网卡	业务 IO 阈值	数据恢复（MiB/s）		数据迁移（MiB/s）	
			最小值	最大值	最小值	最大值
2 * SATA SSD	10 GbE	IOPS = 1500 BW = 150 MiB/s	100	500	50	500
2 * NVMe SSD	10 GbE	IOPS = 5000 BW = 500 MiB/s	100	500	50	500
2 * NVMe SSD	25 GbE	IOPS = 5000 BW = 500 MiB/s	240	1200	120	1200
2 * NVMe SSD	25 GbE + RDMA	IOPS = 5000 BW = 500 MiB/s	312.5	1562.5	156.3	1562.5

如需禁用恢复，使用命令 zbs-meta recover disable。

注意：禁用数据恢复会造成数据副本不及预期时无法恢复副本，业务系统在此状态下运行存在一定的数据风险，通常情况下数据恢复的策略建议是 ASAP（越快越好）。

功能验证

本次测试验证采用三台服务器组建超融合集群，集群部署完成后安装 CentOS 作为业务虚拟机，在虚拟机

内部运行 Fio 模拟业务 I/O。

测试环境

硬件配置

部件	型号	数量
CPU	Intel Silver 4214R 2.40 GHz	2
MEM	32 GB	4
SSD	Intel S4610 960 GB	2
HDD	2 TB SATA	4
NIC	Intel X710 10GbE DP	2

软件版本

软件	版本	用途
SMTX OS	5.0.2	SmartX HCI OS
CentOS	7.9	业务虚拟机
Fio	2.15	通用性能测试工具

测试组网

- 10GbE 网络
- 未开启 RDMA

测试步骤

1. 虚拟机 vm-01 位于第一台服务器 node-01 上
2. 强制关闭第三台服务器 node-03，制造大量数据恢复环境
3. 观察到 node-01 数据恢复速度为 500 MiB/s 左右
4. 虚拟机 vm-01 中运行以下命令模拟业务 I/O

fio -ioengine=libaio -invalidate=1 -rw=randwrite -iodepth=128 -direct=1 -size=100g -name=smtx-fio -bs=256k -filename=/dev/vdb -time_based -runtime=3600

```
[root@fio-1 ~]# fio 256k-write.fio
job: (g=0): rw=write, bs=256K-256K/256K-256K/256K-256K, ioengine=libaio, iodepth=128
fio-2.15
Starting 1 thread
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/857.0MB/0KB /s] [0/3428/0 iops] [eta 13d:21h:19m:56s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/851.3MB/0KB /s] [0/3405/0 iops] [eta 13d:21h:19m:56s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/851.4MB/0KB /s] [0/3405/0 iops] [eta 13d:21h:19m:55s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/872.7MB/0KB /s] [0/3490/0 iops] [eta 13d:21h:19m:53s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/852.8MB/0KB /s] [0/3411/0 iops] [eta 13d:21h:19m:53s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/855.7MB/0KB /s] [0/3422/0 iops] [eta 13d:21h:19m:52s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/862.6MB/0KB /s] [0/3450/0 iops] [eta 13d:21h:19m:51s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/855.2MB/0KB /s] [0/3420/0 iops] [eta 13d:21h:19m:50s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/843.9MB/0KB /s] [0/3375/0 iops] [eta 13d:21h:19m:49s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/856.8MB/0KB /s] [0/3427/0 iops] [eta 13d:21h:19m:47s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/861.9MB/0KB /s] [0/3447/0 iops] [eta 13d:21h:19m:47s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/856.3MB/0KB /s] [0/3425/0 iops] [eta 13d:21h:19m:46s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/844.1MB/0KB /s] [0/3376/0 iops] [eta 13d:21h:19m:45s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/859.8MB/0KB /s] [0/3439/0 iops] [eta 13d:21h:19m:44s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/855.7MB/0KB /s] [0/3422/0 iops] [eta 13d:21h:19m:43s
Jobs: 1 (f=1): [W(1)] [0.0% done] [0KB/864.6MB/0KB /s] [0/3458/0 iops] [eta 13d:21h:19m:42s
]
```

5. 此时观察到 node-01 数据恢复速度立刻降到 100 MiB/s 以内

6. 虚拟机 vm-01 中停止 Fio 命令，观察到 node-01 数据恢复速度从 100 MiB/s 逐步提高至 500 MiB/s

根据测试过程可以得出：

- 集群没有业务 I/O 时，服务器 node-01 数据恢复速度达到上限值 500MiB/s 左右
- 发起业务 I/O（虚拟机运行 Fio 模拟业务 I/O），恢复速度从 500 MiB/s 下降至 100 MiB/s
- 停止业务 I/O（虚拟机取消运行 Fio 命令），恢复速度从 100 MiB/s 逐步上升到 500 MiB/s

总结

集群异常触发数据恢复时，SmartX 分布式存储优先确保业务 I/O 正常使用，并将数据恢复 / 迁移速度调整至合理数值，同时满足业务优先和数据安全的要求，避免数据恢复占用大量集群性能，导致业务没有性能可用，引发业务异常。

SmartX 在分布式存储软件设计之初已充分考虑硬件故障带来的风险和运维复杂度，并通过弹性副本恢复策略、磁盘异常（磁盘不健康、亚健康、寿命不足等）处理机制等功能自动智能地处理硬件故障，确保硬件故障时业务稳定性，同时进一步降低运维操作的复杂度，减少运维人员的工作量。

其他功能后续会有介绍。

Vhost | SPDK Vhost-user 如何帮助超融合架构实现 I/O 存储性能提升

点击链接阅读原文：SPDK Vhost-user 如何帮助超融合架构实现 I/O 存储性能提升

要点总结

SPDK vhost 技术基于 Virtio 半虚拟化方案规范发展衍生，优化 I/O 链路，用于加速 QEMU 中半虚拟化的 Virtio 设备 I/O 性能。

Virtio 后端 Device 用于具体处理 Guest 的请求，负责 I/O 的响应，把 I/O 处理模块放在 QEMU 进程之外去实现的方案称为 vhost。

在超融合架构下采用 vhost-user 方案进行存储加速实现，其优势在于消除 Guest 内核更新 PCI 配置空间和 QEMU 捕获 Guest 的 VMM 陷入所带来的 CPU 上下文开销，实现用户态进程间内存共享，优化数据复制效率。

在 3 节点超融合架构集群中，vhost-user 基于单节点 4k iodepth=128 的 IOPS 性能与 Latency 性能均优于 Virtio。

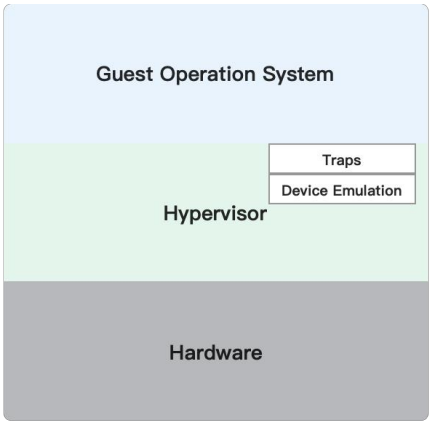
背景介绍

本文主要介绍使用 SPDK vhost-user 技术，来加速 KVM 虚拟机中 virtio-blk/virtio-scsi 存储设备的 I/O 性能，并结合架构场景展开说明，让读者对这项技术带来的特性提升有更直观的了解。

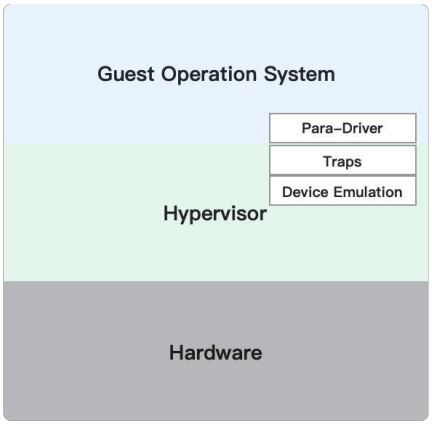
首先我们先看看当前主流的 I/O 设备虚拟化方案：

- QEMU 纯软件模拟，利用软件模拟 I/O 设备提供给虚拟机使用。
- Virtio 半虚拟方案，规范了前后端模型，在虚拟机（Guest OS）中使用 frontend 驱动（virtio Drive），在 Hypervisor（QEMU）中使用 backend 设备（virtio Device）提供 I/O 能力，通过减少内存复制次数和 VM 陷入次数，提升 I/O 性能，这种方案需要安装 Virtio 驱动。

本文的主角 vhost 技术是基于 Virtio 规范发展衍生出来，优化 I/O 链路，可以用来加速 QEMU 中半虚拟化的 Virtio 设备 I/O 性能。



QEMU 软件模拟



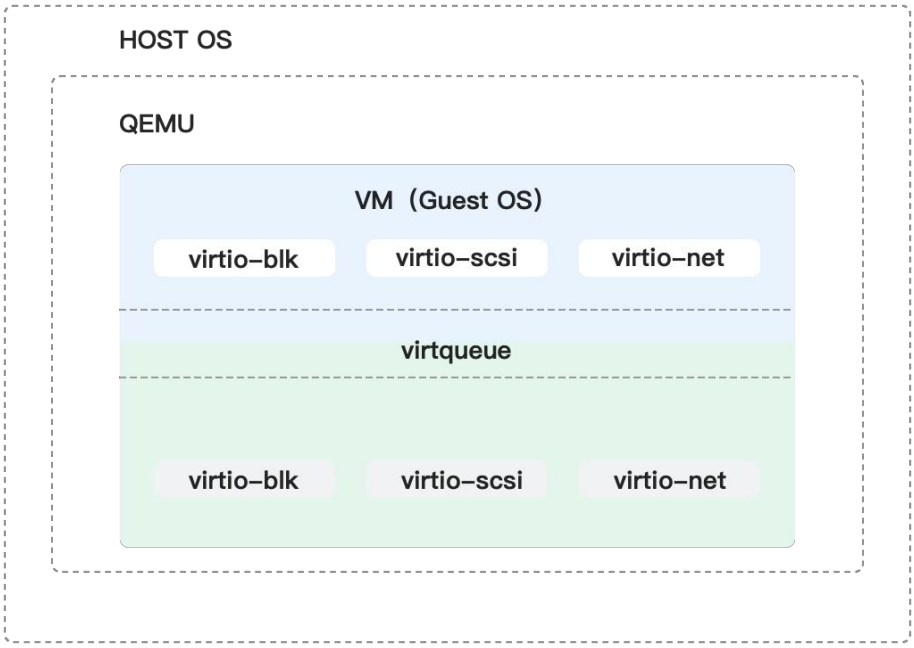
半虚拟化 Virtio 设备

Virtio 介绍

Virtio 基于 Vring 的 I/O 通讯机制，相比 QEMU 的纯软件模拟，有效降低了 I/O 延迟，具有性能优势。这也是 Virtio 普及的原因，各个厂商的半虚拟化 I/O 设备实现方式开始变得统一。

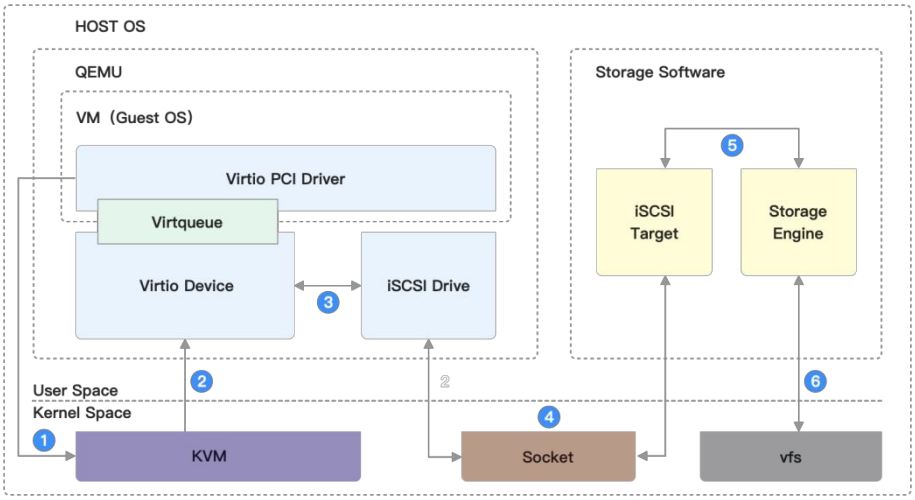
在 QEMU 中，Virtio 设备是为 Guest 操作系统模拟的 PCI/PCIe 设备，遵循 PCI 规范，具有配置空间、中断配置等功能。Virtio 注册了 PCI 厂商 ID（0x1AF4）和设备 ID，不同的设备 ID 代表不同的设备类型，例如面向存储的 virtio-blk（0x1001）和 virtio-scsi 设备 ID（0x1004）。

Virtio 由三部分组成，前端是驱动层，位于 Guest 系统内部，中间是虚拟队列（virtqueue），负责数据传输和命令交互，后端设备层，用于具体处理 Guest 发送的请求。



下面我们一起来看一下超融合架构下，基于 Virtio-blk 的数据路径是什么样子的。

超融合架构是一种 IT 基础架构解决方案，将计算、存储和网络资源整合在一个统一系统（计算服务器）中。超融合基础架构由虚拟化、分布式存储和软件定义网络组成。利用分布式架构实现集群的可靠性和容错性，并通过将计算与存储集成在一起，灵活部署在通用标准硬件上，来降低数据中心复杂性和占用空间，并支持更多的现代工作负载。



1. Guest 发起 I/O 操作，Guest 内核 virtio 驱动写 PCI 配置空间，触发 VM EXIT，返回到 Host KVM 中（通知 KVM）；
2. QEMU 的 vCPU 线程从 KVM 内核态回到 QEMU，让 QEMU Device 来处理 Virtio Vring 请求；

3. QEMU 通过 iSCSI Drive 发起存储连接（iscsi over tcp to localhost）；
4. 通过 Socket 将请求连接到存储进程提供的 iSCSI Target；
5. 存储引擎接收请求并进行 I/O 处理；
6. 存储引擎发起对本地存储介质的 I/O；
7. I/O 操作结束，通过上述逆过程返回至 Virtio 后端 Device，QEMU 会向模拟的 PCI 发送中断通知，从而 Guest 基于该中断完成整个 I/O 流程。

QEMU 通过本地 Socket 连接存储进程，数据流从用户态到内核态再到用户态（数据复制开销），同时 iSCSI 协议层也存在性能消耗，如果存储进程可以直接接收处理本地 I/O，就可以避免这些问题带来的损耗，实现 Virtio Offload to Storage Software。

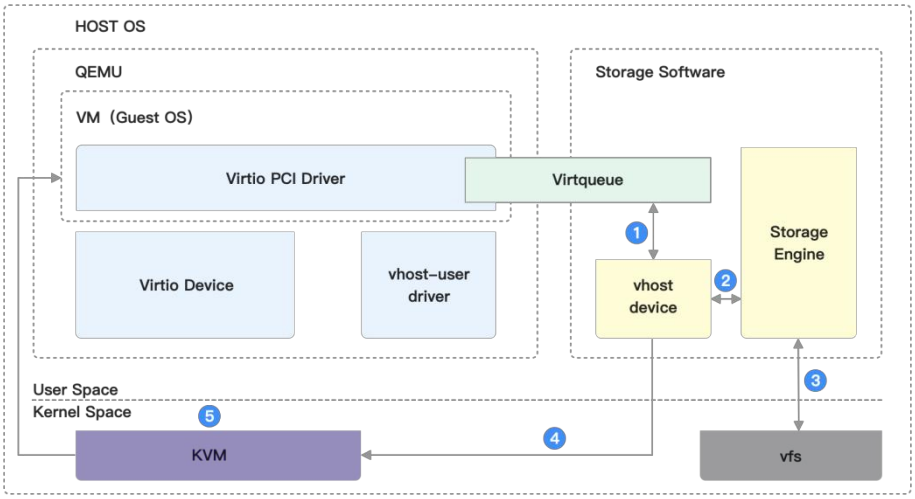
vhost 加速

如前所述，Virtio 后端 Device 用于具体处理 Guest 的请求，负责 I/O 的响应，把 I/O 处理模块放在 QEMU 进程之外去实现的方案称为 vhost。由于我们需要实现的优化目标是在两个用户态进程之间（超融合架构），所以采用 vhost-user 方案进行存储加速实现（vhost-kernel 方案主要是将 I/O 负载卸载到内核完成，所在不在本文讨论）。

vhost-user 的数据平面处理主要分为 Master 和 Slave 两个部分，其中 Master 为 Virtqueue 的供应方，一般由 QEMU 作为 Master，存储软件作为 Slave，负责消费 Virtqueue 中的 I/O 请求。

vhost-user 方案优势

- 消除 Guest 内核更新 PCI 配置空间，QEMU 捕获 Guest 的 VMM 陷入所带来的 CPU 上下文开销（后端处理线程采用轮询所有 virtqueue）；
- 用户态进程间内存共享，优化数据复制效率（零拷贝）。



1. 当 Guest 发起 I/O 操作后，存储软件通过 Polling 机制感知新的请求动作，从 virtqueue 获取数据；
2. 存储引擎接收请求并进行 I/O 处理；
3. 存储引擎发起对本地存储介质的 I/O；
4. I/O 操作完成后，由 vhost device 发送 irqfd（eventfd）通知到 KVM；
5. KVM 注入中断通知 Guest OS 完成 I/O。

注：前端 vhost driver 与后端 vhost device 之间的控制类信息传递通过 UNIX Domain Socket 文件实现。

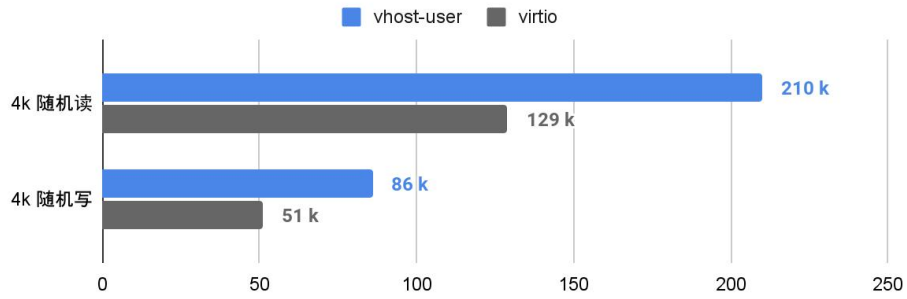
介绍完理论过程，让我们看一组存储性能对比 virtio vs vhost-user（单节点性能）。

I/O 模型	iodepth	指标	Virtio	vhost-user
4k randread	1	Latency	137 us	96.1 us
	128	IOPS	129.17 k	210.08 k
4k randwrite	1	Latency	205 us	125 us
	128	IOPS	51.87 k	86.54 k
256k read	1	BW	391 MiB	483 MiB
	128	BW	2805 MiB	4073 MiB
256k write	1	BW	417 MiB	500 MiB
	128	BW	1934 MiB	1999 MiB

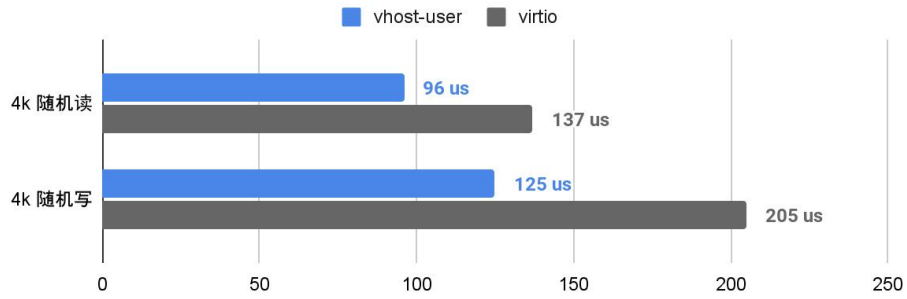
测试数据基于 3 节点集群（超融合架构，集群存储网络未开启 RDMA），节点硬件配置如下

- Intel Xeon Gold 5122 3.6GHz
- 8*16G DDR4 2666MHz
- 2*Intel P4610 1.6T NVMe
- Mellanox CX-4 25Gbps

通过图表对比单节点 4k iodepth=128 下的 IOPS（数值越高越好）性能差异：



通过图表对比单节点 4k iodepth=1 下的 Latency（数值越低越好）性能差异：



总结

通过理论和真实的性能数据介绍，可以看出，相比 Virtio 方案，通过 vhost 技术实现了更加突出的存储性能表现，但作为企业级产品交付，需要考虑到企业级功能，例如存储软件异常下的 vhost 重连/切换、虚拟机的内存热添加等实际场景，希望这篇文章对于读者理解 vhost 以及适配超融合架构有所帮助。

Boost｜性能接近翻倍！利用 Boost 技术优化 SmartX 超融合信创平台承载达梦数据库性能详解

点击链接阅读原文：性能接近翻倍！利用 Boost 技术优化 SmartX 超融合信创平台承载达梦数据库性能详解

要点总结

本次测试使用 BenchmarkSQL 基于 TPC-C 基准执行测试，对比达梦 DM8 数据库在裸金属服务器（分别基于 SATA SSD 和 NVMe SSD）、未进行 Boost 模式优化的 SmartX 超融合信创平台和优化后的超融合平台上的性能。

Boost 模式下的优化方式包括：BIOS 参数优化、启用 Boost 模式和 RDMA 网络优化、虚拟机设置优化（包括开启 CPU 独占功能和调节虚拟磁盘存储策略为厚置备）、虚拟机操作系统参数优化（利用 CPU 多核特性进行网络优化），以及数据库相关优化。

在未做优化时，基于信创架构的 SmartX 超融合运行达梦数据库性能是裸金属服务器（基于 SATA SSD）的 80%。而通过 Boost 模式进行调优后，数据库性能提升近一倍，达到裸金属服务器（以 SATA SSD 为介质）的 1.77 倍，NVMe 裸盘的 88%。

目前，各关键行业都在加速信创转型，并从最初的测试业务、边缘生产业务，到尝试承载重要工作负载，逐渐进入“深水区”。大多数用户对信创生态构成方案的性能和特性不是十分了解，对于核心业务的信创转型难免心中有疑问和顾虑：信创数据库配合信创硬件表现是什么水平？在虚拟化或者超融合平台上运行表现如何？软硬件是否存在调优的空间？

基于以上需求，SmartX 方案中心围绕信创数据库产品（达梦 DM8）在 SmartX 超融合信创平台（基于鲲鹏芯片的信创服务器）上进行性能测试，并利用独有的 Boost 加速技术对数据库进行调优。测试结果表明，结合数据库的参数调整，Boost 模式下 SmartX 超融合信创平台支撑的达梦数据库获得了接近 100% 的性能提升。

测试环境

达梦 DM8 数据库

DM8 是达梦公司在总结 DM 系列产品研发与应用经验的基础上，坚持开放创新、简洁实用的理念，推出的新一代自研数据库。DM8 吸收借鉴当前先进技术思想与主流数据库产品的优点，融合了分布式、弹性计算与云计算的优势，对灵活性、易用性、可靠性、高安全性等方面进行了大规模改进，多样化架构可充分满足不同场景需求，支持超大规模并发事务处理和事务-分析混合型业务处理，动态分配计算资源，实现更精细化的资源利用、更低的成本投入。

信创硬件（鲲鹏芯片）

本次测试采用的信创服务器是神州鲲泰 KunTai R722。

测试服务器配备的是鲲鹏 920 系列 CPU，是目前业界领先的 ARM-based 处理器。该处理器采用 7nm 制造工艺，基于 ARM 架构授权，由华为公司自主完成设计。

鲲鹏 920 5250 一个明显特征是：CPU 核心（cores）比较多，单路 CPU 拥有 48 个核心，2 路 CPU 共 96 个核心（常用的 Intel 至强系列 CPU 单路大多在 20 核左右）。因此，后期性能优化的其中一个重点是如何更好利用多核的优势。

服务器详细配置如下：

配件	配置信息
CPU	HUAWEI Kunpeng 920 5250 48 cores 2.6GHz x2
内存	32GB x 8
缓存盘	NVMe SSD 3.2TB x 2
数据盘	10TB x 4
启动盘	480G x2
存储网卡	25GbE x 2
管理/业务网卡	HUAWEI TM210 x 2

SmartX 超融合信创云基础设施

志凌海纳 SmartX 以超融合软件 SMTX OS 为核心，提供自研、解耦、生产就绪的超融合信创云基础设施产品组合，已助力众多行业用户构建轻量信创云底座。SMTX OS 是构建超融合平台的核心软件，内建原生服务器虚拟化 ELF 和分布式块存储 ZBS，可选配双活、异步复制、备份与恢复、网络与安全等高级功能，结合认证列表内的商用服务器，即可快速构建强大而敏捷的云化资源池。欲深入了解，请阅读：一文了解超融合信创云基础设施。

Boost 模式是 SMTX OS 的高性能模式。该模式通过内存共享技术缩短虚拟机的 I/O 路径，从而提升虚拟机性能，降低 I/O 访问延迟。Boost 模式通常会搭配 RDMA 网络一起启用，可最大化提升存储性能。如希望进一步了解 Boost 模式的实现原理，请阅读：SPDK Vhost-user 如何帮助超融合架构实现 I/O 存储性能提升。

测试方法

本次测试使用 BenchmarkSQL 基于 TPC-C 基准执行测试，对比达梦 DM8 数据库在裸金属服务器（分别基于 SATA SSD 和 NVMe SSD）、未进行 Boost 模式优化的 SmartX 超融合信创平台和优化后的超融合平台上的性能。

测试标准与参照

TPC-C 测试

TPC-C 是一个业界公认的事务处理性能基准测试。它是 Transaction Processing Performance Council（TPC）发布的标准基准测试之一，用于测试在线事务处理（OLTP）系统的性能。TPC-C 测试基于一个虚拟的在线订单处理应用程序，它包括了一系列的事务操作，如客户订单、库存管理、交付处理等。TPC-C 测试结果以“每分钟事务处理量”（TPM）为单位进行度量。

BenchmarkSQL 是一款可以使用 TPC-C 测试规范来运行基准测试的工具。具体来说，BenchmarkSQL 可以使用 TPC-C 测试规范中定义的事务操作和数据结构，来模拟一个 TPC-C 测试环境，并对数据库系统进行性能测试。因此，BenchmarkSQL 可以被认为是TPC-C 测试的一种实现方式。

本次测试使用 BenchmarkSQL 基于 TPC-C 基准执行测试，以便更客观地评价超融合信创平台上数据库的性能表现。

本次测试使用的软件版本如下：

软件名称	版本	备注
SMTX OS	5.0.5	SmartX 超融合软件
CloudTower	2.6.0	SmartX 管理平台
OpenEuler	openEuler-22.03-LTS	物理机/虚拟机操作系统
DM8	dm8_20230104_HWarm_centos7_64.iso	数据库软件
BenchmarkSQL	benchmarksql-5.0rc2-westone-v1.2	压测软件

测试参照

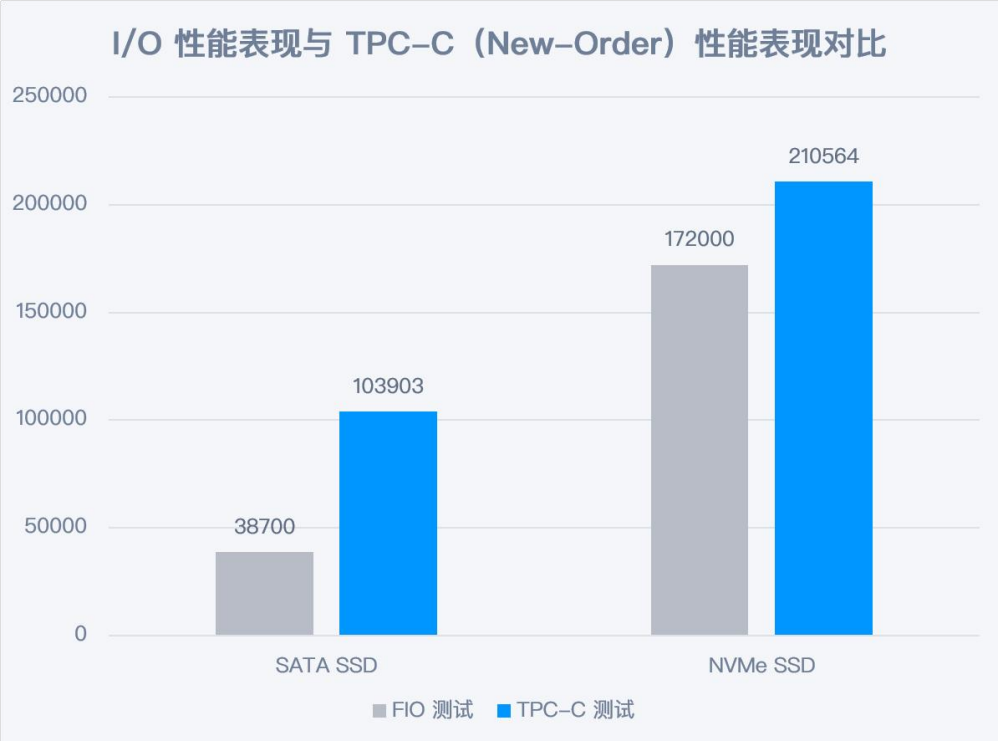
用户以往或许有了解过一些数据库的 TPC-C 测试数据，但这些数据大多基于 x86 架构服务器环境，对于信创芯片的 TPC-C 表现未必是十分了解的。考虑到这一点，我们首先在裸金属服务器（基于鲲鹏芯片）上直接部署达梦数据库软件（物理机部署），然后执行一组 TPC-C 测试作为参照，以便与后续 SmartX 超融合的表现进行对比。

不同存储介质下的性能表现

由于数据库对磁盘 I/O 性能比较敏感，在测试场景中，我们使用了两款不同类型的 SSD 作为存储介质，分别进行测试。首先，通过 FIO 测试工具对 SSD 分别执行 I/O 压力测试（8k 随机读写），作为两款 SSD 的 I/O 基准性能，结果如下：

I/O 模型	SATA SSD（单盘 ext4 ）	NVMe SSD（单盘 ext4 ）
8k-randwrite	38700	172000
8k-randread	47000	176000

然后，我们在两种 SSD 上分别运行达梦数据库的 TPC-C 测试（100 warehouse，200 terminals），结果如下：



注明：TPC-C 测试中取 NewOrder 的值作为测试结果，后续出现的结果亦然，不再赘述。

两组数据是在同一服务器中测试得到的，可以得到以下结论：TPC-C 测试结果随着存储 I/O 能力的增长而增长，但两者不完全是等比关系（其中 NVMe SSD 的 I/O 写入能力相比 SATA SSD 提升了 340%，然而 TPC-C 只提升大约 102%）。

CPU NUMA Group 对性能影响

测试分为两组：

- A 组：数据库程序通过 numactl 命令绑定到同一颗 CPU 的 2 个 NUMA 组（48 核）。
- B 组：数据库不绑定 CPU，利用服务器上所有 CPU 核心（96 核）。

测试结果如下：

测试	48 cores（同 1 个 CPU 中 2 个 NUMA）	96 cores（跨 2 个 CPU 中 4 个 NUMA）
TPC-C	137045	110738.74

测试结果有点出乎所料：A 组（48 核）要比 B组（96 核）的性能更好。一般情况下，更多的 CPU 内核对数据库的性能的影响理论上应该是正向的。但这个测试中有两个因素影响了该结果。

- 达梦数据库的工作线程参数最大支持 64（官方要求工作线程与 CPU 核心数一样），无法充分利用全部 96 个 CPU 内核。
- 数据库在跨 CPU NUMA 组下工作，内存访问效率下降。

考虑到数据库的特点以及 NUMA 的影响，后续超融合平台测试中的虚拟机配置采取 48 vCPU （并确保在同一个 CPU 中）的配置进行测试。

测试过程

测试条件

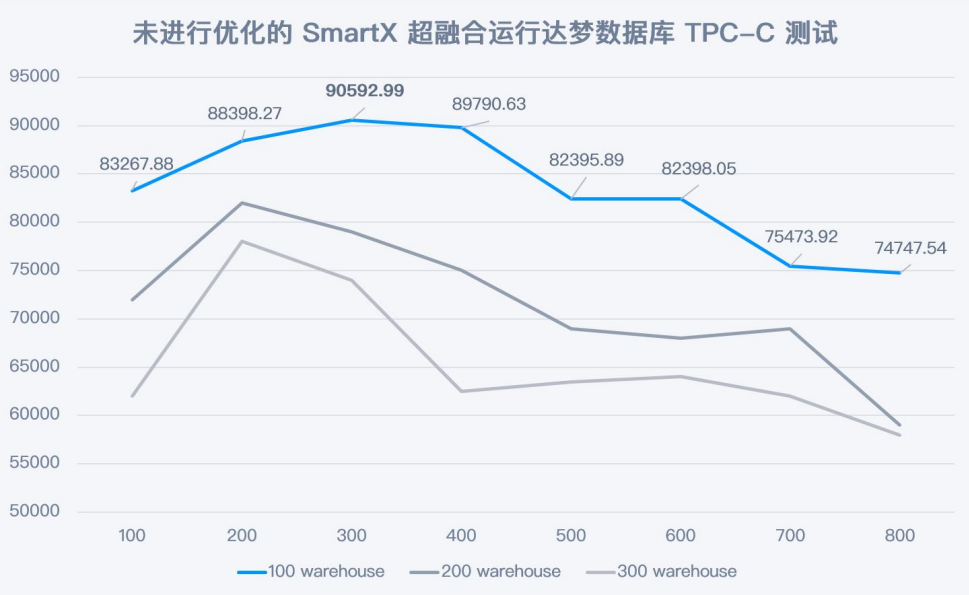
虚拟机资源配置

配置	SMTX OS 虚拟机（DM8）	BenchmarkSQL 虚拟机
CPU	48 vCPU	4 vCPU
内存	96GB	16GB
数据盘	200GB（2 副本）	120GB
网卡	1GbE	1GbE

TPC-C 测试集

- 调整 terminal 数值，以验证数据库在不同并发访问压力下的性能表现。共执行 100 ~ 800 共 8 组 terminals 测试。
- 调整 warehouses 数值，以验证数据库在不同数据集大小下的性能表现。共执行 100 ~ 300 共 3 组 warehouse 测试。每组 warehouse 结合上述不同的 terminal 数量，共执行 24 组测试。

测试一：未做任何优化的 SmartX 超融合运行达梦数据库性能表现



TPC-C NewOrder 最大值在 100 warehouse 下 300 terminal 下产生，每分钟完成 90592 笔新订单（NewOrder）。在没有任何优化的情况下，数据库表现并不算理想，是裸金属服务器（基于 SATA SSD）部署性能的 80%。

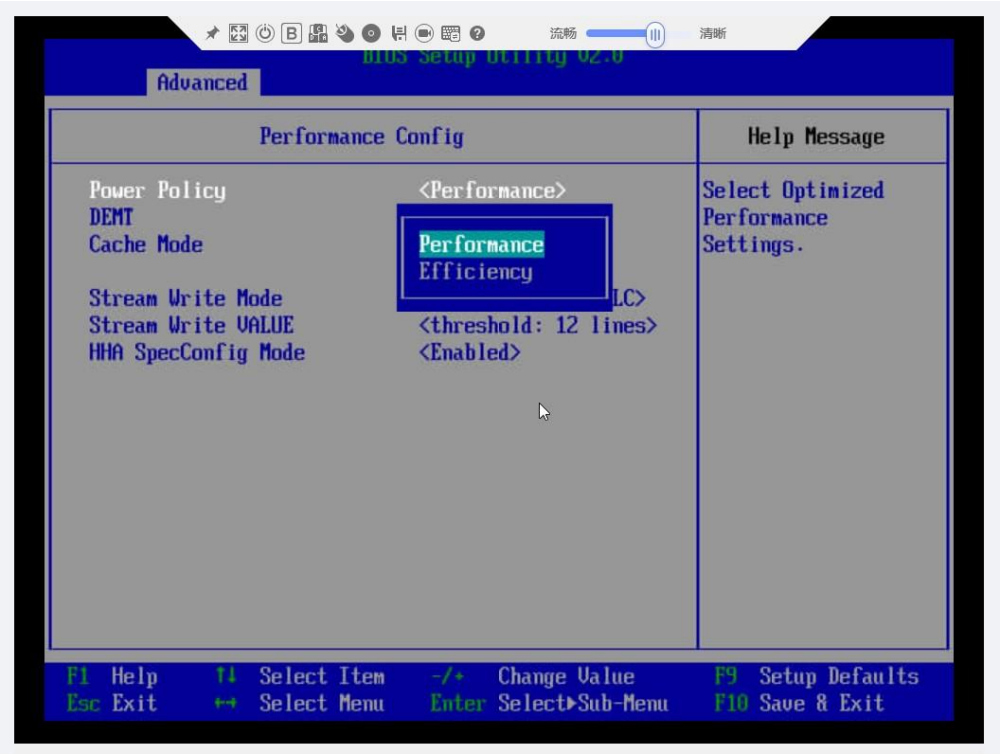
测试二：经过 Boost 模式调优的 SmartX 超融合运行达梦数据库性能表现

SMTX OS Boost 模式下的优化手段

下面将展示在 SMTX OS Boost 模式下，如何提升达梦数据库 TPC-C 测试的性能表现。

BIOS 参数优化

开启 Boost 模式之前，要求在服务器 BIOS 中将电源策略从“节能模式”，切换为“性能模式”，以确保服务器的功率在最佳性能状态。



启用 Boost 模式和 RDMA 网络优化

- 在部署 SMTX OS 集群的第 1 步：集群设置阶段，勾选启用 Boost 模式复选框。
- 在部署 SMTX OS 集群的第 5 步：配置网络阶段，在为存储网络创建虚拟分布式交换机时，通过单击启用 RDMA 按钮，开启集群的 RDMA 功能。

虚拟机设置优化

- 开启 CPU 独占功能

创建数据库虚拟机时，勾选 CPU 独占功能。后台将自动对虚拟机的 vCPU 进行 NUMA 绑定，使得虚拟机获得更佳的性能。

- 虚拟磁盘存储策略调节为厚置备

将数据库所在的虚拟磁盘从默认的精简制备设置为厚置备，将小幅度提升 I/O 性能，同时降低 CPU 占用。

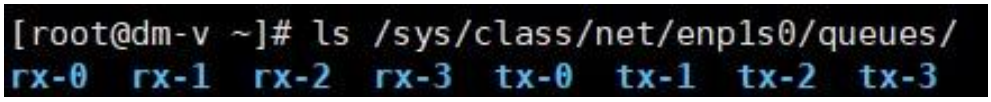
虚拟机操作系统参数优化

- 利用 CPU 多核特性进行网络优化

由于 TPC-C 测试是通过 SMTX OS 集群外部的 benchmarkSQL 虚拟机发起请求，通过网络压测数据库，想要充分发挥 Boost 模式的效果，网络优化是非常必要的。基于鲲鹏 CPU 多核的优势，将网络队列和中断的任务分配到不同的 CPU 核中执行，可减少资源争抢的情况，并有效提升网络传输性能。

方式一：为网卡队列指定 CPU 核

a. 使用 ls /sys/class/net/enp1s0/queues/ 查看网卡队列情况：



在测试环境中，可以看到网卡对应的接收队列和发送队列各有 4 组，具体按实际情况而定。

b. 分别为多组网卡队列指定 CPU 核，命令如下：

```
echo 1 > /sys/class/net/enp1s0/queues/rx-0/rps_cpus
echo 2 > /sys/class/net/enp1s0/queues/rx-1/rps_cpus
echo 4 > /sys/class/net/enp1s0/queues/rx-2/rps_cpus
echo 8 > /sys/class/net/enp1s0/queues/rx-3/rps_cpus
echo 16 > /sys/class/net/enp1s0/queues/tx-0/xps_cpus
echo 32 > /sys/class/net/enp1s0/queues/tx-1/xps_cpus
echo 64 > /sys/class/net/enp1s0/queues/tx-2/xps_cpus
echo 128 > /sys/class/net/enp1s0/queues/tx-3/xps_cpus
```

其中 echo 1 > /sys/class/net/enp1s0/queues/rx-0/rps_cpus 代表将 CPU 1 绑定到 rx-0 号队列，其中 CPU 0、1、2、3 四个 CPU 对应的值分别是 1（20）、2（21）、4（22）、8（23）。

方式二：为网卡中断指定 CPU 核

a. 使用以下命令查看网卡中断情况：

```
cat /proc/interrupts | grep virtio0|cut –f 1 –d ":"
```

b. 修改配置文件，使得 irqbalance 服务不再调度这几个中断。

通过 vim /etc/sysconfig/irqbalance 修改文件，将以下参数改为：

```
IRQBALANCE_ARGS=--banirq=91-99
```

c. 手工为每个网卡中断分配 CPU 核，如下：

```
echo 40 > /proc/irq/91/smp_affinity_list
echo 41 > /proc/irq/92/smp_affinity_list
echo 42 > /proc/irq/93/smp_affinity_list
echo 43 > /proc/irq/94/smp_affinity_list
echo 44 > /proc/irq/95/smp_affinity_list
echo 45 > /proc/irq/96/smp_affinity_list
echo 46 > /proc/irq/97/smp_affinity_list
echo 47 > /proc/irq/98/smp_affinity_list
echo 48 > /proc/irq/99/smp_affinity_list
```

执行上述两个部分的网络优化，可以明显提升 TPC-C 测试中的网络性能，其中发送速度的峰值最高提升 17.6%，接收速度峰值最高提升 27.1%。

数据库相关优化

– 调整数据库日志参数，充分发挥 I/O 并发能力

达梦 DM8 的数据库日志文件（logfiles）的数量默认是 2 个。由于 SMTX OS 开启 Boost 模式后，获得更强的 I/O 并发能力，通过增加日志文件数量可充分挖掘存储的并发性能。测试中，将日志文件数量从 2 个增加到 8 个，性能在全部场景中都能获得明显提升。结果如下图：

增加日志文件后，100 warehouse 场景下的性能提升的比例范围是 21%–35%（如图）。在 300 warahouse 场景下，最高提升 47%（有相关测试数据，未展示图表）。

– 调整 DM8 数据库内存缓存区参数，优化缓存命中率

由于数据库所在的虚拟机分配的内存是 96GB，因此将内存池参数和内存目标参数设置为 90GB（预留 6G 给操作系统使用）。通过 /dm8/data/DAMENG/dm.ini 修改数据库参数文件，可调整相关参数。

```
MEMORY_POOL = 90000              #Memory Pool Size In Megabyte
MEMORY_TARGET = 90000            #Memory Share Pool Target Size In Megabyte
```

DM8 数据库中有四种类型的数据缓冲区，分别是 NORMAL、KEEP、FAST 和 RECYCLE。

其中 NORMAL 缓冲区对应的 BUFFER 参数建议尽可能大，需确保命中率较高（90% 以上）。在本次测试中调整 BUFFER 缓冲区大小为 70GB，BUFFER_POOLS 数量为 48（保持与 CPU 核数一致）。

```
BUFFER = 70000                    #Initial System Buffer Size In Megabytes
BUFFER_POOLS = 48                 #number of buffer pools
```

此外，RECYCLE 缓存区供临时表空间使用，因此也要调整相关参数。这里调整 RECYCLE 缓冲区大小为 12GB，RECYCLE_POOLS 数量为 48（保持与 CPU 核数一致）。

```
RECYCLE = 12000                   #system RECYCLE buffer size in Megabytes
RECYCLE_POOLS = 48                #Number of recycle buffer pools
```

最后需要根据 CPU 的核数，调整数据库的工作线程，在这里将工作线程调整为 48（保持与 CPU 核数一致）。

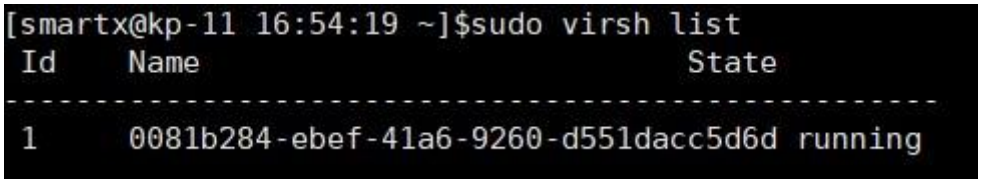
```
WORKER_THREADS = 48               #Number Of Worker Threads
```

**注明：修改 dm.ini 文件参数后，必须重启数据库才能生效。*

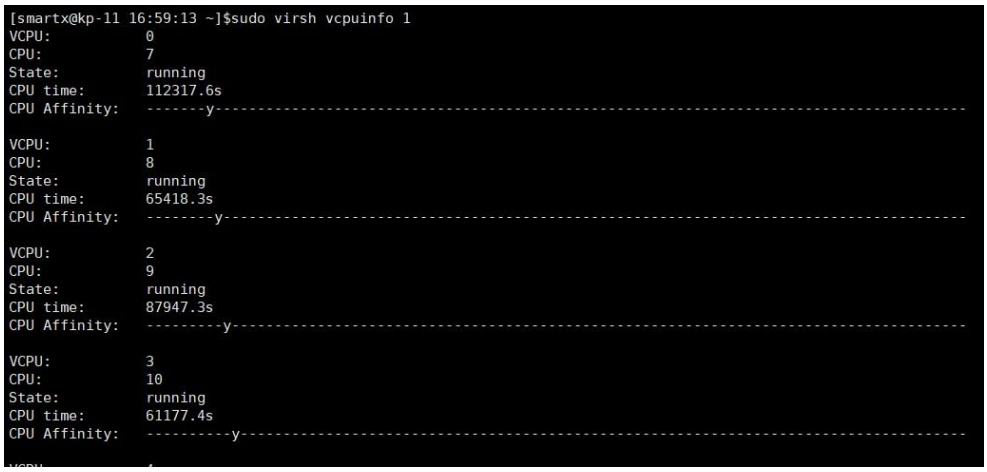
- 数据库程序设置 NUMA 绑定

DM8 数据库程序可通过绑定 NUMA 限制程序在同一个物理 CPU 内，提升内存访问效率，从而提升数据库性能。

a. ssh 登陆 SMTX OS 节点（数据库虚拟机所在节点），执行 sudo virsh list 查看虚拟机的 ID 号。



b. 根据虚拟机 ID 执行 sudo virsh vcpuinfo 1，查看 vCPU 核与物理 CPU 核的对应关系 。



c. 运行 sudo numactl —hardware 查看 NUMA 亲和性关系。

```
[smartx@kp-11 16:59:15 ~]$sudo numactl --hardware
available: 4 nodes (0-3)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
node 0 size: 64873 MB
node 0 free: 1429 MB
node 1 cpus: 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
node 1 size: 65467 MB
node 1 free: 92 MB
node 2 cpus: 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
node 2 size: 65467 MB
node 2 free: 4215 MB
node 3 cpus: 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
node 3 size: 64423 MB
node 3 free: 5412 MB
node distances:
node   0   1   2   3
  0:  10  12  20  22
  1:  12  10  22  24
  2:  20  22  10  12
  3:  22  24  12  10
```

d. 通过 numactl 命令启动数据库，实现绑定 NUMA 目的：

```
numactl –C 0–16,17–40,41–47 sh DmServiceDMSERVER start
```

完成上述所有优化操作后，重新执行 TPC-C 测试并与优化前的测试数据进行对比。

SMTX OS Boost 模式优化后性能大幅提升

通过开启 SMTX OS Boost 模式以及配套相关优化设置后，数据库性能在每个测试场景下的提升都是非常明显的，几乎都是翻倍提升。详细数据如下：

100 warehouse 场景

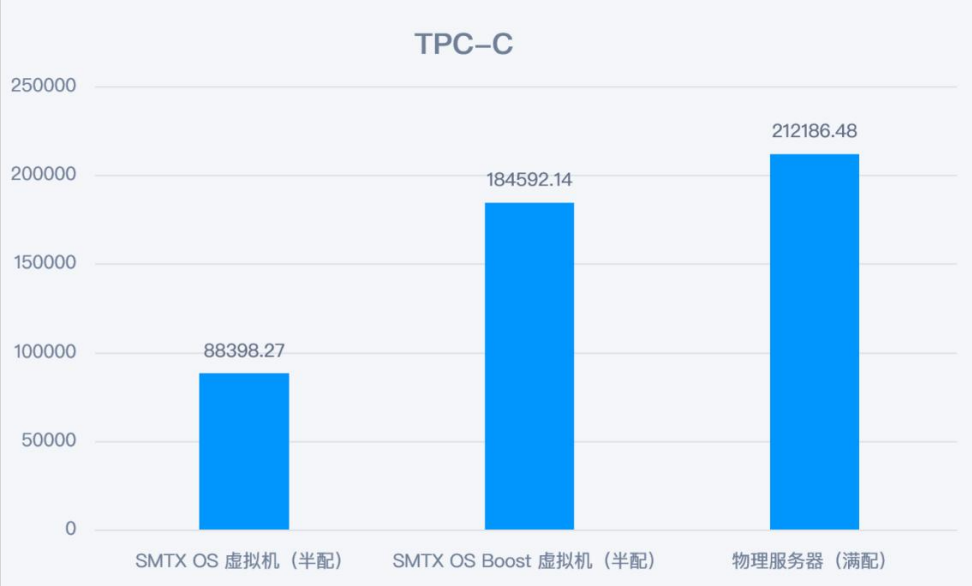
200 warehouse 场景

300 warehouse 场景

测试结论

通过 Boost 模式以及相关优化，在 SmartX 超融合信创平台上运行达梦数据库可获得以下收益：

- 性能是裸金属服务器（以 SATA SSD 为介质）的 1.77 倍，并已接近裸金属服务器（以 NVMe SSD 为介质），达到 NVMe 裸盘性能的 87.6%。
- SMTX OS 提供了 2 副本数据冗余保护（而裸盘虽然性能好，但无数据冗余保护）。
- SMTX OS 只占用了单台服务器主机的 CPU 和内存资源的 50%，意味着剩下的资源可以运行更多的业务，有效提升资源的利用率。



**满配：数据库使用单台服务器所有 CPU 核以及内存资源，96 CPU，256G 内存。*

**半配：数据库使用单台服务器部分 CPU 核以及内存资源，48 CPU，96G 内存。*

本次测试不仅为读者展示了信创数据库在超融合信创平台上的真实表现，也验证了 SmartX 超融合 Boost 模式对数据库的性能优化效果。欲了解更多 SmartX 超融合在数据库场景下的性能表现，请阅读：SmartX 超融合金融行业数据库支撑评测合集与长期落地案例综述。

后记

从测试结果上看，SmartX 超融合平台凭借杰出的 I/O 性能及相关针对性优化，可明显提升达梦数据库 TPC-C 性能测试表现。由于上述测试模型是基于模拟生产场景，数据库的参数是注重 I/O 真实落盘（写入存储介质）。大家可能会有一个疑问：是否能通过内存缓存，以不落盘的方式进一步提升数据库 TPC-C 性能测试表现？

答案是可以的。一方面，可以调整数据库参数，使得数据库减少 I/O 落盘，同时扩大数据库虚拟机的内存，通过大量使用内存加速数据库响应能力。另一方面，由于原来的测试模型是由外部压力虚拟机经过千兆网络发出请求，最后到达数据库虚拟机进行处理，中间会经过多个环节：压力机的虚拟网卡→虚拟交换机→物理网卡→物理交换机→物理网卡→虚拟交换机→数据库机的虚拟网卡。整个网络传输的环节会带来一定的性能损耗。我们可以模拟屏蔽网络传输影响，额外做一个测试作为参考：将压力程序安装在数据库虚拟机本地，使得请求压力不经过网络，直接在数据库虚拟机内部发出，并在虚拟机内部处理。

经过上述一系列变更后，我们再次执行 TPC-C 测试，结果如下：

当 warehouse= 100，不同并发数量场景下 TPC-C 测试的 tpmc（NewOrder） 值：

warehouse	terminal（并发量）	Tpmc
100	100	468113
	200	423574.6
	300	366962.5
	400	391429.2
	500	383533.6
	600	331174.3
	700	351382.2
	800	320975.7

当 warehouse= 200，不同并发数量场景下 TPC-C 测试的 tpmc（NewOrder） 值：

warehouse	terminal（并发量）	Tpmc
200	100	361277.8
	200	253534.9
	300	253742.4
	400	202263.2
	500	317091.1
	600	305937.4
	700	302694.3
	800	297403.4

当 warehouse= 300，不同并发数量场景下 TPC-C 测试的 tpmc（NewOrder） 值：

warehouse	terminal（并发量）	Tpmc
300	100	250258
	200	221147.5
	300	218891
	400	237003.5
	500	228237
	600	237272
	700	211769.1
	800	238786.8

从测试结果可以看到，TPC-C 性能有明显提升，在 100 warehouse/100 terminal 场景下可达 468113 TPM（最高）。但这种数据库配置模型由于有大量数据缓存在内存中，I/O 没有及时落盘，如系统遭遇突然断电，有可能导致数据库不一致的情况发生，所以生产环境中的数据库一般很少采用（除非是只读数据库），测试结果仅作为参考。

GPU 直通 & vGPU｜超融合为 GPU 应用场景提供高性能支持

点击链接阅读原文：GPU 直通 & vGPU：超融合为 GPU 应用场景提供高性能支持

近些年，随着大数据、区块链、AI 等技术快速兴起，越来越多的企业在生产场景中使用人工智能、机器学习、高性能计算等前沿应用，加速企业现代化发展。由于这些应用普遍需要强大的并行计算能力，相较于擅长处理串行任务的 CPU，具备更多核心、可并发处理多个任务的 GPU 成为了支撑高性能应用的不二之选，企业对 GPU 算力的需求也水涨船高。

然而，GPU 芯片在国内市场上非常稀缺，且价格高昂，为了尽可能提高 GPU 资源的利用效率，不少企业选择将 GPU 资源虚拟化，或在虚拟化（或超融合）环境中使用 GPU。那么，如何在虚拟化架构上充分发挥 GPU 能力和优势？怎样以一套架构支持资源的灵活发放，在满足多种 GPU 应用计算需求的同时，为应用提供稳定、高性能存储支持？

针对企业的 GPU 应用场景需求，SmartX 在全新发布的超融合软件 SMTX OS 5.1 版本中新增了原生虚拟化 ELF 平台 GPU 直通和 vGPU 支持能力。目前，SmartX 超融合可为 ELF 集群提供完整的 GPU 支持能力，并为多种虚拟化环境（包括 VMware ESXi 和 Citrix XenServer）的 GPU 应用提供高性能的存储资源池，帮助企业实现人工智能、机器学习、图像识别处理、VDI（如三维建模、图像渲染）等多种高性能应用的生产级使用。

要点总结

越来越多的前沿应用需要强大的并行计算能力，具备更多核心、可并发处理多个任务的 GPU 成为了支撑高性能应用的选择。为了尽可能提高 GPU 资源的利用效率，不少企业选择将 GPU 资源虚拟化，或在虚拟化（或超融合）环境中使用 GPU。

SmartX 在全新发布的超融合软件 SMTX OS 5.1 版本中新增了原生虚拟化 ELF 平台 GPU 直通和 vGPU 支持能力，具备快速置备开发环境、灵活的资源发放、高性能的存储支持等优势。

SMTX OS 5.1 GPU 直通和 vGPU 功能

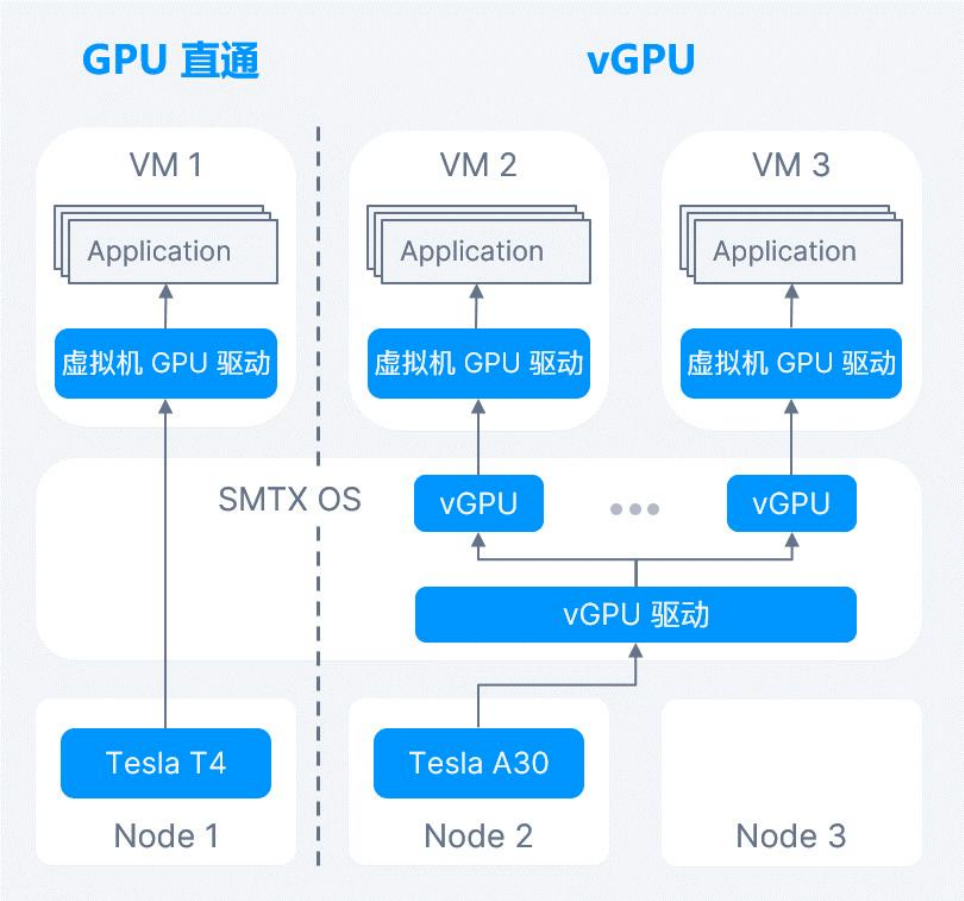
功能特性

为了满足不同用户和应用场景需求，SMTX OS 5.1 支持两种虚拟化环境中 GPU 使用模式：

- GPU 直通：将主机上的物理 GPU 设备透传给虚拟机使用，GPU 的全部资源由一台虚拟机独占。
- vGPU：单个物理的 GPU 切割成多个逻辑的 vGPU，并将 vGPU 分配给虚拟机作为虚拟显卡，可实现多个虚拟机共享 GPU 计算/图形处理能力。

两种模式下，每个主机上可以使用多个不同型号的 GPU 设备*，每个虚拟机可挂载多个 GPU 或 vGPU。

**兼容列表见文末附录。*



GPU 直通

GPU 直通模式是利用 PCIe Pass-through 的技术，将 SMTX OS 主机上的整块 GPU 显卡透传挂载到虚拟机上使用。这种模式适用广泛，可支持大部分的 GPU 卡型号。直通模式具有良好的兼容性，虚拟机识别到 GPU 型号与实际一致，直接安装官方驱动，可无损使用 GPU 的各项特性和功能。同时，由于虚拟机操作系统可直接访问 GPU 设备，绕过了 SMTX OS 超融合操作系统，使得它能获得接近裸金属使用 GPU 的性能。

不过需要注意的是，该模式下一张 GPU 卡不能同时直通给多个虚拟机使用，如果多个虚拟机需要同时使用 GPU，需要在服务器中安装多块 GPU 卡，分别直通给不同的虚拟机使用。另外，拥有直通 GPU 的虚拟机不支持 HA、在线迁移和分段迁移功能。

vGPU

vGPU 模式允许多台虚拟机共享一张 GPU 物理卡资源，进一步提高资源利用率，节约成本。同时，管理员可以按照用户的实际需求分配不同 GPU 资源，例如 ⅛ GPU、¼ GPU 等，使得 GPU 资源分配更加灵活。

不过，由于 GPU 拥有多种切分方式和切分粒度，不同 GPU 型号对切分方式的支持情况有所区别，适用的工作负载也不尽相同，需要用户注意。同时，NVIDIA 要求 vGPU 搭配相应系列的 NVIDIA GRID vGPU 软件许可才能使用，对应关系如下表所示。

GRID License 类型		支持的 vGPU 类型
vApps	Virtual Application	A 系列（虚拟应用）
vCS	Virtual Compute Server	C 系列（AI 训练）
vPC	Virtual PC	B 系列（虚拟桌面）
vWS	Virtual Workstation	Q 系列（虚拟工作站）、C 系列、B 系列

适用场景

基于以上对比，我们分别整理了 GPU 直通和 vGPU 两种模式的适用场景，供用户参考：

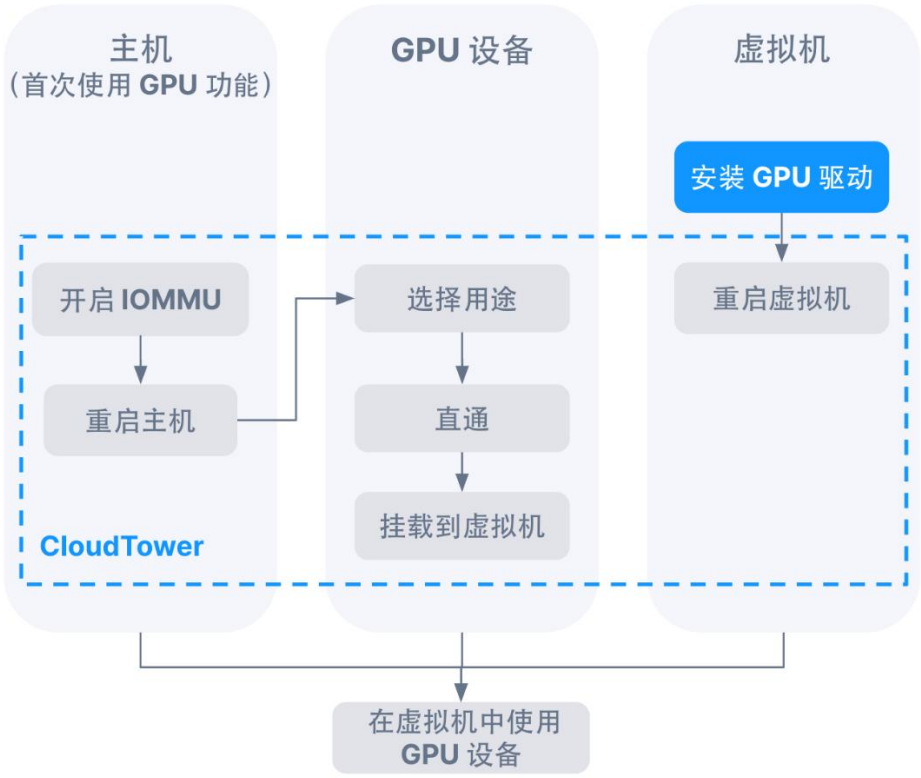
GPU 配置方式	GPU 直通	vGPU
评估要点	• 基于数据安全、资源整合等原因，希望将 GPU 相关应用负载转移到虚拟机上运行，使得相关业务的数据和负载不再零散地运行在物理的工作站之上。 • 应用对 GPU 资源要求较高，可消耗整个 GPU 的性能，无多个业务共享单个 GPU 的需求。	• 虚拟机上的应用程序无需使用单个 GPU 的全部性能。 • 希望使用有限的 GPU 设备满足多个开发团队共享使用的需求。 • 希望根据需求灵活切换 GPU 的使用模式，某个时期允许一个虚拟机使用 GPU 的全部性能（直通模式），某个时期只允许使用 GPU 部分性能（vGPU 模式）。
常见应用场景	• 机器学习模型（如金融行业使用的反欺诈、交易算法等） • 高性能计算 • VAPP	• 开发和测试（如三维设计与建模、游戏开发、图像渲染等） • 机器学习推理 • VDI

优势与价值

- 快速置备开发环境：通过虚拟机克隆和 vGPU 功能可快速置备多套 GPU 开发环境，提升开发效率。
- 灵活的资源发放：用户可根据自身需求，自由切换 GPU 直通和 vGPU 模式，灵活切分 GPU 资源，满足不同业务场景对 GPU 使用的需求，提升资源使用率的同时降低成本。
- 高性能的存储支持：三维建模、深度学习等依赖 GPU 的应用场景，通常对存储 I/O 也有较高的要求。例如：在深度学习的训练和推理场景，经常需要访问大量图像、视频、音频、文本以及结构化数据，涉及多种不同的 I/O 类型，对存储的带宽和 IOPS 都有较高的要求，存储延时也会直接影响训练算法的性能。SmartX 超融合内置自主研发的存储引擎 ZBS，能为 GPU 应用场景提供稳定、高性能、低延时存储服务。

配置方式

GPU 直通



1. 配置主机

首先确保主机上的 GPU 设备符合 SMTX OS 兼容要求，详细型号可参考文末附录。然后登录 CloudTower，为 GPU 设备所在主机开启 IOMMU 支持，并重启 SMTX OS 主机。

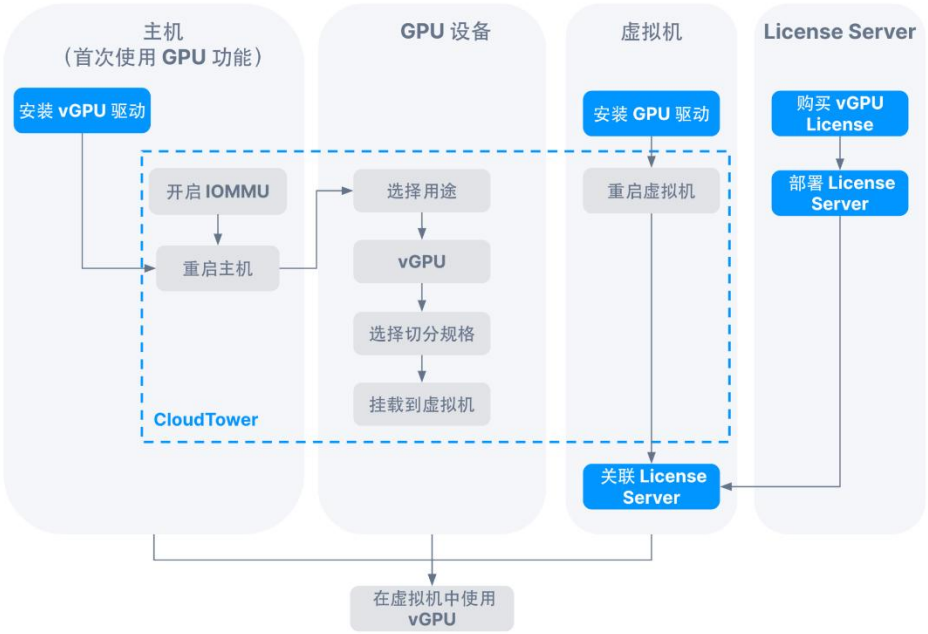
2. 挂载 GPU 设备

登录 CloudTower，先将 GPU 用途选择为直通，然后编辑指定虚拟机，选择对应的 GPU 设置直通挂载到虚拟机上。

3. 配置虚拟机

在配备了 GPU 直通的虚拟机上安装 NVIDIA vGPU 软件图形驱动并重启。

vGPU



1. 配置主机

首先确保主机上的 GPU 设备符合 SMTX OS 兼容要求，详细型号可参考文末附录。然后需在 SMTX OS 主机中安装 vGPU 驱动（NVIDIA Virtual GPU Manager），并登录到 CloudTower，为 GPU 设备所在主机开启 IOMMU 并重启主机。

2. 部署 License Server

根据业务需要，从 NVIDIA 购买对应类型和数量的 vGPU 授权。准备一个虚拟机，在其中部署并配置 NVIDIA vGPU software license server。部署及配置方式可参考 NVIDIA 用户指南（文末附链接）。

3. 选择 GPU 切分方案

GPU 卡一般可支持多种切分方案，通过登录 CloudTower ，选择合适的切分方案，如下图所示：

4. 挂载 vGPU 设备

登录 CloudTower ，然后编辑指定虚拟机，选择 vGPU模式，挂载对应的 vGPU 到虚拟机上，如下图所示：

5. 配置虚拟机

在虚拟机中安装 GPU 驱动（NVIDIA vGPU 软件图形驱动）并重启。部署及配置方式可参考 NVIDIA 用户指南。

除了对 GPU 场景的支持，SMTX OS 5.1 还通过 DRS、USB 跨节点访问、PCI 设备直通、大页内存分配、存储并发访问机制、I/O 逻辑优化等多项技术，进一步提升虚拟化和存储能力。欲了解更多 SmartX 超融合最新版本功能与性能，请阅读：SmartX HCI 5.1 发布：是超融合，更是虚拟化与容器生产级统一架构。

附录：SMTX OS 5.1 GPU 兼容列表

GPU 品牌	GPU 型号	GPU 直通支持情况	vGPU 支持情况
NVIDIA	Tesla T4	✓	✓
	Tesla V100-PCIE-16GB	✓	✓
	Tesla V100-PCIE-32GB	✓	✓
	A30	✓	✓
	A6000	✓	✓
	A40	✓	✓
	A16	✓	✓

1 NVIDIA Virtual GPU Client Licensing User Guide. （此处以 15.3 版本为例，请根据具体使用的 vGPU Manager 版本查看对应文档）
https://docs.nvidia.com/grid/15.0/grid-licensing-user-guide/index.html#abstract

DRS｜主流虚拟化动态资源平衡机制分析与 SmartX 超融合的实现优化

点击链接阅读原文：主流虚拟化 DRS 机制分析与 SmartX 超融合的实现优化

要点总结

当前主流虚拟化软件 DRS 机制大部分与 VMware 6.x 类似，以主机资源负载平衡为调度目标，其运作机制大致可分为评价体系和虚拟机迁移两部分。DRS 的评价体系负责定义集群是否属于均衡状态并执行迁移操作，该机制更适合稳态应用。

SMTX OS 优化了 DRS 评价体系，综合评估虚拟机的响应能力与主机资源利用率这两个因素，可充分适应基于容器的“敏态”应用环境，有效提升业务运行与日常管理效率。

资源的动态调度是虚拟化软件（或超融合软件）中的一项重要功能，主要指在虚拟化集群中，通过动态改变虚拟机的分布，达到优化集群可用性的目标。这一功能以 VMware vSphere 发布的 Distributed Resource Scheduler（DRS）最为人们所熟知，它凭借简单易用的特点，在用户中迅速流行，甚至成为此类功能的代名词。

作为国内领先的专业超融合厂商，SmartX 在最新发布的超融合软件版本 SMTX OS 5.1（以下简称 SMTX OS）中，也新增了对 DRS 功能的支持。不同于 VMware 6.x 及其他市场常见的虚拟化平台 DRS 实现机制，SMTX OS 优化了 DRS 评价体系，可充分适应基于容器的“敏态”应用环境，有效提升业务运行与日常管理效率。

本文，我们将深入解读目前主流虚拟化平台 DRS 实现机制与 SmartX 虚拟化平台上 DRS 功能的优化与创新，通过对比分析，帮助读者进一步了解 DRS 及其不同实现机制对集群运行带来的影响。

主流虚拟化软件 DRS 机制：更适合“稳态”业务场景

目前用户较为熟悉的 DRS 机制是 VMware 6.x 引入的实现方式（7.0 后有新机制）。该功能的初衷即通过动态改变虚拟机的放置，实现主机资源负载平衡。同时，它还可与其他虚拟化特性进行结合，衍生出丰富的应用场景：

- 通过资源平衡提升虚拟机 SLA。
- 业务低谷时，通过集中放置，减少主机使用，降低能源消耗。
- 创建业务负载时，自动进行资源评估并完成初始化放置。
- 自动完成负载均衡，最大化发挥主机性能。
- 当主机进行维护时，自动化完成迁移，降低运维复杂度。
- 修正亲和性约束。

市场中其他主流的虚拟化平台的 DRS 机制大都与 VMware 6.x 类似，以主机资源负载平衡为调度目标。该功能的运作机制大致可分为两个部分：

- 评价体系：周期性收集当前集群中的主机/虚拟机的资源使用情况，根据 DRS 的评价体系判断集群是否处于资源争抢或者分布不平衡的状态。

- 虚拟机迁移：DRS 算法基于集群当前状态进一步生成迁移建议，并根据设定的策略自动或手动触发一系列虚拟机迁移操作，最终实现集群均衡的目标。

由于 DRS 的评价体系负责定义集群是否属于均衡状态并执行迁移操作，它的实现是整个功能的重中之重。在 VMware 6.x 中，DRS 的评价体系主要关注集群状态，检查主机是否需要重新平衡。因为集群中经常会出现某台主机的资源消耗过多，而另一台主机消耗的资源较少的情况，DRS 每隔一段时间会对集群执行一次检查，如果 DRS 评价体系认为集群当前的状态可以改善，那么它将执行虚拟机热迁移（将负载高的主机上的部分虚拟机迁移到负载低的主机上）来实现集群重新平衡。

这种 DRS 评价体系的核心是平衡主机之间的 CPU、内存利用率，目前常见的虚拟化软件大都是采用类似的评价体系 。该评价体系对于传统“稳态”业务是有效的。在“稳态”业务场景下，通常单个虚拟机里面只会部署单个应用，它对 CPU、内存的占用是相对稳定的，甚至对应的虚拟机数量也是相对固定的；由于每个虚拟机的资源消耗相对固定，通过调整虚拟机的位置去重新平衡主机的负载是比较有效的。

但随着基于容器的”敏态“业务越来越流行，工作负载的形态已经发生了变化：虚拟机上可能运行着多个业务容器，容器的数量也是随时变化的，对资源的占用自然也有较大的波动，而且这是该类型应用的一个常态。如果还是基于上述这种传统的 DRS 评价体系，那么就有可能出现这样的情况：某个时刻虚拟机启动更多数量的容器后，触发 DRS 平衡被迁移到其他主机，然而过了一段时间又由于资源占用下降再次触发迁移——虚拟机可能会不断地执行一些无用的迁移，并对应用运行带来较大的干扰。面对应用负载的变化，传统的 DRS 评价体系未必适用，用户或许需要一种更适应容器环境的新型评价体系。

SMTX OS 5.1 DRS 功能：新型评价体系更灵活高效

SmartX 最近发布了新版本超融合软件 SMTX OS 5.1，新增了对 DRS 功能的支持。SmartX 研发团队在 DRS 功能设计之初就意识到，基于主机资源利用率的 DRS 评价体系无法完全满足新型应用负载的需求。为此，研发团队设计了一种新型评价体系，综合评估虚拟机的响应能力与主机资源利用率这两个因素。

DRS 评价体系

SMTX OS 5.1 DRS 的运作逻辑是：

- 周期性为虚拟机和主机分别进行评分。
- 根据虚拟机状态进行评分，虚拟机因为资源争抢导致响应能力受损的程度越高，虚拟机的评分就越低，反之评分越高。
- 根据主机的资源利用率进行评分，主机空闲度越高，评分越高；主机越繁忙则评分越低。
- 评分最低的虚拟机会优先迁移到评分最高的主机（同时考虑迁移收益），依次类推直到主机之间达到平衡状态。

虚拟机评分

虚拟机的评分会针对虚拟机的 CPU、内存和存储 3 种资源的使用情况进行评分，其中 CPU 和内存的评分占比 80%，而存储评分占 20%。当前 DRS 版本暂不支持虚拟机网络相关评分。

- 虚拟机 CPU 评分

  系统通过监控虚拟机 CPU 的 Steal Time，以相应的公式计算出虚拟机 CPU 分数。虚拟机 CPU 资源争抢越严重，虚拟机 CPU 分数就越低。
- 虚拟机内存评分

  当内存没有超分的时候，虚拟机内存评分应为 100%，因为内存资源没有争抢的现象发生。在内存超分的场景下，系统会监控虚拟机使用共享内存的程度，使用比例越高，证明资源争抢越严重，得分越低。
- 虚拟机存储评分

  SMTX OS 拥有 I/O 本地化功能，虚拟机读取数据时会优先访问本地宿主机的副本以缩短 I/O 延时，

提升性能。系统会根据虚拟机所在节点拥有虚拟机的副本数据块比例进行评分，如虚拟机运行的宿主机拥有它的一个完整副本的所有数据块，那么它的存储评分就是 100%；如果只拥有部分副本，意味着虚拟机访问数据的时候需要跨网络读取，那么虚拟机的存储评分就会降低。

主机评分

主机评分与虚拟机评分类似，同样会考察 CPU、内存和存储三者的情况，进行综合评价。

- 主机 CPU 评分

  系统会收集主机的 CPU 空闲时间，空闲时间越多，得分也就越高。反之，主机 CPU 越繁忙得分越低。
- 主机内存评分

  系统收集主机可用内存占比，并根据主机是否存在内存超分进行评分，在没有超分的情况下，可用内存越多，评分越高。
- 主机存储评分

  由于 SMTX OS 内置的存储引擎可支持存储容量的自动均衡，因此主机存储评分并不是以主机本地存储容量使用率进行评价的。主机存储评分主要关注主机拥有虚拟机副本的数据块比例，拥有比例越高，得分越高。

收益评价

对于虚拟机的动态调整，单纯参考虚拟机和主机的评分仍然是不够的，因为不是所有情况都需要虚拟机迁移到分数更高的主机上，还需要考虑到迁移的成本——只有当迁移带来的收益大于成本，DRS 才建议或执行迁移动作。例如，某个虚拟机的内存比较大，迁移时需要传输比较多的数据，虽然迁移后会使得负载更为平衡（但差距不大），这个时候收益可能小于成本，那么 DRS 便不会建议虚拟机进行迁移。

DRS 建议与迁移

DRS 建议敏感度

SMTX OS 的 DRS 功能为用户提供了 3 种敏感度设置，便于用户根据业务情况选择合适的虚拟机迁移建议生成条件。

敏感度级别	说明
保守	仅在虚拟机发生资源争抢且影响性能的情况下生成建议。
标准	资源分布出现重度不均衡时才生成建议。
激进	资源分布出现轻度不均衡时即生成建议。

DRS 自动化级别

同时，SMTX OS 的 DRS 功能支持手动迁移和自动迁移两种虚拟机迁移方式。

- 手动迁移

  DRS 只会给出迁移建议，但不会自动执行迁移。该模式采取保守的调度策略，可以让用户充分评估 DRS 考察范围之外的因素（如：某个业务十分重要，不希望在非预期的时间执行迁移），再决定是否执行迁移。与此同时，用户需要定期关注迁移建议，并决定执行建议或放弃建议。
- 自动迁移

  DRS 会根据生成的迁移建议自动执行虚拟机迁移操作，直到符合预设 DRS 阈值。该模式采取自动化调度策略，用户不需要人工接入迁移，后台会自动完成平衡，对于大规模虚拟机集群，可以大幅降低运维的压力。

SMTX OS DRS 优势与价值

得益于新型的评价体系和运维友好的操作设置，SMTX OS DRS 功能可以充分满足云环境下的资源调度需求，帮助用户在优化集群性能的同时提升运维管理效率。相比主流传统 DRS 功能，SMTX OS DRS 能为用户带来以下具体收益：

- 更丰富的应用场景：采用以虚拟机响应能力为核心的 DRS 评价体系，使得 DRS 适用范围更广。
- 更智能的调度机制：评价体系更加全面，充分结合超融合架构特点，将存储性能因素纳入评价体系，同时引入收益评价，减少不必要的迁移动作，进一步优化决策、降低开销。
- 更简单的运维管理：敏感度与自动化设置在降低运维难度的同时，给予了运维人员更大的使用灵活性。

除了对 DRS 机制的优化，SMTX OS 5.1 版本中还新增了 GPU 直通与 vGPU、USB 跨节点访问、大页内存分配等创新性虚拟化与存储能力。搭配新发布的容器管理与服务软件 SMTX Kubernetes 服务 1.0 和跨虚拟化与容器平台的软件定义负载均衡、可观测平台等产品组件，SmartX HCI 5.1 全面提升虚拟化、分布式存储、分布式防火墙、系统运维、灾备、迁移等基础设施能力，助力客户以一套架构平稳实现基础架构云化、国产化替代和容器化转型目标。

网络 I/O 虚拟化 | 一文了解 SMTX OS 的虚拟网卡、PCI 直通、SR-IOV 直通技术

要点总结

越来越多的行业应用对网络性能和隔离性有了更高要求，如低延迟、高带宽等。针对以上需求，SmartX 在全新发布的超融合软件 SMTX OS 5.1 版本中新增了 PCI 网卡直通的能力，对于 ELF 虚拟机，SMTX OS 支持虚拟网卡、SR-IOV 直通网卡、PCI 直通网卡三种网络设备，以满足期货交易、高性能计算等多种高网络要求场景的生产级使用。

背景

随着技术的不断发展，不少行业应用都对网络性能和隔离性有着越来越高的要求。例如：

- 低延迟：一些期货行业用户选择在期货公司机房托管服务器并自行编写交易程序，以实现对市场波动的快速（微秒级）反应。尤其是在高频交易场景下，毫秒级甚至微秒级的延迟都可能对交易的最终受益带来较大影响。对于这些应用，降低网络延迟能够为业务价值带来显著提升。
- 高带宽：科学计算、模拟和仿真等高性能计算工作负载通常涉及大规模数据的传输和处理。在科学研究、工程模拟、气象预测等领域，需要传输大量的数据；同时，高性能计算通常利用大规模的并行计算集群，其中包括数百甚至数千台计算节点。这些节点需要频繁地互相通信以同步计算结果、传递数据和协调计算任务。这些业务场景都需要高网络带宽来提升数据传输和整体计算效率。

针对这些需求，网络 I/O 虚拟化技术可以优化网络资源、提升 I/O 性能，目前有三种主流实现方式：仿真（Emulation）、半虚拟化（Para-virtualiztion）和直通（Pass-through）。为了帮助用户更好地实现网络优化，SmartX 在全新发布的超融合软件 SMTX OS 5.1 版本中新增了 PCI 网卡直通的能力。结合之前版本已经支持的 SR-IOV 直通功能和虚拟网卡，SmartX 超融合可为原生虚拟化 ELF 集群提供完整的网络 I/O 虚拟化技术支持能力。用户可根据自己在网络性能、隔离性和成本投入上的实际需求选择最合适的方案，实现期货交易、高性能计算等多种高网络要求场景的生产级使用。

本文，我们将详细介绍 SmartX 超融合网络 I/O 虚拟化支持能力，并针对网卡在 PCI 直通和 SR-IOV 直通模式下的性能表现开展测试。

SMTX OS 5.1 网络 I/O 虚拟化

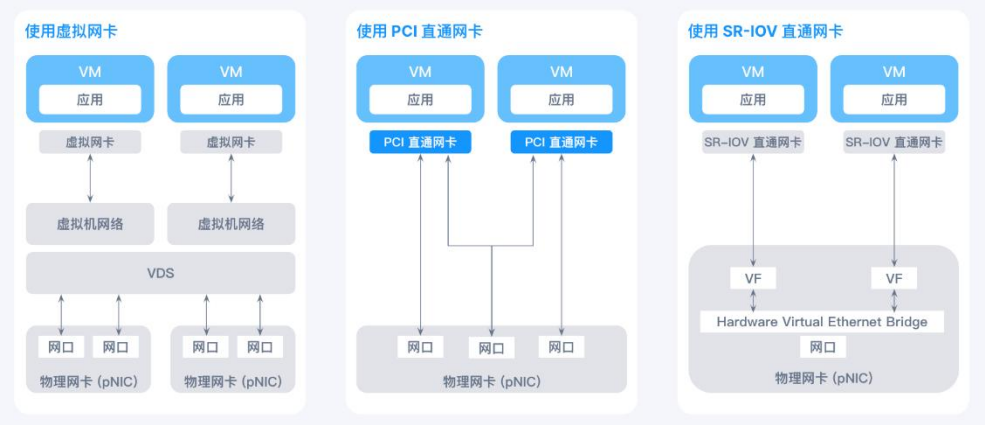
功能特性

为了满足不同用户和应用场景需求，SMTX OS 5.1 新增了 PCI 网卡直通的功能。目前，对于 ELF 虚拟机，SMTX OS 支持虚拟网卡、SR-IOV 直通网卡、PCI 直通网卡三种网络设备。

- 虚拟网卡：通过软件模拟物理网卡的功能，使得虚拟机可以与外部网络互联。
- PCI 网卡直通：将主机上的网卡作为 PCI 直通网卡透传给虚拟机使用，该网卡由这台虚拟机独占。
- SR-IOV 直通：将一个支持 SR-IOV 的物理网卡虚拟化出多个 VF (Virtual Function)，作为 SR-IOV 直通网卡直接挂载给虚拟机使用，可实现多个虚拟机共享同一个物理网卡的通信能力。

每个主机上可以使用不同型号的网卡*，每个虚拟机可挂载多种网络设备。

**兼容列表见文末附录。*



虚拟网卡

虚拟网卡是虚拟化环境中使用最广泛的网络适配器，除了帮助虚拟机与网络通信之外，虚拟网卡还通过网络隔离策略确保了不同虚拟机之间的网络流量隔离，并可结合虚拟交换机（VDS）和虚拟机网络创建复杂的虚拟网络拓扑。

虚拟网卡具备相当优秀的灵活性和弹性：为虚拟机配置虚拟网卡后，可以根据需要再次修改配置，也可以通过克隆已有配置来加快新服务的部署速度；同时，由于消除了对硬件的依赖，虚拟机在快照和迁移时都可保留虚拟网卡的配置，可轻松将虚拟机迁移或重建至其他物理主机。

尽管在大部分场景下，虚拟网卡的性能已经可以满足需要，但通过仿真（E1000 类型的虚拟网卡使用此技术）或者半虚拟化（例如 VIRTIO 类型的虚拟网卡）来模拟物理网卡都会带来额外的性能开销。另外，如果某些业务虚拟机对网络要求较高，在共享 VDS 或虚拟机网络时，这些比较大的流量都会挤占其他虚拟机的正常流量，造成网络性能分配的失衡。

PCI 直通网卡

网卡直通利用了 PCIe Pass-through 的技术，允许虚拟机直接访问并使用 SMTX OS 主机上的网卡。直通网卡具有良好的兼容性，可以支持大部分 Guest OS 和符合 PCIe 总线标准的网卡。同时，由于虚拟机操作系统绕过了虚拟化层，直接使用物理网卡，缩短了数据传输的路径，使得虚拟机可以获得接近物理机使用物理网卡的性能与完整的硬件特性。此外，虚拟机独享物理网卡也提供了更高级别的隔离性。

然而，PCI 直通模式下，一张物理网卡不能同时直通给多个虚拟机使用，因此如果有多个虚拟机有通过直通网卡提升性能的需要时，需要保证主机上有多张物理网卡。另外，挂载了 PCI 直通网卡的虚拟机不支持 HA、热迁移等操作。

SR-IOV 直通网卡

SR-IOV (Single Root – I/O Virtualization) 是一种基于硬件的虚拟化解决方案，启用了 SR-IOV 能力的物理网卡可被切分为多个 VF (Virtual Function) 并作为 SR-IOV 直通网卡被挂载给虚拟机使用，这允许了多台虚拟机共享一个物理网卡，在提升性能的同时兼顾性价比。用户可以按照实际的业务需求，为同一台虚拟机分配多个 SR-IOV 直通网卡。

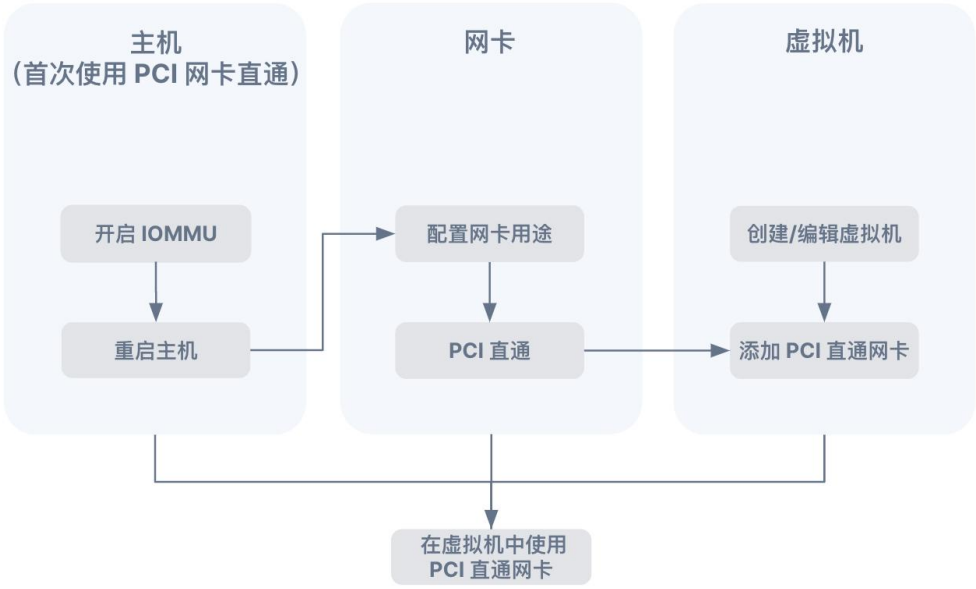
**该功能要求物理网卡本身具备 SR-IOV 特性，且对于特定网卡需要在操作系统中安装驱动。*

适用场景

虚拟机网络设备选择	虚拟网卡	PCI 直通网卡	SR-IOV 直通网卡
评估要点	• 承载的业务对于网络性能无特别要求，能处理适度网络流量和延迟即可 • 希望能够灵活地支持虚拟机 HA 和迁移，要求虚拟机的网络设备具有通用性	• 业务对网络性能要求高，可消耗整个网卡的性能，无多个业务虚拟机共享单个网卡的需求 • 业务有高级别的网络隔离要求	• 业务对网络性能要求较高 • 希望使用有限的物理网卡满足多台虚拟机共享使用的需求 • 希望能够灵活切换网卡的用途为 PCI 直通或 SR-IOV 直通
常见应用场景	• 通用计算虚拟机 • 开发和测试环境	• 高性能计算（科学计算、仿真、模拟等需要低延迟网络的业务） • 金融、医疗或政府等对安全性有严格要求的环境	• 期货投资交易 • 多租户环境 • 网络密集型应用，如视频流处理
优势与价值	• 虚拟网卡的配置与管理相比于直通网卡更为灵活和简单 • 不需要额外的硬件支持，可以降低投入的成本	• 提供接近物理网络性能的高带宽和低延迟 • 可充分利用硬件的高级网络功能 • 提供更好的网络隔离性	• 均衡高网络性能与硬件投入成本，实现资源的有效利用 • 相比于 PCI 直通网卡，可为 SR-IOV 直通网卡配置 MAC 地址和 IP 地址

配置方式

PCI 直通



1.配置主机

首先确保主机上要用于 PCI 直通的物理网卡符合 SMTX OS 兼容要求，详细型号可参考文末附录。

登录 CloudTower，为要直通的物理网卡所在主机开启 IOMMU 支持，并需确保主机 BIOS 中也已启用 IOMMU，完成后重启 SMTX OS 主机。

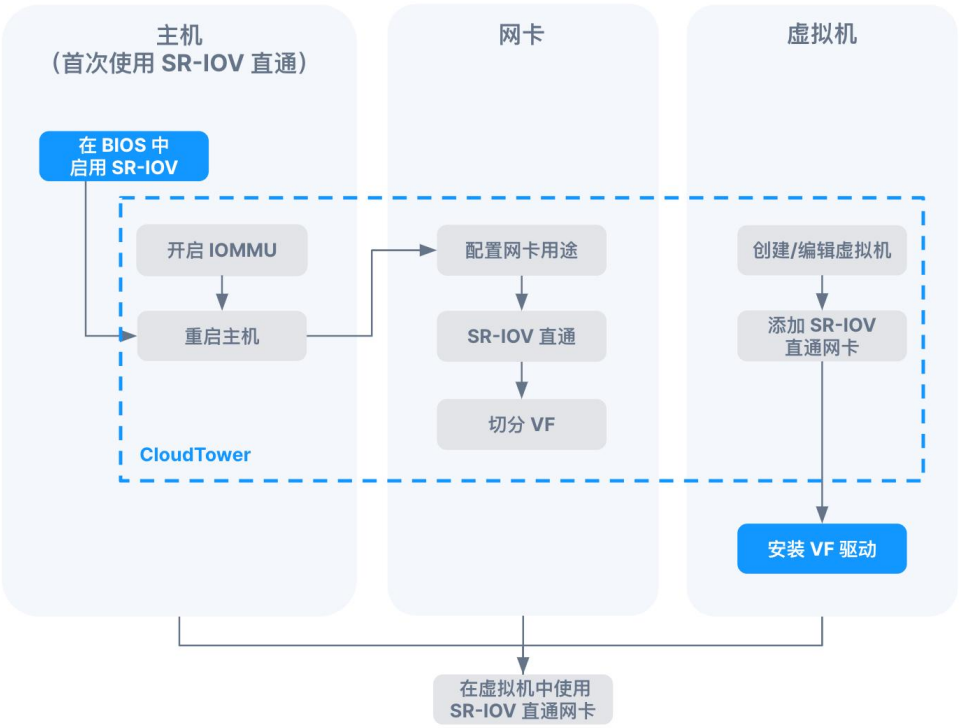
2.配置网卡用途

登录 CloudTower，将网卡用途置为 PCI 直通。

3.添加 PCI 直通网卡

编辑指定虚拟机，选择对应的物理网卡作为 PCI 直通网卡添加到虚拟机上。

SR-IOV 直通



1.配置主机

首先确保主机上的要用于 SR-IOV 直通的物理网卡符合 SMTX OS 兼容要求，详细型号可参考文末附录。登录 CloudTower，为要直通的物理网卡所在主机开启 IOMMU 支持，并需确保主机 BIOS 中也已启用 IOMMU 和 SR-IOV，完成后重启 SMTX OS 主机。

2.配置网卡用途并切分 VF

登录 CloudTower，将网卡用途置为 SR-IOV 直通；并根据实际需要和网卡的支持情况配置制定数量的 VF。

3.添加 SR-IOV 直通网卡

编辑指定虚拟机，选择对应的物理网卡作为 SR-IOV 直通网卡添加到虚拟机上。

4.配置虚拟机

根据物理网卡的型号与虚拟机的客户端操作系统，按需为虚拟机安装 VF 驱动。

性能测试

为了让读者直观感受 SMTX OS PCI 直通与 SR-IOV 直通能力，我们使用不同型号的网卡进行了性能测试。

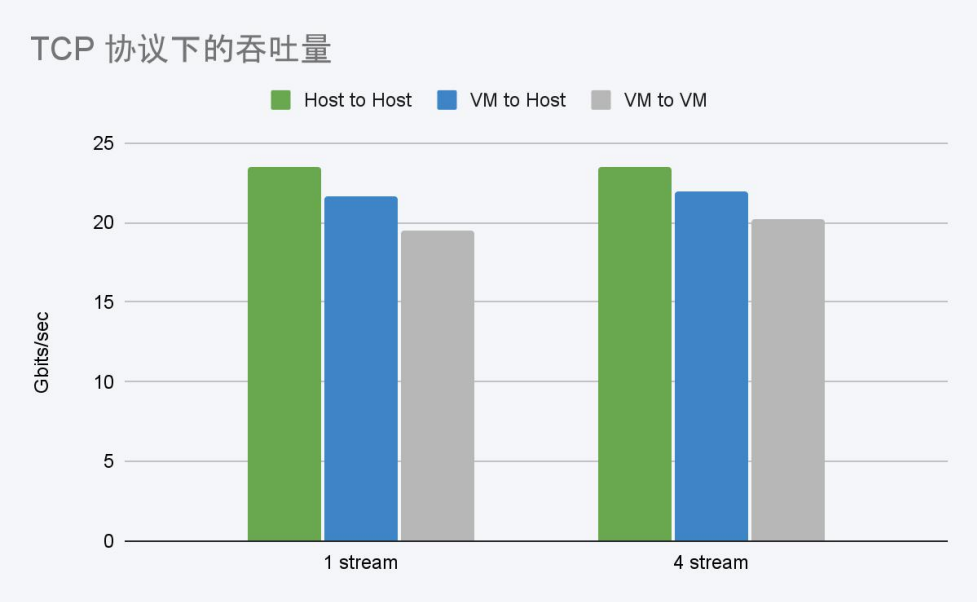
PCI 直通

性能测试借助 netperf 和 iperf3 工具进行，对两种不同型号的网卡（Solarflare、Mellanox）展开 3 个场景的测试。测试结果以物理机到物理机（Host to Host）作为基准，通过对比挂载 PCI 直通网卡的虚拟机到虚拟机（VM to VM）、虚拟机到物理机（VM to Host）在不同测试场景下的吞吐量和延迟数据，分析

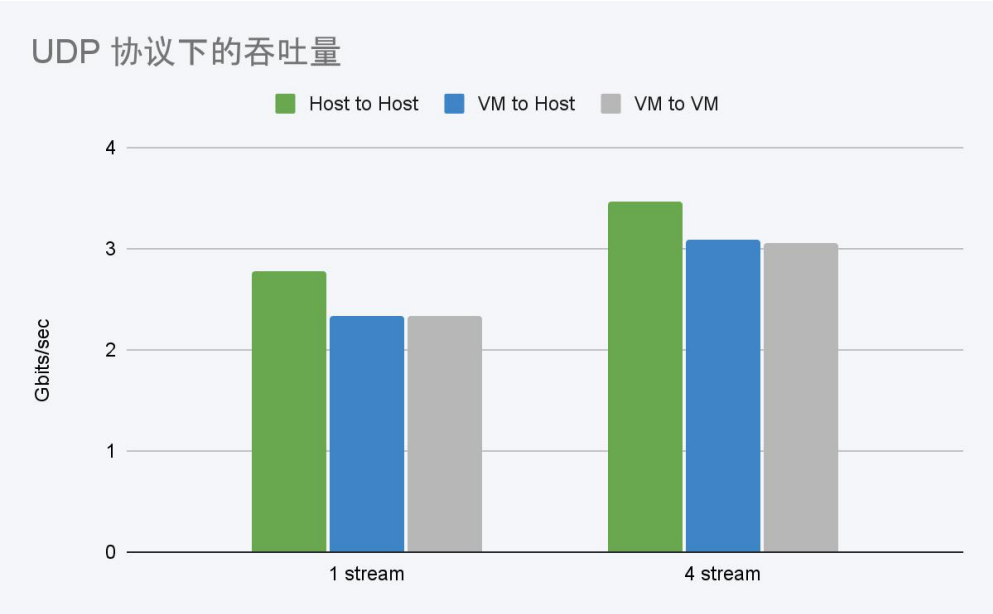
得出使用 SMTX OS PCI 直通的效果。

**硬件配置及测试工具见附录。*

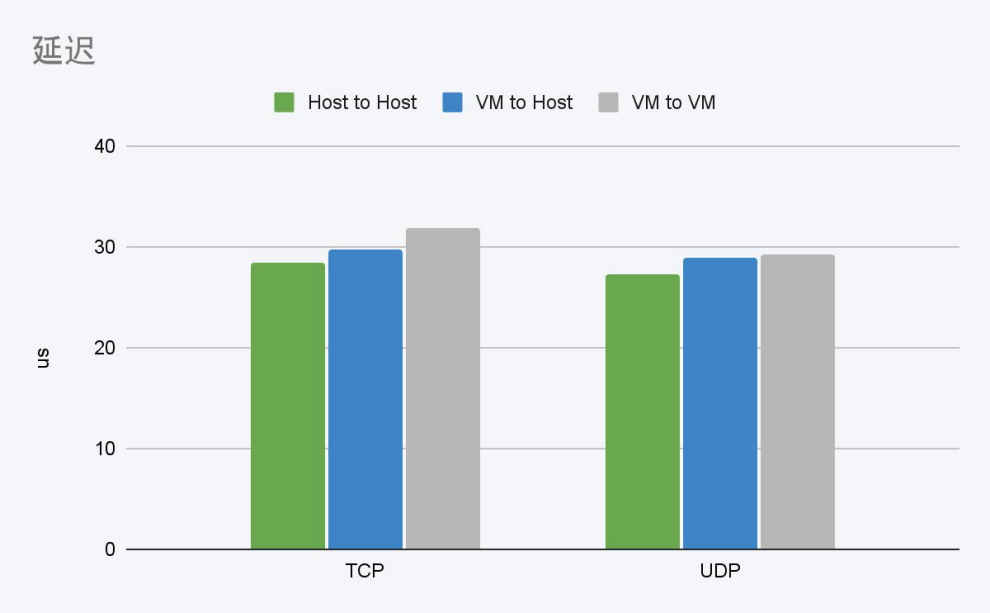
测试数据



将 PCI 物理网卡直通给虚拟机后，网卡在 TCP 协议下的吞吐量会有小幅度的降低；这是因为虚拟机需要与物理主机上的其他虚拟机和应用程序共享物理资源，而 KVM 虚拟化通常需要额外的计算能力来处理这些管理和调度资源的任务，从而增加了处理开销和网络延迟。



将 PCI 物理网卡直通给虚拟机后，网卡在 UDP 协议下的吞吐量同样会有小幅度的降低；降低是因为虚拟机的网络栈必须绕过虚拟化层直接与物理网络适配器通信，使用 PCI 直通网络设备可能会引入额外的开销，例如中断处理和内存访问。



将 PCI 物理网卡直通给虚拟机后，网卡的延迟会有小幅提升；提升是因为虚拟机的网络栈必须绕过虚拟化层直接与物理网络适配器通信，使用 PCI 直通网络设备可能会引入额外的开销，例如中断处理和内存访问。这种开销的影响在 TCP 流量中更加明显，因为相对于 UDP 流量，TCP 流量往往涉及更频繁和更小的数据包。

测试结论

在 PCI 直通模式下，虚拟机可以直接访问物理网卡，但是这也会带来额外的开销，例如 DMA 编程、中断处理等，这些操作会消耗一定的 CPU 和内存资源。同时，虚拟化技术中多重层次的处理和转发也可能会导致网络吞吐量的降低。

但是，这样的额外开销占比非常小。因此，排除部分干扰因素，物理网卡使用 PCI 直通可以获得与物理网卡极接近的性能，完全能够满足虚拟机网络环境低延迟的需求。

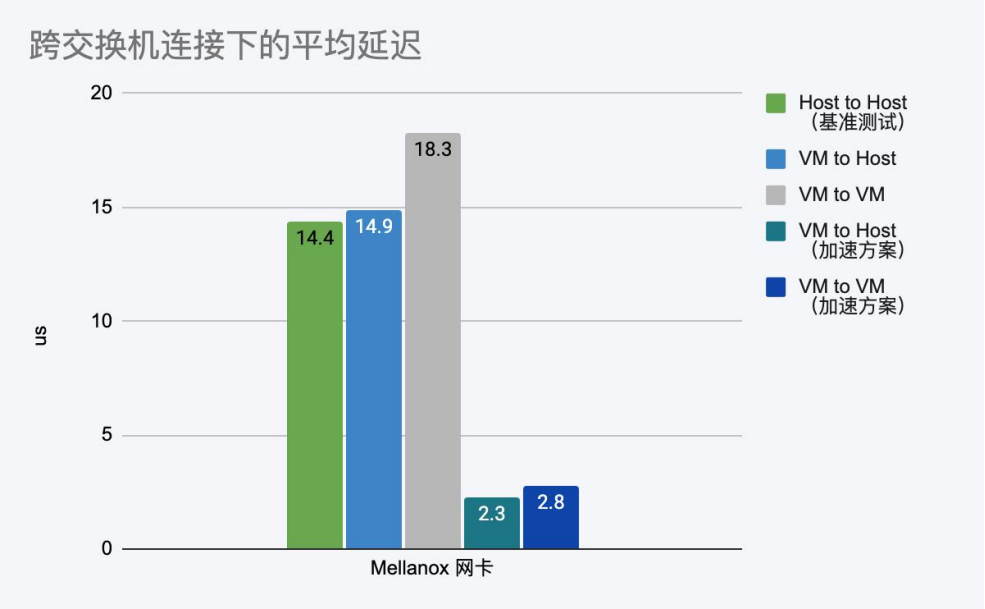
SR-IOV 直通

为验证使用 SR-IOV 直通网卡能否获得与物理机接近的性能，我们使用 sfnettest 工具对不同型号网卡进行了性能测试。以物理机到物理机（Host to Host）、挂载 VIRTIO 网卡的虚拟机到虚拟机（VM to VM）的测试数据作为基准，通过对比挂载 SR-IOV 直通网卡的虚拟机到虚拟机（VM to VM）、虚拟机到物理机（VM to Host）在不同测试场景下的延迟数据，分析 SR-IOV 直通网卡的性能水平。

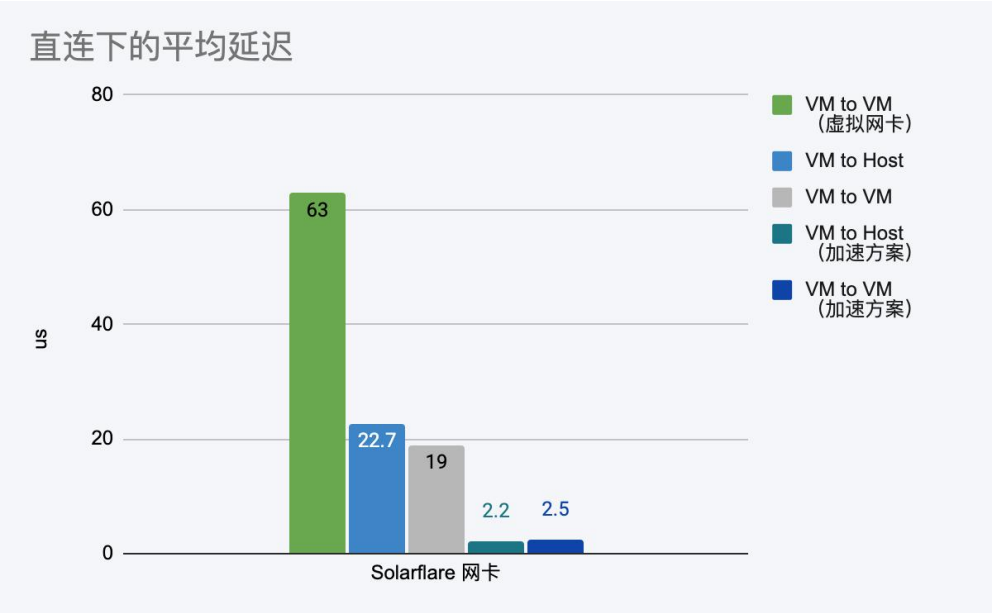
对于 Solarflare 和 Mellanox 网卡，结合了各自的加速方案，对比启用加速前后的数据。

**硬件配置及测试工具见附录。*

测试数据



跨交换机的连接方式下，与基准测试中 Host to Host 的平均延迟 14.4 us 相比，使用 SR-IOV 直通网卡在不使用加速方案的情况下，在 VM to VM、VM to Host 场景下均可提供与物理机接近的性能。Mellanox 网卡结合加速方案，在各场景下均可进一步降低延迟至 2-3 us。



直连方式下，与基准测试中 VM to VM 使用 VIRTIO 网卡的平均延迟 63 us 相比，使用 SR-IOV 直通网卡在不使用加速方案的情况下，在 VM to VM 和 VM to Host 场景下均可大幅降低延迟。Solarflare 网卡结合加速方案，在各场景下还可进一步降低延迟至 2-3 us。

**期货公司一个交易订单的数据量大小通常为 64~128 个字节，本次测试均采用发包大小为 64 字节（size = 64）的延迟平均值（mean）进行对比。*

测试结论

在 SR-IOV 直通模式下，虚拟机可以直接访问物理网卡虚拟化出的 VF。排除部分额外开销带来的影响，虚拟机使用 SR-IOV 直通网卡相比于使用 VIRTIO 网卡可以大幅降低延迟，并获得与物理机极接近的延迟性能。结合加速方案， Solarflare 网卡和 Mellanox 网卡平均延迟还可以得到进一步降低。

临时副本机制｜副本降级导致数据丢失？SmartX 超融合利用临时副本优化多副本机制

点击链接阅读原文：副本降级导致数据丢失？SmartX 超融合利用临时副本优化多副本机制

要点总结

多副本机制是超融合软件常用的数据保护方式，可以为存储数据提供冗余保护。但在主流实现方式下，这一机制无法避免“副本降级”期间带来的风险：在副本恢复完成之前，集群整体副本数依旧少于预期，此时若健康副本同样遭遇故障或意外离线，将很有可能导致数据丢失。

SMTX OS 5.1 提供了一种创新性的数据恢复机制，引入了“临时副本”，可确保新写入的数据维持副本级别（整个数据恢复过程不降级）；甚至在数据恢复期间发生其他叠加故障导致所有健康副本异常时，依然允许系统通过特殊的恢复机制进行数据修复（支持完全修复和部分修复），有效降低了副本降级问题带来的风险，提高故障场景下的数据安全等级。

多副本机制是超融合软件常用的数据保护方式，可以为存储数据提供冗余保护——即使一个或部分副本异常，系统仍可通过健康副本进行副本恢复。但是，主流实现方式下，这一机制依旧无法避免“副本降级”期间带来的风险：在副本恢复完成之前，集群整体副本数依旧少于预期，此时若健康副本同样遭遇故障或意外离线，将很有可能导致数据丢失。为进一步提高数据安全性，SMTX OS 5.1 引入了“临时副本”这一创新机制，保证副本数据恢复期间“副本不降级、数据不丢失”，满足关键业务连续稳定运行的需求。

主流多副本保护机制的缺陷

多副本保护机制介绍

多副本技术，顾名思义就是一份数据对应多个相同的数据副本，多个副本按照既定的规则放置在不同的设备当中，以避免硬件故障导致的数据损坏或丢失。当发生硬件故障导致集群中一个副本或多个副本离线或损坏时，一方面，健康副本可保证正常 I/O 读写；另一方面，系统可通过拷贝健康副本重新生成多个副本，以恢复数据的副本级别，实现数据冗余保护。

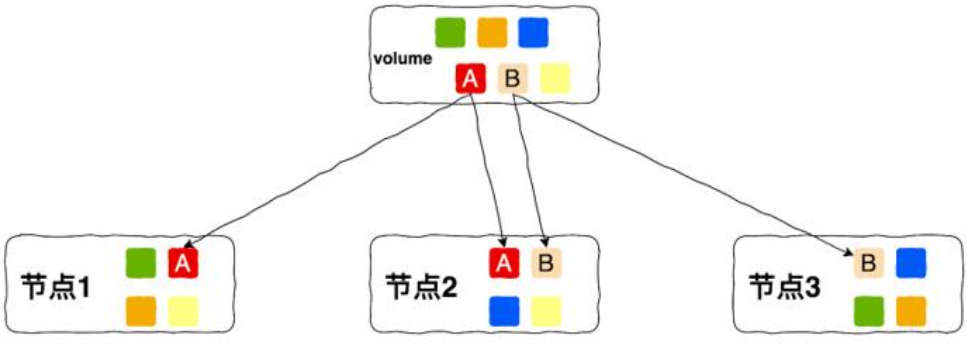


图 1

如图 1 所示：以 2 副本为例，存储卷（volume）被切分为多个数据块，而每个数据块都拥有 2 个副本。如数据块 A 的两个副本分别放置在服务器节点 1 和服务器节点 2 上。即使任意一台服务器节点宕机，至少还有一个副本可以正常访问，从而达到数据冗余保护的目的。

SmartX 超融合基础架构同样采用多副本技术为虚拟机提供数据冗余保护。它支持配置 2 副本、3 副本共两种存储策略，不同的副本策略可以容忍不同级别的硬件损坏。

当前问题

超融合集群除了需要面临硬件损坏， 还可能面临诸如硬盘误操作拔出、服务器节点意外重启、存储网络闪断等问题，最终导致硬件的短暂离线（一段时间后重新上线）。这些情况通常会引起副本降级（副本数量少于预期），而且系统也很难辨别这是一次短暂离线还是永久离线。以 2 副本为例，在发生副本离线超时后，后台会自动剔除异常副本，以健康副本来响应正常的 I/O 请求，同时触发数据恢复来重建缺失的数据副本。但在数据完成恢复之前，新的数据读写都只发生在唯一的健康副本之上。

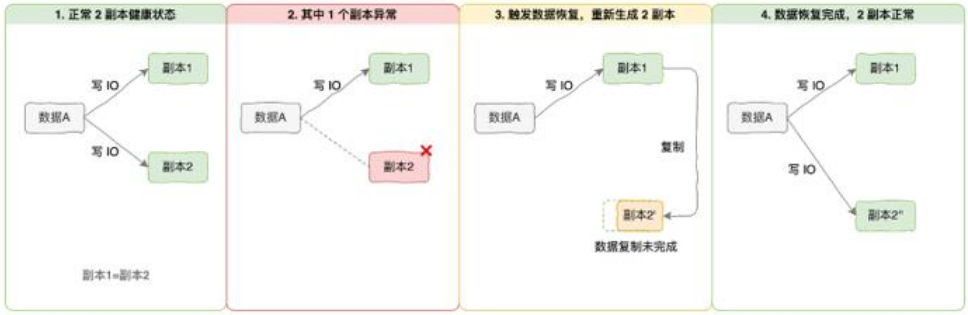


图 2

如图 2 所示：以 2 副本策略为例，正常情况下，数据 A 包含 2 个副本，当发生 I/O 写入时，副本 1 和副本 2 会同步写入 I/O。当副本 2 异常下线之后，副本 2 将被剔除，并触发数据恢复到副本 2’。在数据恢复期间，数据的变化只写入了副本 1 （副本降级），直至副本 2’ 重建完成之后，副本 2’’ 才能重新接受写 I/O（副本级别恢复）。

注：副本 2” 表示完成数据恢复的新副本，与未完成数据复制的副本 2’ 进行区分。

在副本降级期间，实际只有 1 副本在工作，一旦出现健康副本受损离线的情况，那么就有很大几率导致数据丢失。下面将通过图例说明副本降级期间的风险。

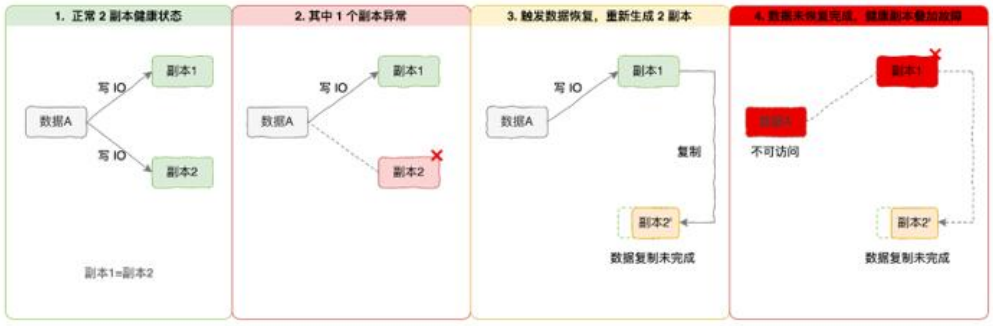


图 3

如图 3 所示：当副本 2 发生异常而无法访问时，I/O 可以正常写入并更新到副本 1，同时后台将触发数据恢复（通过拷贝副本 1 的数据重新生成副本 2’’）。而数据恢复是需要一定时间的（时长取决于数据量的大小），在数据恢复完成之前，副本 2’ 是无法正常访问的（数据不完整）。如果这个时候副本 1 遭遇硬件故障或其他原因的损坏而无法访问，那么数据 A 将没有办法通过任何可用的副本进行恢复，也无法正常读写，大概率导致数据丢失的情况发生。

SmartX 超融合临时副本技术原理

针对这一以上问题，SMTX OS 5.1 提供了一种创新性的数据恢复机制，可有效降低副本降级问题带来的风险，提高故障场景下的数据安全等级。新的处理机制引入“临时副本”概念，可确保新写入的数据维持副本

级别（整个数据恢复过程不降级）；甚至在数据恢复期间发生其他叠加故障导致所有健康副本异常时，依然允许系统通过特殊的恢复机制进行数据修复（支持完全修复和部分修复），从而最大限度地保障数据恢复期间的数据安全性。

概念定义

- **健康副本**：可提供完整读写能力的数据副本。
- **失败副本**：出现异常、无法提供正常读写能力的数据副本，可基于临时副本进行数据修复。
- **临时副本**：在健康副本降级期间，负责记录新数据的写入，不负责数据的读取。

临时副本运作机制

SMTX OS 5.1 采用临时副本策略提高异常发生后数据副本的安全性。

- **数据组成**：当出现副本降级，需要剔除异常副本时，通过分配临时副本响应写请求，记录数据恢复期间新写入的数据，以保证数据副本数满足预期（临时副本数据 + 健康副本数据 = 完整副本数据）。
- **生命周期**：数据恢复期间，会保留异常副本并标记为失败副本，每恢复一个健康副本，就移除对应的失败副本和临时副本。
- **无损修复**：如果在数据恢复期间叠加其他故障导致所有副本均出现异常，在失败副本恢复访问的情况下，可以通过特殊恢复机制人工进行数据修复，将临时副本上的数据重放至失败副本上，形成完整的健康副本。
- **适用场景**：临时副本策略仅能改善因为可恢复故障（由于网络或者其他原因造成的副本临时下线）带来的副本降级。

数据恢复

正常数据恢复

2 副本或 3 副本数据在出现单个副本异常时，会为异常副本分配对应的临时副本，记录副本异常后的新写入的数据，并使用健康副本作为数据源进行数据恢复。

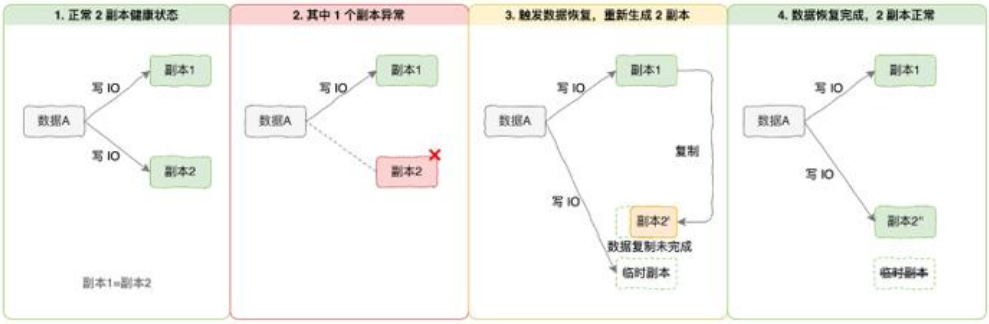


图 4

如图 4 所示： 以 2 副本为例，当副本 2 离线无法访问时，系统将从元数据中剔除异常副本，同步触发数据恢复，并额外创建临时副本，所有新写入的数据会同步写入副本 1 和临时副本（新写入数据维持 2 副本级别，不降级）；与此同时数据恢复也同时在进行，通过复制副本 1 重新形成副本 2‘，当所有数据恢复完成，副本 2’‘ 处于正常状态，系统会自动删除临时副本。

利用临时副本进行无损修复

在数据恢复的过程中，如果不幸，唯一的健康副本也发生了损坏或下线，那么虚拟机将无法访问任何副本数据，也没有办法正常进行 I/O。但如果这时候当初离线的失败副本已经重新上线（如机器重新启动或者网络恢复等），且数据没有损坏的情况下，系统仍然可以合并失败副本（首次副本离线前的旧数据）和临时副本（离线后写入的新数据）进行修复，但虚拟机在数据修复期间无法进行 I/O，直至修复完成。下面将举例说明无损修复是如何实现的。

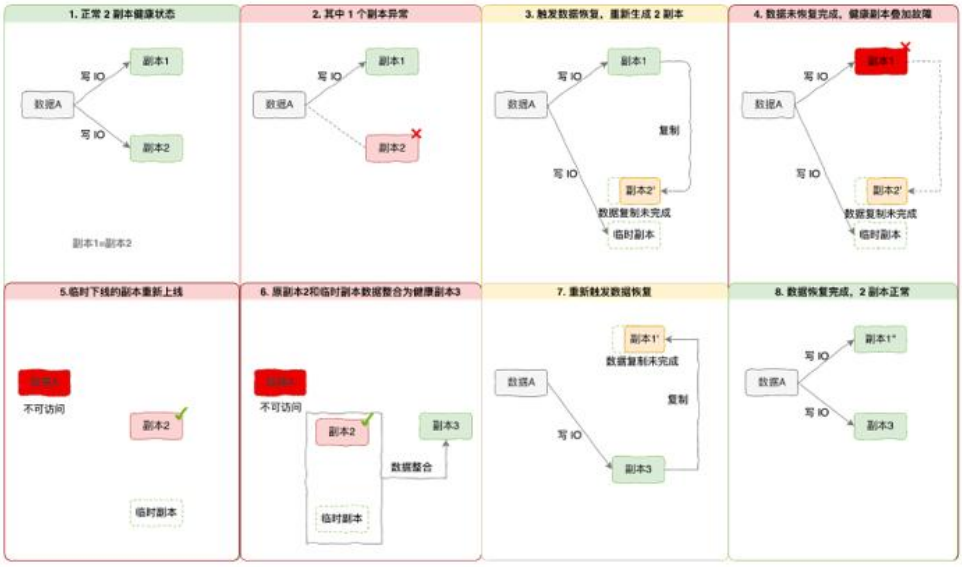


图 5

如图 5 所示： 当副本 2 发生异常离线，系统自动触发数据恢复并创建临时副本，新写入的数据会同时记录到副本 1 和临时副本当中，并通过数据恢复生成副本 2'。如果在数据恢复过程中（副本 2' 未完全复制完成的情况），发生叠加故障，副本 1 发生了损坏或离线时，那么这个时候已经没有任何完整副本可供访问了，数据 A 也完全离线了。而系统可对原有的副本 2 的数据（已经重新上线）和临时副本（增量数据）进行数据整合，形成完整的数据副本 3，此时数据 A 可以重新上线并接受读写请求，同时可重新触发数据恢复，等待完成数据恢复之后，重新形成健康的 2 副本状态。

临时副本机制的限制和影响

限制

临时副本主要针对硬件短暂离线下的副本降级风险，但对于不可恢复的硬件故障（如磁盘损坏等），临时副本没有直接的效果。如要应对此类多个硬件叠加故障的场景，建议采用高级别的副本策略（3 副本策略）进行保护。

影响

- **空间影响**：临时副本的创建会额外占用存储空间，但是数据恢复完成后，这部分空间会自动回收。
- **性能影响**：由于虚拟机变化的数据会同步写入临时副本，因此这一过程对虚拟机的写入性能会有一定的影响（持续至数据恢复完成）。目前性能影响在可接受的范围内，而且该功能也在持续优化过程中，后续版本这部分影响将进一步降低。

除了对多副本机制的优化，SMTX OS 5.1 还新增了 GPU 直通与 vGPU 支持、PCI 直通支持、DRS 优化等虚拟化与存储能力。搭配软件定义的网络负载均衡、可观测平台等产品组件，SmartX 超融合 5.1 版本全面提升虚拟化、存储、网络与安全、运维管理支持等基础设施能力。

更多资源

下载文档

SmartX 超融合基础设施及 SMTX Halo 一体机产品介绍

SmartX 分布式块存储 ZBS 自主研发之旅

SmartX 行业客户案例集

行业用户超融合转型实战合集

观看视频

360 秒了解 SmartX 超融合基础设施

3 分钟技术解读 —— SMTX OS 副本分配策略

3 分钟技术解读 —— SMTX OS VM HA 高可用

SMTX 迁移工具流程讲解与操作实践

阅读博客

SmartX HCI 5.1 发布：是超融合，更是虚拟化与容器生产级统一架构

更多精彩内容请关注 SmartX 官网资源中心。

smartx
MAKE IT SIMPLE