

ZStack 技术白皮书精选

ZStack——存储模型：主存储和备份存储器

扫一扫二维码，获取更多技术干货吧



版权声明

本白皮书版权属于上海云轴信息科技有限公司，并受法律保护。转载、摘编或利用其它方式使用本调查报告文字或者观点的，应注明来源。违反上述声明者，将追究其相关法律责任。

摘要

大道至简·极速部署，ZStack 致力于产品化私有云和混合云。

ZStack 是新一代创新开源的云计算 IaaS 软件，由英特尔、微软、CloudStack 等世界上最早一批虚拟化工程师创建，拥有 KVM、Xen、Hyper-V 等成熟的技术背景。

ZStack 创新提出了云计算 4S 理念，即 Simple（简单）、Strong（健壮）、Smart（智能）、Scalable（弹性），通过全异步架构，无状态服务架构，无锁架构等核心技术，完美解决云计算执行效率低，系统不稳定，不能支撑高并发等问题，实现 HA 和轻量化管理。

ZStack 发起并维护着国内最大的自主开源 IaaS 社区——zstack.io，吸引了 6000 多名社区用户，对外公开的 API 超过 1000 个。基于这 1000 多个 API，用户可以自由组装出自己的私有云、混合云，甚至利用 ZStack 搭建公有云对外提供服务。

ZStack 拥有充足的知识产权储备，积极申报多项软著和专利，参与业内标准、白皮书的撰写，入选云计算行业方案目录，还通过了工信部云服务能力认证和信通院可信云认证。ZStack 面向企业用户提供基于 IaaS 的私有云和混合云，是业内唯一一家实现产品化，并领先业内首家推出同时打通数据面和控制面无缝混合云的云服务商。选择 ZStack，用户可以官网直接下载、1 台 PC 也可上云、30 分钟完成从裸机的安装部署。

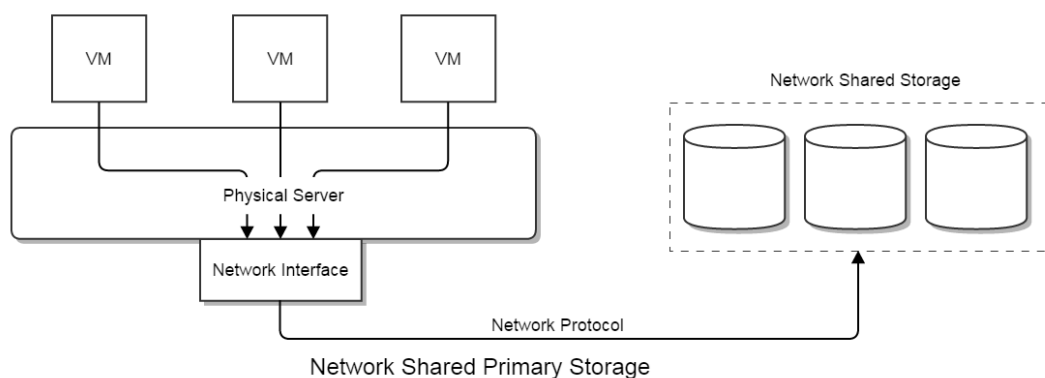
目前已有 1000 多家企业用户选择了 ZStack 云平台。

ZSTACK——存储模型：主存储和备份存储器

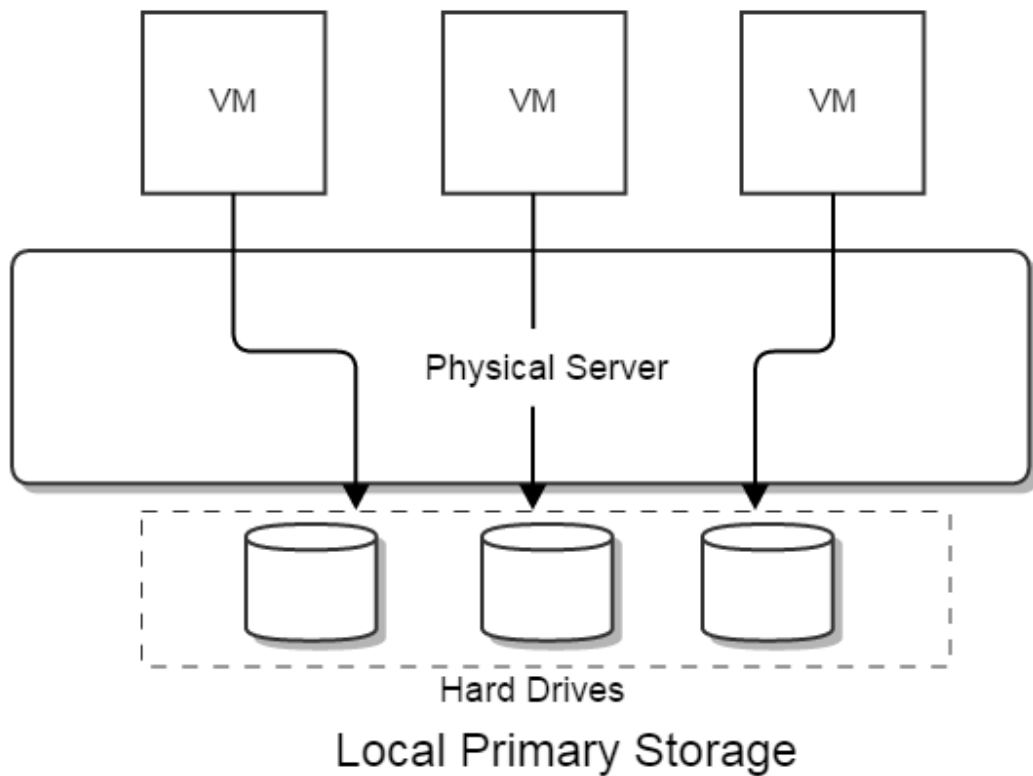
ZStack 通过逻辑功能，将存储系统抽象成主存储和备份存储。一个主存储是一个存放 VM 磁盘的存储池；一个备份存储是这么一个存储，用户存储镜像模板、备份的磁盘、快照。主存储和备份存储可以是物理分离的存储系统，也可以是同一个存储系统同时扮演两种角色。存储厂商可以轻松地，通过实现相应的存储插件，在 ZStack 中加入他们的产品。

概述

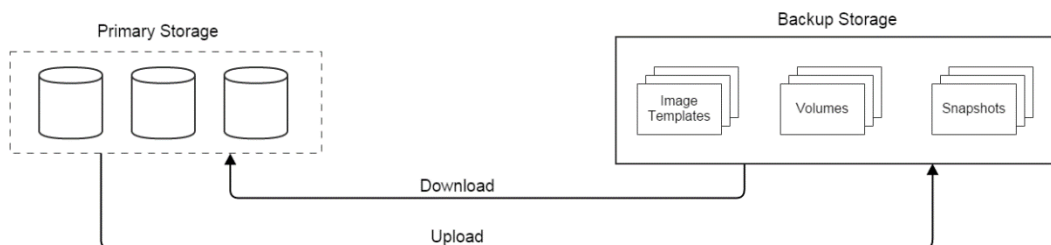
云中的存储系统可以以它们的逻辑功能被分为两类。一类作为存储池工作，存储 VM 的磁盘，并可以被运行中的 VM 访问；这类存储可以是基于文件系统的，磁盘被作为文件存储；或者基于块存储，磁盘则变成了块设备。在 ZStack 的术语表中，这类存储被称为**主存储**，要么可以是网络共享的存储，如 NFS、ISCSI：



要么是本地存储，如物理主机的硬盘：



另一类存储系统作为仓库存在，存储含有操作系统的镜像模板，以及备份的磁盘和快照；这类存储可以是基于文件系统的，实体作为文件被存储；或者是基于对象存储的，实体作为对象被存储。在 ZStack 的术语表中，这类存储被称为**备份存储**，对 VM 无法直接访问，只能是网络共享的存储：

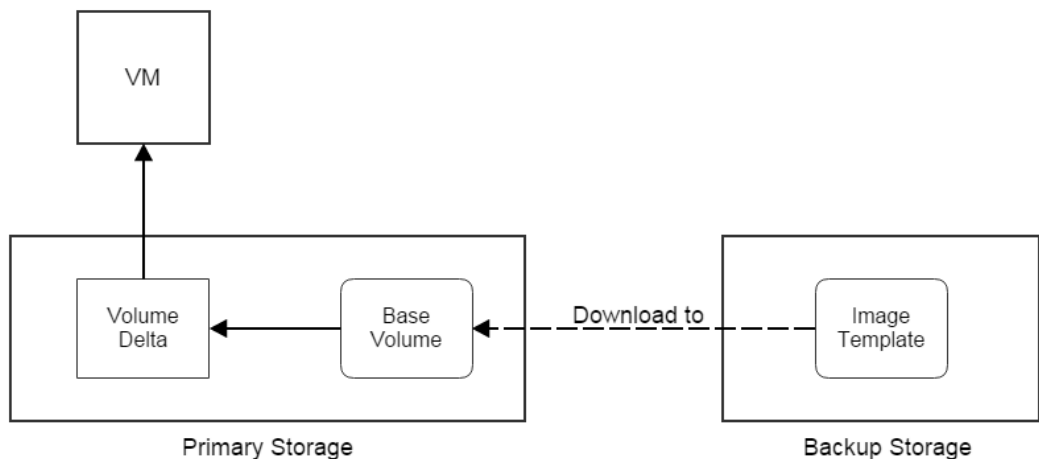


这两种存储都是逻辑概念，事实上，它们可以是各自独立的存储系统，使用不同的协议。例如，ISCSI 主存储和 NFS 备份存储。或者同一个存储系统，同时扮演两种角色。例如，ceph，它的块存储部分是用于满足主存储，而它的对象存储部分则扮演了备份存储的

角色。存储厂商可以很容易地在 ZStack 中，同时为主存储和备份存储加入他们的存储系统，通过实现存储插件的方式。

内部实现

主存储和备份存储并不是分开工作的；它们为了执行存储相关的活动，确实需要相互合作。最重要的活动是为了创建一个新的虚拟机。当一个虚拟机是第一次在一个主存储上被创建，它的镜像模板将会被从备份存储下载到主存储的镜像缓存中。由于大多数 hypervisor 使用称为[链式克隆](#)的技术，一旦镜像模板被下载，它将为所有的，使用了同样的镜像模板且在同样的主存储中有根磁盘的虚拟机，作为基础磁盘来工作。



在下载镜像之外，主存储也会上传实体，像磁盘、快照，到备份存储；这些上传活动都是备份相关的；例如，当用户备份一个数据磁盘时，数据磁盘的一个副本将会被上传到备份存储，作为一个镜像模板，可以在之后被下载到主存储用于创建新的数据磁盘。

在源代码中，主存储和备份存储在不同的插件中实现。在复杂性方面，备份存储显得更直接，因为只处理自身的事情。备份存储的主要活动是下载、上传和删除。一个备份存储需要定义一些协议，规定主存储怎样下载和上传实体，但它不需要知道主存储的细节，因为这是主存储的责任去使用这些协议来执行这些活动。另外，备份存储必须实现一些协

议，这些协议允许镜像服务注册和删除镜像模板。和所有的其他资源类似，备份存储有一个抽象的基类 BackupStorageBase, 已经实现了大多数通用的业务逻辑，存储厂商只需要实现那些和他们后台存储系统直接相关的操作，通常是通过调用 SDK 或调用 agent。

主存储更加复杂。复杂的根源来自于这么一个事实，即它的业务逻辑不只是依赖于备份存储，也依赖于 hypervisor 的细节。一个主存储，首先，必须理解备份存储的协议，以下载和上传实体；例如，一个 NFS 主存储必须知道 Sftp 备份存储，亚马逊 S3 备份存储，Swift 备份存储的信息，如果它计划支持所有的这些。另一方面，对于同一个备份存储，协议的使用方法也会随着不同的 hypervisor 而不同；例如，NFS 主存储可以调用 KVM agent 去使用 s3tool 来从亚马逊 S3 备份存储下载一个镜像模板；然而，由于 VMWare 有一个封闭的生态系统，对于 NFS 主存储来说要做同样事情的唯一方式是通过 VMWare 的 SDK。基于这些事实，主存储的复杂性是 $M \times N$ ，其中 M 是备份存储的种类， N 是它所支持的 hypervisor 的种类。

正如 [ZStack—通用插件系统](#) 一文中所描述的, ZStack 是一个插件系统，每一个特性都被做成一个小的插件；一个主存储需要定义两个接口来打破这个复杂性。第一个是一个 hypervisor 的后端，用于处理只和 hypervisor 有关的活动；例如，NFS 主存储有个定义好的接口：NfsPrimaryStorageBackend，对每一个支持的 hypervisor，都会有一个具体的类，类似 NfsPrimaryStorageKVMBackend 用于 KVM。第二个，称之为 PrimaryToBackupStorageMediator，是一个 *hypervisor 到备份存储* 的后端，用于处理同时涉及到 hypervisor 和备份存储的后端；例如，Nfs 主存储有一个 NfsPrimaryToSftpBackupKVMBackup 的实现，用于为 KVM 支持 Sftp 备份存储。

这听起来非常糟糕，因为一个主存储必须实现如此多的东西；然而，事实上，一个主存储可能不需要去为所有的 hypervisor 支持所有的备份存储；例如，为 VMWare 支持 Sftp 备份存储是毫无意义的，因为 VMWare SDK 没有可能允许用 scp 传输一个文件到它的存储仓（即使可以通过绕过 SDK 使得这成为可能，我们不把它视为一种可靠的方式）。而且网络共享存储上，流行的协议并不特别多，大多数的使用场景可以被处理，一旦我们把 Nfs 主存储和 Iscsi 主存储准备就位。

注意: 在当前的 ZStack 版本中 (0.6), 只有 Nfs 主存储和 Sftp 备份存储被实现了。

总结

在这篇文章中, 我们演示了 ZStack 的存储模型。通过以逻辑功能将存储划分成主存储和备份存储, ZStack 提供了一个非常棒的灵活性, 使得存储厂商可以选择性地以各种意图插入他们的存储系统。而且随着越来越的普遍的存储协议, 比如 NFS、ISCSI、S3、Swift, 将被作为默认插件加入, 用户将不需要忧虑他们是否能够为他们现存的存储系统找到合适的组合。