



SANGFOR
深信服科技

深信服 aBos 一体机

技术白皮书

深信服科技股份有限公司
2017年3月

版权声明

深信服科技股份有限公司版权所有，并保留对本文档及本声明的最终解释权和修改权。

本文档中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除另有特别注明外，其著作权或其它相关权利均属于深信服科技股份有限公司。未经深信服科技股份有限公司书面同意，任何人不得以任何方式或形式对本文档内的任何部分进行复制、摘录、备份、修改、传播、翻译成其他语言、将其全部或部分用于商业用途。

免责条款

本文档仅用于为最终用户提供信息，其内容如有更改，恕不另行通知。

深信服科技股份有限公司在编写本文档的时候已尽最大努力保证其内容准确可靠，但深圳市深信服科技股份有限公司不对本文档中的遗漏、不准确、或错误导致的损失和损害承担责任。

信息反馈

如果您有任何宝贵意见，请反馈至：

信箱：广东省 深圳市 学苑大道 1001 号南山智园 A1 栋

邮编：518055

电话：0755-26581949

传真：0755-26581959

您也可以访问深信服科技网站：www.sangfor.com.cn 获得技术和产品信息

缩写和约定

英文缩写	英文全称	中文解释
Hypervisor	Hypervisor	虚拟机管理器（和 VMM 同义）
VMM	VMM Virtual Machine Manager	虚拟机监视器
HA	HighAvailability	高可用性
vMotion	vMotion	实时迁移
DRS	Distributed Resource Scheduler	分布式资源调度
RAID	Redundant Arrays of Independent Disks	磁盘阵列
IOPS	Input/Output Operations Per Second	每秒读写（I/O）操作的次数
VM	Virtual Machine	虚拟机
SDN	Software Defined Network	软件定义网络

NFV	Network Function Virtualization	网络功能虚拟化
-----	------------------------------------	---------

修订记录

修订版本号	作者	日期	备注
V1.0	黄航	2016-10	
V2.0	黄航	2016-12	
V3.0	黄航	2017-3	

目 录

1	前言	2
1.1	边缘 IT 时代变革.....	2
1.2	白皮书总览.....	2
2	深信服 aBos 一体机架构技术	4
2.1	aBos 超融合架构概述.....	4
2.1.1	aBos 超融合架构的定义.....	4
2.2	深信服 aBos 超融合架构组成模块.....	4
2.2.1	系统总体架构.....	4
2.3	aSV 计算虚拟化.....	4
2.3.1	计算虚拟化概述.....	4
2.3.2	aSV 技术原理.....	5
2.3.3	深信服 aSV 的技术特性.....	14
2.3.4	aSV 的特色技术.....	17
2.4	aNet 网络设备虚拟化.....	18
2.4.1	网络设备虚拟化概述.....	18
2.4.2	aNET 网络虚拟化技术原理.....	19
2.4.3	aNet 功能特性.....	22
2.4.4	深信服 aNet 的特色技术.....	28
2.5	aSAN 存储虚拟化.....	29
2.5.1	存储虚拟化概述.....	29
2.5.2	aSAN 技术原理.....	29
2.5.3	aSAN 存储数据可靠性保障.....	38
2.5.4	深信服 aSAN 功能特性.....	38
3	深信服 aBos 一体机核心价值	40
3.1	高稳定性.....	40
3.2	简化分支机构 IT.....	40
3.3	简化运维、集中管理.....	40
3.4	灵活部署、扩展性好.....	40
4	超融合架构最佳实践	41

1 前言

1.1 边缘 IT 时代变革

随着在物联网的构建过程中传感器的大量使用，企业客户及员工使用的移动设备和云服务的大量增加，企业边缘的概念也正在被重新定义。对于一个企业来说，它的新的边缘包括了下属办事机构、商店、机场、商场、公共图书馆、运动场以及其它公共空间。任何可能有顾客的地方都在迅速成为企业的边缘。不仅是企业边缘的概念在变化，与此同时，在这些地点的顾客参与行为也在发生本质性的变化。

近年来，随着主机托管、代管和云服务被越来越多地采用，IT 基础架构也呈现出融合化和集中化的趋势。而对企业边缘的重新定义则将在某种程度上扭转这一趋势，因为计算、存储和网络这三大功能都被推向了新的企业边缘，以解决那些具有较高处理要求和带宽要求的工作负载。未来，微型数据中心将有望重新崛起，但届时它不会采取目前的这种形式了。企业边缘的变化还将促进 IT 与非 IT 架构/设备的持续创新和整合。服务供应商将其 IT 资产部署在客户和合作伙伴所在的地理位置上这一举措的逐渐增长的可能性，将给双方的 IT 团队都带来一系列的新挑战。

我们现在正处于一场几十年未见的分支机构边缘数据中心革命性转变中，究其核心，这一转变是由“软件”基础设施的崛起而驱动。虚拟网络设备、虚拟机、存储设备能够以高速自动化的方式分配与重新配置，不会受到分支非动态设置的硬件基础设施的限制，在“软件定义广域网”的模型下，用户首先考虑的是分支应用，根据应用的模式便可灵活的调配其所需的 IT 基础架构资源，也就是通过软件化的方式实现分支硬件资源调配。

深信服 aBos 一体机是软件定义分支 IT 基础架构一套非常成熟的解决方案，是建立一个各个软件组件间相对独立、松耦合的软件系统，极大提高分支整个 IT 基础架构系统的可靠性和可扩展性。除满足上面所述的边缘 IT 架构虚拟化，标准化和自动化诉求外，秉承深信服公司产品的优秀基因，向您提供简单易用，安全可靠，易交付的产品。

1.2 白皮书总览

本书介绍的内容大致如下：

第一章、在前言部分，给您对企业边缘 IT 变革，软件定义广域网有一个概括性的认识，并对本文档的阅读给出指导。

第二章、讲述 aBos 一体机各个功能模块的技术细节。

第三章、介绍深信服 aBos 一体机涵盖的技术。

第三章、向您介绍深信服 aBos 一体机中的技术在为客户带来的核心价值。

第四章、分享 aBos 一体机在客户中的实际应用场景，并给出深信服超融合架构产品的体验途径，非常欢迎您来试用。

2 深信服 aBos 一体机架构技术

2.1 aBos 一体机架构概述

2.1.1 aBos 一体机架构的定义

深信服 aBos 一体机解决方案，是一种将网络设备、计算、存储等资源作为基本组成元素，通过一体机的方式承载中小型或者分支机构的 IT 网络建设技术。满足多种类型的小型机构业务办公需求，提供具备业务上线速度快、分支安全性考虑全面、业务发展扩展性高、集中管理的一站式分支机构解决方案，实现分支机构“ZERO IT”。

2.2 深信服 aBos 一体机架构组成模块

2.2.1 系统总体架构



深信服 aBos 一体机架构图

深信服的 aBos 一体机解决方案软件架构主要包含三大组件（网络设备虚拟化、服务器虚拟化、存储虚拟化）、一个 WEB 控制平台（虚拟化管理平台 VMP）、总部集中管理（BBC 管理中心）。硬件架构上，可以通过一体机的方式实现开机即用，一体机服务器实现基础架构的承载

后续章节，会针对 aBos 超融合架构中的三大功能模块：计算虚拟化（aSV）、网络设备虚拟化（aNET）、存储虚拟化（aSAN）所涵盖的产品技术来做详细说明。

2.3 aSV 计算虚拟化

2.3.1 计算虚拟化概述

计算资源虚拟化技术是将通用的 x86 服务器经过虚拟化软件，对最终用户呈现标准的虚拟机。这些虚拟机就像同一个厂家生产的系列化的产品一样，具备系列化的硬件配置，使用相同的驱动程序。

虚拟机的定义：虚拟机 (Virtual Machine) 是由虚拟化层提供的高效、独立的虚拟计算机系统，每台虚拟机都是一个完整的系统，它具有处理器、内存、网络设备、存储设备和 BIOS，因此操作系统和应用程序在虚拟机中的运行方式与它们在物理服务器上的运行方式没有什么区别。

虚拟机与物理服务器相比：虚拟机不是由真实的电子元件组成，而是由一组虚拟组件（文件）组成，这些虚拟组件与物理服务器的硬件配置无关，关键与物理服务器相比，虚拟机具有以下优势：

抽象解耦

1. 可在任何 X86 架构的服务器上运行；
2. 上层应用操作系统不需修改即可运行；

分区隔离

1. 可与其他虚拟机同时运行；
2. 实现数据处理、网络连接和数据存储的安全隔离；

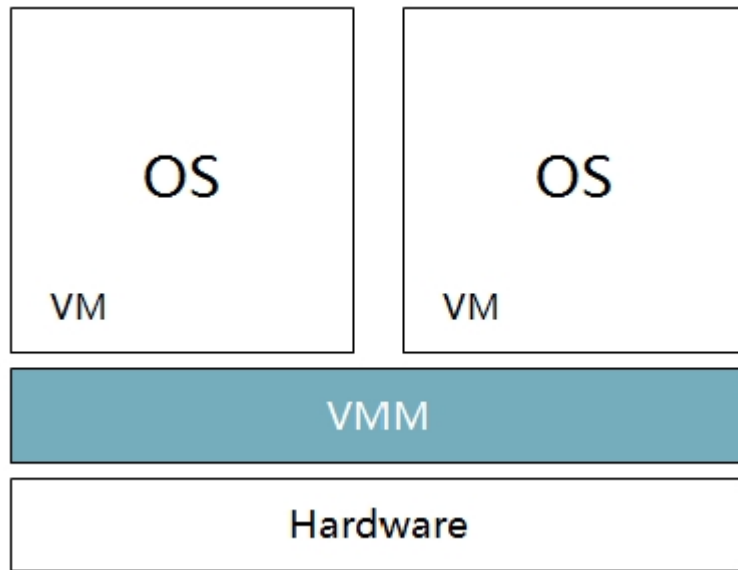
封装移动

1. 可封装于文件之中，通过简单的文件复制实现快速部署、备份及还原；
2. 可便捷地将整个系统（包括虚拟硬件、操作系统和配置好的应用程序）在不同的物理服务器之间进行迁移，甚至可以在虚拟机正在运行的情况下进行迁移；

深信服的 aBos 一体机解决方案中的计算虚拟化采用 aSV 虚拟化系统，通过将服务器资源虚拟化为多台虚拟机。最终用户可以在这些虚拟机上安装各种软件，挂载磁盘，调整配置，调整网络，就像普通的 x86 服务器一样使用它。

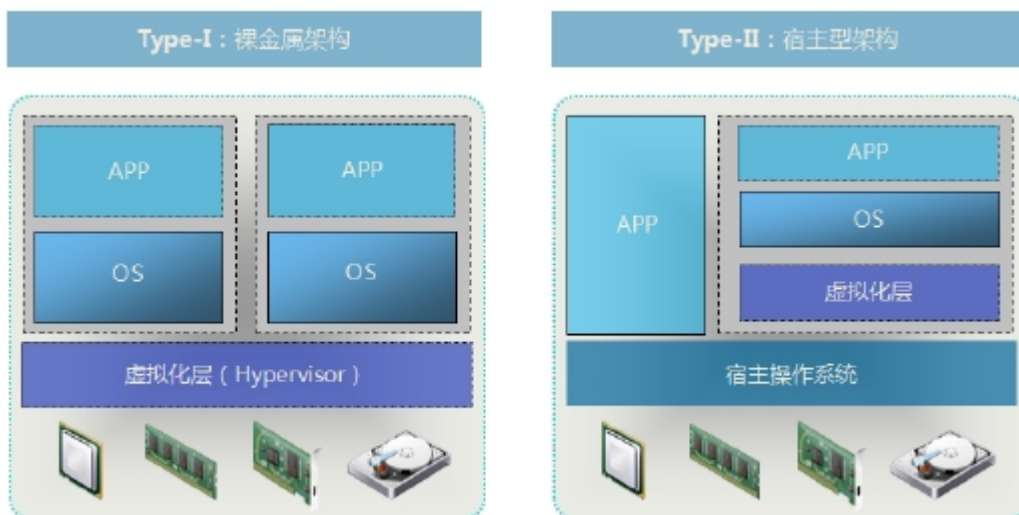
2.3.2 aSV 技术原理

2.3.2.1 Hypervisor 架构



Hypervisor 是一种运行在物理服务器和操作系统之间的中间软件层,可允许多个操作系统和应用共享一套基础物理硬件,因此也可以看作是虚拟环境中的“元”操作系统,它可以协调访问服务器上的所有物理设备和虚拟机,也叫虚拟机监视器 (Virtual Machine Monitor)。

Hypervisor 是所有虚拟化技术的核心。非中断地支持多工作负载迁移的能力是 Hypervisor 的基本功能。当服务器启动并执行 Hypervisor 时,它会给每一台虚拟机分配适量的内存、CPU、网络和磁盘,并加载所有虚拟机的客户操作系统。



虚拟化技术架构

Hypervisor, 常见的 Hypervisor 分两类:

Type-I (裸金属型)

指 VMM 直接运作在裸机上,使用和管理底层的硬件资源, GuestOS 对真实硬件资

源的访问都要通过 VMM 来完成，作为底层硬件的直接操作者，VMM 拥有硬件的驱动程序。裸金属虚拟化中 Hypervisor 直接管理调用硬件资源，不需要底层操作系统，也可以理解为 Hypervisor 被做成了一个很薄的操作系统。这种方案的性能处于主机虚拟化与操作系统虚拟化之间。代表是 VMware ESX Server、Citrix XenServer 和 Microsoft Hyper-V, Linux KVM。

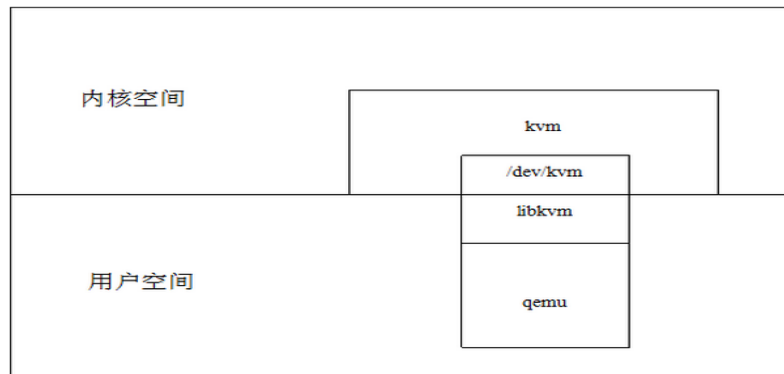
Type-II 型（宿主型）

指 VMM 之下还有一层宿主操作系统，由于 Guest OS 对硬件的访问必须经过宿主操作系统，因而带来了额外的性能开销，但可充分利用宿主操作系统提供的设备驱动和底层服务来进行内存管理、进程调度和资源管理等。主机虚拟化中 VM 的应用程序调用硬件资源时需要经过:VM 内核->Hypervisor->主机内核，导致性能是三种虚拟化技术中最差的。主机虚拟化技术代表是 VMware Server (GSX)、Workstation 和 Microsoft Virtual PC、Virtual Server 等。

由于主机型 Hypervisor 的效率问题，深信服的 aSV 采用了裸机型 Hypervisor 中的 Linux KVM 虚拟化，即为 Type-I（裸金属型）。

KVM(Kernel-based Virtual Machine)是基于 linux 内核虚拟化技术，自 linux 2.6.20 之后就集成在 linux 的各个主要发行版本中。它使用 linux 自身的调度器进行管理，所以相对于 xen，其核心源码很少。

KVM 是基于硬件虚拟化扩展 (Intel VT-X) 和 QEMU 的修改版，KVM 属于 Linux kernel 的一个模块，可以用命令 modprobe 去加载 KVM 模块。加载了该模块后，才能进一步通过工具创建虚拟机。但是仅有 KVM 模块是不够的。因为用户无法直接控制内核去做事情，还必须有一个运行在用户空间的工具才行。这个用户空间的工具，我们选择了已经成型的开源虚拟化软件 QEMU，QEMU 也是一个虚拟化软件，它的特点是可虚拟不同的 CPU，比如说在 x86 的 CPU 上可虚拟一个 power 的 CPU，并可利用它编译出可运行在 power 上的 CPU，并可利用它编译出可运行在 power 上的程序。KVM 使用了 QEMU 的一部分，并稍加改造，就成了可控制 KVM 的用户空间工具了。这就是 KVM 和 QEMU 的关系。如下图：



一个普通的 linux 进程有两种运行模式：内核和用户。而 KVM 增加了第三种模式：客户模式（有自己的内核和用户模式）。在 kvm 模型中，每一个虚拟机都是由 linux 调度程序管理的标准进程。

总体来说，kvm 由两个部分组成：一个是管理虚拟硬件的设备驱动，该驱动使用字符设备/dev/kvm 作为管理接口；另一个是模拟 PC 硬件的用户空间组件，这是一个稍作修改的 qemu 进程。

同时，aSV 采用 KVM 优势有：

- 嵌入到 Linux 正式 Kernel (提高兼容性)
- 代码级资源调用（提高性能）
- 虚拟机就是一个进程（内存易于管理）
- 直接支持 NUMA 技术（提高扩展性）
- 保持开源发展模式（强大的社区支持）

2.3.2.2 aSV 的 Hypervisor 实现

VMM (Virtual Machine Monitor)对物理资源的虚拟可以划分为三个部分：CPU 虚拟化、内存虚拟化和 I/O 设备虚拟化，其中以 CPU 的虚拟化最为关键。

经典的虚拟化方法：现代计算机体系结构一般至少有两个特权级（即用户态和核心态，x86 有四个特权级 Ring0~ Ring3）用来分隔系统软件和应用软件。那些只能在处理器的最高特权级（内核态）执行的指令称之为特权指令，一般可读写系统关键资源的指令（即敏感指令）决大多数都是特权指令（X86 存在若干敏感指令是非特权指令的情况）。如果执行特权指令时处理器的状态不在内核态，通常会引发一个异常而交由系统软件来处理这个非法访问（陷入）。经典的虚拟化方法就是使用“特权解除”和“陷入-模拟”的方式，即将 GuestOS 运行在非特权级，而将 VMM 运行于最高特权级（完全控制系统资源）。解除了 GuestOS 的特权级后，Guest OS 的大部分指令仍可以在硬件上直接运行，只有执行到特权指令时，才会陷入到 VMM 模拟执行（陷入-模拟）。“陷入-模拟”的本质是保证可能影响 VMM 正确运行的指令由 VMM 模拟执行，

大部分的非敏感指令还是照常运行。

因为 X86 指令集中有若干条指令是需要被 VMM 捕获的敏感指令，但是却不是特权指令（称为临界指令），因此“特权解除”并不能导致他们发生陷入模拟，执行它们不会发生自动的“陷入”而被 VMM 捕获，从而阻碍了指令的虚拟化，这也称之为 X86 的虚拟化漏洞。

X86 架构虚拟化的实现方式可分为：

1、X86 “全虚拟化”（指所抽象的 VM 具有完全的物理机特性，OS 在其上运行不需要任何修改）Full 派秉承无需修改直接运行的理念，对“运行时监测，捕捉后模拟”的过程进行优化。该派内部之实现又有些差别，其中以 VMWare 为代表的基于二进制翻译（BT）的全虚拟化为代表，其主要思想是在执行时将 VM 上执行的 Guest OS 指令，翻译成 x86 指令集的一个子集，其中的敏感指令被替换成陷入指令。翻译过程与指令执行交叉进行，不含敏感指令的用户态程序可以不经翻译直接执行。

2、X86 “半虚拟化”（指需 OS 协助的虚拟化，在其上运行的 OS 需要修改）半虚拟化的基本思想是通过修改 Guest OS 的代码，将含有敏感指令的操作，替换为对 VMM 的超调用 Hypercall，类似 OS 的系统调用，将控制权转移到 VMM，该技术因 VMM 项目而广为人知。该技术的优势在于 VM 的性能能接近于物理机，缺点在于需要修改 GuestOS（如：Windows 不支持修改）及增加的维护成本，关键修改 Guest OS 会导致操作系统对特定 hypervisor 的依赖性，因此很多虚拟化厂商基于 VMM 开发的虚拟化产品部分已经放弃了 Linux 半虚拟化，而专注基于硬件辅助的全虚拟化开发，来支持未经修改的操作系统。

3、X86 “硬件辅助虚拟化”：

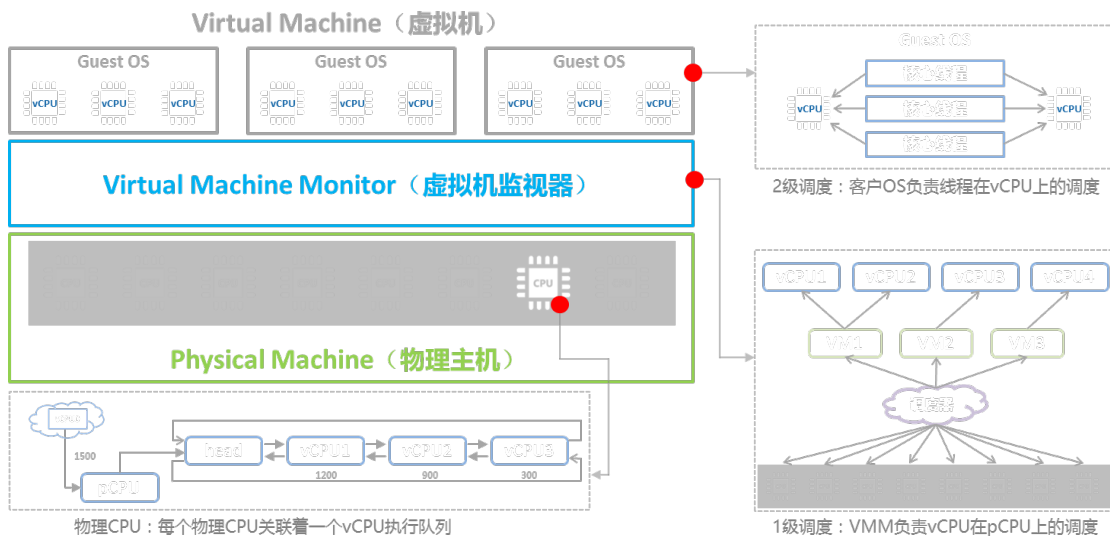
其基本思想就是引入新的处理器运行模式和新的指令，使得 VMM 和 Guest OS 运行于不同的模式下，Guest OS 运行于受控模式，原来的一些敏感指令在受控模式下全部会陷入 VMM，这样就解决了部分非特权的敏感指令的“陷入-模拟”难题，而且模式切换时上下文的保存恢复由硬件来完成，这样就大大提高了“陷入-模拟”时上下文切换的效率。

以 Intel VT-x 硬件辅助虚拟化技术为例，该技术增加了在虚拟状态下的两种处理器工作模式：根（Root）操作模式和非根（Non-root）操作模式。VMM 运作在 Root 操作模式下，而 Guest OS 运行在 Non-root 操作模式下。这两个操作模式分别拥有自己的特权级环，VMM 和虚拟机的 Guest OS 分别运行在这两个操作模式的 0 环。这样，既能使 VMM 运行在 0 环，也能使 Guest OS 运行在 0 环，避免了修改 Guest OS。Root 操作模式和 Non-root 操作模式的切换是通过新增的 CPU 指令（如：

VMXON, VMXOFF) 来完成。

硬件辅助虚拟化技术消除了操作系统的 ring 转换问题，降低了虚拟化门槛，支持任何操作系统的虚拟化而无须修改 OS 内核，得到了虚拟化软件厂商的支持。硬件辅助虚拟化技术已经逐渐消除软件虚拟化技术之间的差别，并成为未来的发展趋势。

● vCPU 机制



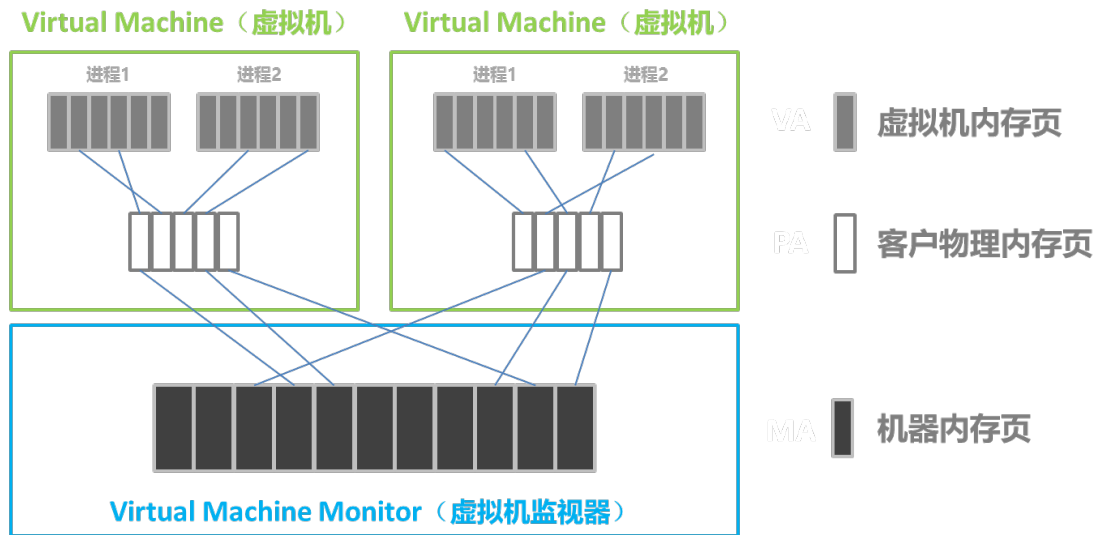
vCPU 调度机制

对虚拟机来说，不直接感知物理 CPU，虚拟机的计算单元通过 vCPU 对象来呈现。虚拟机只看到 VMM 呈现给它的 vCPU。在 VMM 中，每个 vCPU 对应一个 VMCS (Virtual-MachineControl Structure) 结构，当 vcpu 被从物理 CPU 上切换下来的时候，其运行上下文会被保存在其对应的 VMCS 结构中；当 vcpu 被切换到 pcpu 上运行时，其运行上下文会从对应的 VMCS 结构中导入到物理 CPU 上。通过这种方式，实现各 vCPU 之间的独立运行。

从虚拟机系统的结构与功能划分可以看出，客户操作系统与虚拟机监视器共同构成了虚拟机系统的两级调度框架，如图所示是一个多核环境下虚拟机系统的两级调度框架。客户操作系统负责第 2 级调度，即线程或进程在 vCPU 上的调度（将核心线程映射到相应的虚拟 CPU 上）。虚拟机监视器负责第 1 级调度，即 vCPU 在物理处理单元上的调度。两级调度的调度策略和机制不存在依赖关系。vCPU 调度器负责物理处理器资源在各个虚拟机之间的分配与调度，本质上即把各个虚拟机中的 vCPU 按照一定的策略和机制调度在物理处理单元上可以采用任意的策略来分配物理资源，满足虚拟机的不同需求。vCPU 可以调度在一个或多个物理处理单元执行（分时复用或空间复用

物理处理单元)，也可以与物理处理单元建立一对一固定的映射关系（限制访问指定的物理处理单元）。

内存虚拟化



内存虚拟化三层模型

因为 VMM (Virtual Machine Monitor) 掌控所有系统资源，因此 VMM 握有整个内存资源，其负责页式内存管理，维护虚拟地址到机器地址的映射关系。因 Guest OS 本身亦有页式内存管理机制，则有 VMM 的整个系统就比正常系统多了一层映射：

- A. 虚拟地址(VA)，指 Guest OS 提供给其应用程序使用的线性地址空间；
- B. 物理地址(PA)，经 VMM 抽象的、虚拟机看到的伪物理地址；
- C. 机器地址(MA)，真实的机器地址，即地址总线上出现的地址信号；

映射关系如下： $Guest\ OS: PA = f(VA)$ 、 $VMM: MA = g(PA)$ VMM 维护一套页表，负责 PA 到 MA 的映射。Guest OS 维护一套页表，负责 VA 到 PA 的映射。实际运行时，用户程序访问 VA1，经 Guest OS 的页表转换得到 PA1，再由 VMM 介入，使用 VMM 的页表将 PA1 转换为 MA1。

● **页表虚拟化技术**

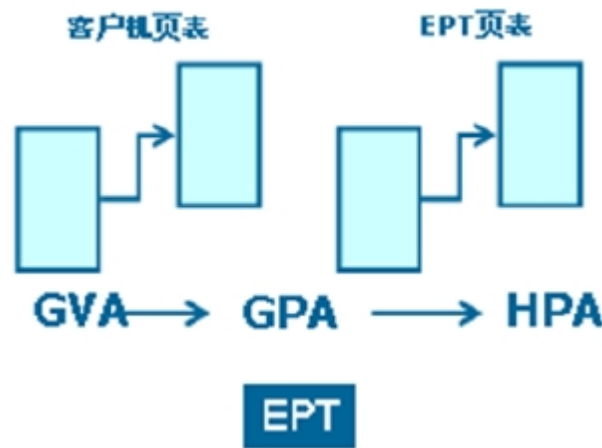
普通 MMU 只能完成一次虚拟地址到物理地址的映射，在虚拟机环境下，经过 MMU 转换所得到的“物理地址”并不是真正的机器地址。若需得到真正的机器地址，必须由 VMM 介入，再经过一次映射才能得到总线上使用的机器地址。如果虚拟机的每个内存访问都需要 VMM 介入，并由软件模拟地址转换的效率是很低下的，几乎不具有实际可用性，为实现虚拟地址到机器地址的高效转换，现普遍采用的思想是：由 VMM 根据映射 f 和 g 生成复合的映射 fg，并将这个映射关系写入 MMU。当前采用的页表虚拟化方法主要是 MMU 类虚拟化 (MMU Paravirtualization) 和影子页表，后者已被

内存的硬件辅助虚拟化技术所替代。

1、MMU Paravirtualization

其基本原理是：当 Guest OS 创建一个新的页表时，会从它所维护的空闲内存中分配一个页面，并向 VMM 注册该页面，VMM 会剥夺 Guest OS 对该页表的写权限，之后 GuestOS 对该页表的写操作都会陷入到 VMM 加以验证和转换。VMM 会检查页表中的每一项，确保他们只映射了属于该虚拟机的机器页面，而且不得包含对页表页面的可写映射。后 VMM 会根据自己所维护的映射关系，将页表项中的物理地址替换为相应的机器地址，最后再把修改过的页表载入 MMU。如此，MMU 就可以根据修改过页表直接完成虚拟地址到机器地址的转换。

2、内存硬件辅助虚拟化



内存硬件辅助虚拟化技术原理图

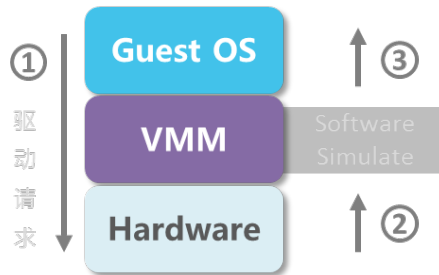
内存的硬件辅助虚拟化技术是用于替代虚拟化技术中软件实现的“影子页表”的一种硬件辅助虚拟化技术，其基本原理是：GVA（客户操作系统的虚拟地址）→ GPA（客户操作系统的物理地址）→ HPA（宿主操作系统的物理地址）两次地址转换都由 CPU 硬件自动完成（软件实现内存开销大、性能差）。以 VT-x 技术的页表扩充技术 Extended PageTable（EPT）为例，首先 VMM 预先把客户机物理地址转换到机器地址的 EPT 页表设置到 CPU 中；其次客户机修改客户机页表无需 VMM 干预；最后，地址转换时，CPU 自动查找两张页表完成客户机虚拟地址到机器地址的转换。使用内存的硬件辅助虚拟化技术，客户机运行过程中无需 VMM 干预，去除了大量软件开销，内存访问性能接近物理机。

● I/O 设备虚拟化

VMM 通过 I/O 虚拟化来复用有限的外设资源，其通过截获 Guest OS 对 I/O 设备的访问请求，然后通过软件模拟真实的硬件，目前 I/O 设备的虚拟化方式主要有三

种：设备接口完全模拟、前端 / 后端模拟、直接划分。

1、设备接口完全模拟：



即软件精确模拟与物理设备完全一样的接口，Guest OS 驱动无须修改就能驱动这个虚拟设备。

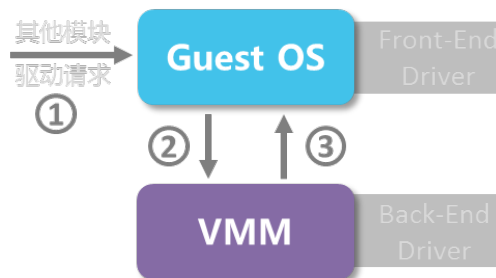
优点：没有额外的硬件开销，可重用现有驱动程序；

缺点：为完成一次操作要涉及到多个寄存器的操作，使得 VMM 要截获每个寄存器访

问并进行相应的模拟，这就导致多次上下文切换；由于是软件模拟，性能较低。

2、前端 / 后端模拟：

VMM 提供一个简化的驱动程序（后端，Back-End），Guest OS 中的驱动程序为前端(Front-End, FE)，前端驱动将来自其他模块的请求通过与 Guest OS 间的特殊通信机制直接发送给 Guest OS 的后端驱动，后端驱动在处理完请求后再发回通知给前端，VMM 即采用该方法。

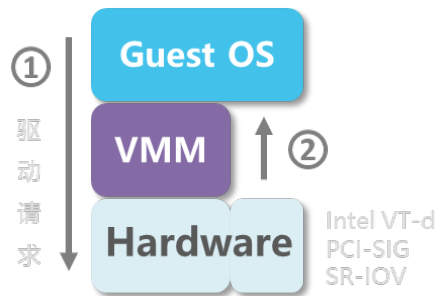


优点：基于事务的通信机制，能在很大程度上减少上下文切换开销，没有额外的硬件开销；

缺点：需要 GuestOS 实现前端驱动，后端驱动可能成为瓶颈。

3、直接划分：

即直接将物理设备分配给某个 Guest OS，由 Guest OS 直接访问 I/O 设备（不经 VMM），目前与此相关的技术有 IOMMU（Intel VT-d, PCI-SIG 之 SR-IOV 等），旨在建立高效的 I/O 虚拟化直通通道。



优点：可重用已有驱动，直接访问减少了虚拟化开销；

缺点：需要购买较多额外的硬件。

2.3.3 深信服 aSV 的技术特性

2.3.3.1 内存 NUMA 技术

非统一内存访问（NUMA）是服务器 CPU 和内存设计的新架构。传统的服务器架构下把内存放到单一的存储池中，这对于单处理器或单核心的系统工作良好。但是这种传统的统一访问方式，在多核心同时访问内存空间时会导致资源争用和性能问题。毕竟，CPU 应该可以访问所有的服务器内存，但是不需要总是保持占用。实际上，CPU 仅需要访问工作负载实际运行时所需的内存空间就可以了。

因此 NUMA 改变了内存对 CPU 的呈现方式。这是通过对服务器每个 CPU 的内存进行分区来实现的。每个分区（或内存块）称为 NUMA 节点，而和该分区相关的处理器可以更快地访问 NUMA 内存，而且不需要和其它的 NUMA 节点争用服务器上的资源（其它的内存分区分配给其它处理器）。

NUMA 的概念跟缓存相关。处理器的速度要比内存快得多，因此数据总是被移动到更快的本地缓存，这里处理器访问的速度要比通用内存快得多。NUMA 本质上为每个处理器配置了独有的整体系统缓存，减少了多处理器试图访问统一内存空间时的争用和延迟。

NUMA 与服务器虚拟化完全兼容，而且 NUMA 也可以支持任意一个处理器访问服务器上的任何一块内存区域。某个处理器当然可以访问位于不同区域上的内存数据，但是需要更多本地 NUMA 节点之外的传输，并且需要目标 NUMA 节点的确认。这增加了整体开销，影响了 CPU 和内存子系统的性能。

NUMA 对虚拟机负载不存在任何兼容性问题，但是理论上虚拟机最完美的方式应该是在某个 NUMA 节点内。这可以防止处理器需要跟其它的 NUMA 节点交互，从而导致工作负载性能下降。

深信服的 aSV 支持 NUMA 技术，使得 hypervisor 和上层 OS 内存互连，这样 OS 不会在 CPU 和 NUMA 节点之间迁移工作负载。

2.3.3.2 SR-IOV

通常针对虚拟化服务器的技术是通过软件模拟共享和虚拟化网络适配器的一个物理端口，以满足虚拟机的 I/O 需求，模拟软件的多个层为虚拟机作了 I/O 决策，因此导致环境中出现瓶颈并影响 I/O 性能。aSV 虚拟化平台提供的 SR-IOV 是一种不需要软件模拟就可以共享 I/O 设备 I/O 端口的物理功能的方法，主要利用 iNIC 实现网桥卸载虚拟网卡，允许将物理网络适配器的 SR-IOV 虚拟功能直接分配给虚拟机，可以提高网络吞吐量，并缩短网络延迟，同时减少处理网络流量所需的主机 CPU 开销。

技术原理：SR-IOV (Single Root I/O Virtualization) 是 PCI-SIG 推出的一项标准，是虚拟通道（在物理网卡上对上层软件系统虚拟出多个物理通道，每个通道具备独立的 I/O 功能）的一个技术实现，用于将一个 PCIe 设备虚拟成多个 PCIe 设备，每个虚拟 PCIe 设备如同物理 PCIe 设备一样向上层软件提供服务。通过 SR-IOV 一个 PCIe 设备不仅可以导出多个 PCI 物理功能，还可以导出共享该 I/O 设备上的资源的一组虚拟功能，每个虚拟功能都可以被直接分配到一个虚拟机，能够让网络传输绕过软件模拟层，直接分配到虚拟机，实现了将 PCI 功能分配到多个虚拟接口以在虚拟化环境中共享一个 PCI 设备的目的，并且降低了软加模拟层中的 I/O 开销，因此实现了接近本机的性能。如图所示，在这个模型中，不需要任何透传，因为虚拟化在终端设备上发生，允许管理程序简单地将虚拟功能映射到 VM 上以实现本机设备性能和隔离安全。SR-IOV 虚拟出的通道分为两个类型：

1、PF (Physical Function) 是完整的 PCIe 设备，包含了全面的管理、配置功能，Hypervisor 通过 PF 来管理和配置网卡的所有 I/O 资源。

2、VF (Virtual Function) 是一个简化的 PCIe 设备，仅仅包含了 I/O 功能，通过 PF 衍生而来好象物理网卡硬件资源的一个切片，对于 Hypervisor 来说，这个 VF 同一块普通的 PCIe 网卡一模一样。

通过 SR-IOV 可满足高网络 IO 应用要求，无需特别安装驱动，且无损热迁移、内存复用、虚拟机网络管控等虚拟化特性。

2.3.3.3 Fake-raid

一般情况下，当主机系统有多块硬盘时，通过组建 Raid 以提升磁盘性能或提供磁盘冗余，往往成为人们的首选考量。当今主流 raid 实现方案大致可分为三种：

硬件 raid (hardware raid)：通过购买昂贵的 raid 卡实现。

软件 raid (software raid)：通过操作系统内软件创建阵列，raid 处理开销由 CPU 负责。

主板 raid (fake raid)：通过主板内建 raid 控制器创建阵列，由操作系统驱动识

别。

相对于昂贵的硬件，主板 raid(fake raid)就成了我们不错的选择。Fake raid 仅提供廉价的控制器，raid 处理开销仍由 CPU 负责，因此性能与 CPU 占用基本与 software raid 持平。

aSV 3.7 融入了对 Fake-RAID 的支持，现可支持 Fake-RAID 安装与使用 Fake-RAID 存储，目前可以使用 intel 模式的 raid0, raid1, raid5, raid10, LSI 模式的 raid0

2.3.3.4 虚拟机生命周期管理

aSV 提供了虚拟机从创建至删除整个过程中的全面管理，就像人类的生命周期一样，虚拟机最基本的生命周期就是创建、使用和删除这三个状态。当然还包含如下几个状态：

- 创建虚拟机
- 虚拟机开关机、重启、挂起
- 虚拟机上的操作系统安装
- 创建模板
- 更新虚拟机硬件配置
- 迁移虚拟机及/或虚拟机的存储资源
- 分析虚拟机的资源利用情况
- 虚拟机备份
- 虚拟机恢复
- 删除虚拟机

在虚拟机生命周期内，虚拟机可能会在某一个时间点经历上述这些状态。aSV 提供了完善的虚拟机生命周期管理工具，我们可以通过对虚拟机生命周期的规划，可以想要最大化的发挥虚拟机的作用。

2.3.3.5 虚拟机热迁移

虚拟化环境中，物理服务器和存储上承载更多的业务和数据，设备故障时造成的影响更大。aSV 虚拟化平台提供虚拟机热迁移技术，降低宕机带来的风险、减少业务中断的时间。

aSV 虚拟机热迁移技术是指把一个虚拟机从一台物理服务器迁移到另一台物理服务器上，即虚拟机保存/恢复(Save/Restore)。首先将整个虚拟机的运行状态完整保存下来，同时可以快速的恢复到目标硬件平台上，恢复以后虚拟机仍旧平滑运行，用户不会察觉到任何差异。虚拟机的热迁移技术主要被用于双机容错、负载均衡和节能降

耗等应用场景。aSV 虚拟化平台热迁移提供内存压缩技术，使热迁移效率提升一倍，可支持并发多达 4 台虚拟机同时迁移。

功能价值：

1. 在设备维护过程中，通过热迁移手动将应用迁移至另一台服务器，维护结束后再迁回来，中间应用不停机，减少计划内宕机时间。

2. 可结合资源动态调度策略，例如在夜晚虚拟机负荷减少时，通过预先配置自动将虚拟机迁移集中至部分服务器，减少服务器的运行数量，从而降低设备运营能耗上的支出。

2.3.4 aSV 的特色技术

2.3.4.1 虚拟机的 HA

HA 全称是 High Availability(高可用性)。在 aSV 环境中，如果出现部署了 HA 的虚拟机所在主机的物理口网线被拔出、或存储不能访问等出现的物理故障时，会将此虚拟机切换到其他的主机上运行，保障虚拟机上的业务正常使用。

aSV 存在后台进程，通过轮询的机制，每隔 5s 检测一次虚拟机状态是否异常，发现异常时，切换 HA 虚拟机到其他主机运行。

下面任意一种情况发生，都会触发 HA 虚拟机切换主机，

- 1、连续三次检测到，虚拟机所连接的物理网卡被拔出（不包括网卡被禁用情况）
- 2、连续两次检测到，虚拟机当前主机无法访问虚拟机的存储

通过 aSV 的 HA 技术，对业务系统提供了高可用性，极大缩短了由于各种主机物理或者链路故障引起的业务中断时间。

2.3.4.2 动态资源调度

在虚拟化环境中，如果生产环境的应用整合到硬件资源相对匮乏的物理主机上，虚拟机的资源需求往往会成为瓶颈，全部资源需求很有可能超过主机的可用资源，这样业务系统的性能也无法保障。

aSV 虚拟化管理平台提供的动态资源调度技术，通过引入一个自动化机制，持续地动态平衡资源能力，将虚拟机迁移到有更多可用资源的主机上，确保每个虚拟机在任何节点都能及时地调用相应的资源。即便大量运行对 CPU 和内存占用较高的虚拟机（比如数据库虚拟机），只要开启了动态资源调度功能，就可实现全自动化的资源分配和负载平衡功能，也可以显著地降低数据中心的成本与运营费用。

aSV 的动态资源调度功能其实现原理：通过跨越集群之间的心跳机制，定时监测集群内主机的 CPU 和内存等计算资源的利用率，并根据用户自定义的规则来判断是否

需要为该主机在集群内寻找有更多可用资源的主机，以将该主机上的虚拟机通过虚拟机迁移技术迁移到另外一台具有更多合适资源的服务器上，或者将该服务器上其它的虚拟机迁移出去，从而保证某个关键虚拟机的资源需求。

2.3.4.3 多 USB 映射

当物理服务器部署虚拟化之后。其中类似金蝶等需要通过 usb key 进行应用加密的服务器，转化到虚拟化后，需要将插在虚拟化平台上的硬件 key，映射给虚拟机，而且需要满足虚拟机热迁移、跨主机映射的需求。

业界给出的方案有三种：

一、采用主机映射：直接采用主机映射的方式来完成，缺点是不支持网络映射，无法支持热迁移、网络映射的需求。

二、采用 Usb Anywhere：通过使用中间设备，将中间设备 IP 化，然后在虚拟机上安装驱动并配置对端设备的方式进行的。缺点是需要 guest 虚拟机内部进行修改安装特定软件，与第三方应用进行配合才能完成。

三、采用底层硬件虚拟化加网络代理：支持热迁移、网络映射、无需修改 guest 机内部。最终实现，物理设备迁移到虚拟化平台后，可以直接无缝的操作读取原 usb 硬件设备。同时解决上述两种方案中的缺陷，破除了在虚拟化推广中外设映射造成的阻碍。

热迁移功能的实现机制：由于整体方案是基于网络代理处理，所以在迁移到对端设备，进行虚拟机切换时，发送消息，触发 usb 服务程序修改连接的目的端 ip，然后发起网络重连。隧道一旦重连成功，usb 设备通信随即恢复，对于 guest 上层来说，是无感知的。

aSV 采用上述的第三种方案，融入了对多 USB 的支持，带来的优势有：

- 1、usb 设备动态插入提示
- 2、guest 虚拟机无需安装插件；
- 3、能支持热迁移，跨主机映射，适应 VMP 集群环境；
- 4、虚拟机迁移完成可以自动挂载上原 usb 设备；
- 5、可以简化集成为类似 usb hub 的小设备，与 VMP 配套，搭建 usb 映射环境；
- 6、虚拟机故障重启、目标端 usb 设备网络中断等异常情况恢复后自动重映射。

2.4 aNet 网络设备虚拟化

2.4.1 网络设备虚拟化概述

网络虚拟化也是构建 aBos 一体机中非常重要的一部分，在私有云、公有云发展迅

速的今天，我们分支机构 IT 基础架构如果继续采用传统 IT 架构中硬件方式定义网络的话，就会存在诸多问题：

- 1、如何保障分支网络设备单点故障问题，双机部署必然带来高成本。
- 2、虚拟化后的分支数据中心涉及多出口，如何识别流量，只能访问到公有云或者总部数据中心。
- 3、对如何满足分支机构快速部署网络设备，灵活扩展分支业务，提出了更高的要求。
- 4、分支业务需要安全防护，部分数据如何保证安全清洗后传输到数据中心。
- 6、高昂的专线线路，如何实现 VPN 替换，满足广域网链路优化。
- 7、集团业务庞大，多分支机构的集中管理如何智能化，能否实现分支“ZERO IT”。

基于上述问题，深信服采用了业界成熟的 NFV 的解决方案，我们称之为 aNet，通过 NFV 实现网络中的所需各类网络功能资源（包括基础的路由交换、安全防护、上网行为管理、IPSEC VPN、广域网优化等）按需分配和灵活调度，从而实现 aBos 超融合架构中的网络虚拟化。除此之外，可以按需交付功能模块即可满足业务需求。

2.4.2 aNET 网络虚拟化技术原理

2.4.2.1 NFV

以开放取代封闭，以通用替代专有——将原本传统的专业网元设备上的网络功能提取出来虚拟化，运行在通用的硬件平台上，业界称这种变化为 NFV。NFV（Network Functions Virtualisation 网络功能虚拟化）的目标是希望通过广泛采用的硬件承载各种各样的网络软件功能，实现软件的灵活加载，在数据中心、网络节点和客户端等各个位置灵活的配置，加快网络部署和调整的速度，降低业务部署的复杂度及总体投资成本，提高网络设备的统一化、通用化、适配性。

NFV 与 SDN 有很强的互补性，NFV 增加了功能部署的灵活性，SDN 可进一步推动 NFV 功能部署的灵活性和方便性。

通过 NFV 技术，将网络功能资源进行虚拟化，使得网络资源升级为虚拟化、可流动的流态资源，Overlay 模型使流态网络资源的流动范围跳出了物理网络的束缚，可以在全网范围内按需流动，呈现出网络资源的统一池化状态，最终实现了超融合架构中网络资源的灵活定义、按需分配、按需调整。

aNet 底层的实现-高性能平台

aNet 的实现主要包含两个层面：数据平面和控制平面。

传统数据平面：

在典型的虚拟化网络场景下，数据包将由网络接口卡接收，然后进行分类并生成规定的动作，并对数据包付诸实施。在传统的 Linux 模式下，系统接收数据包和将数据包发送出系统的过程占了包处理中很大一部分时间，换句话说，即使用户空间应用程序什么都不做，而只是将数据包从接收端口传送到发送端口，那么仍然会花费大量的处理时间。

当网卡从网络接收到一个数据帧后，会使用直接内存访问（DMA）将数据帧传送到针对这一目的而预先分配的内核缓冲区内，更新适当的接收描述符环，然后发出中断通知数据帧的到达。操作系统对中断进行处理，更新环，然后将数据帧交给网络堆栈。网络堆栈对数据进行处理，如果数据帧的目的地是本地套接字，那么就将数据复制到该套接字，而拥有该套接字的用户空间应用程序就接收到了这些数据。

进行传输时，用户应用程序通过系统调用将数据写入到一个套接字，使 Linux 内核将数据从用户缓冲区复制到内核缓冲区中。然后网络堆栈对数据进行处理，并根据需要对其进行封装，然后再调用网卡驱动程序。网卡驱动程序会更新适当的传输描述符环，并通知网卡有一个等待处理的传输任务。

网卡将数据帧从内核缓冲区转移到自己内置的先进先出（FIFO）缓冲区，然后将数据帧传输到网络。接着网卡会发出一个中断，通知数据帧已经成功传输，从而使内核释放与该数据帧相关的缓冲区。

传统模式下 CPU 损耗主要发生在如下几个地方：

中断处理：这包括在接收到中断时暂停正在执行的任务，对中断进行处理，并调度 soft IRQ 处理程序来执行中断调用的实际工作。随着网络流量负荷的增加，系统将会花费越来越多的时间来处理中断，当流量速度达到 10G 以太网卡的线路速度时就会严重影响性能。而对于有着多个 10G 以太网卡的情况，那么系统可以被中断淹没，对所有的服务产生负面影响。

上下文切换：上下文切换指的是将来自当前执行线程的寄存器和状态信息加以保存，之后再将来自被抢占线程的寄存器和状态信息加以恢复，使该线程能够从原先中断的地方重新开始执行。调度和中断都会引发上下文切换。

系统调用：系统调用会造成用户模式切换到内核模式，然后再切换回用户模式。这会造成管道冲刷并污染高速缓存。

数据复制：数据帧会从内核缓冲区复制到用户套接字，并从用户套接字复制到内核缓冲区。执行这一操作的时间取决于复制的数据量。

调度：调度程序使每个线程都能运行很短的一段时间，造成多任务内核中并发执

行的假象。当发生调度定时器中断或在其他一些检查时间点上，Linux 调度程序就会运行，以检查当前线程是否时间已到。当调度程序决定应该运行另一个线程时，就会发生上下文切换。

2.4.2.2 aNet 底层的实现-构建高性能平台

● 数据平面

Linux 等通用操作系统，必须公平地对待网络应用程序和非网络应用程序，导致设计上达不到高 IO 吞吐，深信服的 aNet 数据面设计上，借鉴了 netmap 和 dpdk 的方案，针对数据 IO 密集型网络应用程序设计。

1. 支持专有网卡和通用网卡

对于 Intel 和 Broadcom 的 e1000e, igb, ixgbe, bnx2, tg3, bnx2x 等可编程网卡，支持高性能方案，对 e1000 等网卡，支持通用方案。保证硬件兼容性。

2. 跨内核跨进程的全局内存池

深信服设计并实现了零拷贝的数据面环境，一个跨内核跨进程的全局内存引用机制，真正做到网卡收包一次拷贝，所有进程共享引用的方式，数据可以从网卡传送到内核、应用层、虚拟机而无需再次拷贝。内存池自动增长，自动回收。

3. 避免中断处理和上下文切换

单数据线程亲和锁定到硬件线程，避免内核和用户空间之间的上下文切换、线程切换和中断处理，同时每个线程有直接的高速缓冲，避免了缓冲区争用。

在理想情况下，当数据包到达系统时，所有处理该数据包所需的信息最好都已经在内核的本地高速缓存中。我们可以设想一下，如果当数据包到达时，查找表项目、数据流上下文、以及连接控制块都已经在高速缓存中的话，那么就可以直接对数据包进行处理，而无需“挂起”并等待外部顺序内存访问完成。

4. 应用层数据面更稳定

内核态的小 BUG，可能导致系统宕机，而应用层进程，最糟糕的情况是进程死掉，我们设计了检测监控机制，在最极端的情况，即使进程意外死亡，也能秒级别做到虚拟机无感知的网络恢复。

数据平面负责报文的转发，是整个系统的核心，数据平面由多个数据转发线程和一个控制线程组成，控制线程负责接收控制进程配置的消息，数据线程是实现报文的处理。

在数据线程中实现快速路径与慢速路径分离的报文处理方式，报文的转发是基于 session 的，一条流匹配到一个 session，该条流的第一个报文负责查找各种表项，创建 session，并将查找表项的结果记录到 session 中，该条流的后续的报文只需查找

session，并根据 session 中记录的信息对报文进行处理和转发的。

系统中所有的报文都是由数据线程接收的，需要做转发的报文，不需要送到 linux 协议栈，直接在数据线程中处理后从网卡发出，对于到设备本身的报文(如 ssh, telnet, ospf, bgp, dhcp 等等)，数据线程无法直接处理，通过 TUN 接口将报文重新送到 linux 协议栈处理，从 linux 协议栈的发出的报文需经过数据线程中转后才可从折本发出。

aNet 数据层面，在六核 2.0 GHz 英特尔至强处理器 L5638 上使用最长前缀匹配 (LPM) 时，对于使用六个核心中的四个核心、每个核一个线程、四个 10G 以太网端口的情况，64 字节数据包的 IP 第三层转发性能达到了 900 万 pps。这比原始 Linux 的性能差不多提高了九倍（原始 Linux 在双处理器六核 2.4GHz 模式下的性能为 100 万 pps）。

数据面为底层处理和数据包 IO 提供了与硬件打交道的功能，而应用层协议栈在上方提供了一个优化的网络堆栈实现。与 Linux SMP 解决方案相比，降低了对 Linux 内核的依赖性，从而具有更好的扩展性和稳定性。

● 控制平面

有了数据面和协议栈做支持，控制面就可以实现丰富的应用功能。控制面实现了本地配套服务，一些基础功能例如 DHCP 服务，RSTP 服务，DNS 代理功能。这些内置服务可以直接提供给虚拟机，用户无需安装第三方类似软件。

2.4.3 aNet 功能特性

2.4.3.1 vAC 上网行为管理

在分支网络建设中，上网行为管理是必不可少的网络设备。不仅能够满足分支员工上网行为的审计，同时进行流量控制保证分支核心业务的正常使用。除此之外，实现应用控制规范化员工的上网行为，对分支私接 WiFi 进行管理，保证内网环境不泄露。

1、全网全终端统一管控、管理无漏洞

- 能够管理有线网络、无线网络，同时管理移动终端、PC/笔记本，管理无漏洞；
- 能够对 600 多种移动应用进行有效地识别和精细管控（如微信传文件、微信聊天、微信朋友圈、微信游戏）；
- 对非法无线热点能够及时发现和精准控制，秒级识别非法热点；
- 能够基于位置、应用、终端、用户四维一体的识别与权限控制；

2、上网行为管控更有效：上网应用识别更有效、管控更精细

- 具备全国最大的应用识别特征库和 URL 库，应用识别种类更多，并且每 2 周更新和淘汰一次，时效性更强、准确度更高。可以对迅雷、PPStream、风行等应用全流量识别；
- 应用控制更精细，可区分应用动作（如社交网站的浏览、发帖回帖、上传）、区分方向（如网盘的上传、下载）进行管控；
- 应用行为标签化管理，做到对指定类别的应用进行批量管理，策略部署更简单。

3、上网行为管控更有效：流量管理更精准、控制保障两不误

- P2P 智能流控技术：精确控制 P2P 上下行流量，相比市场上其他 P2P 控制技术，AC 可提高 30%以上的带宽利用率；
- 动态流控技术：根据具体的带宽空闲程度和业务需要，受限的通道可以突破上限，提高带宽利用率和用户体验；
- 流量管理策略更灵活，呈现更直观（能够针对 URL 类型、文件类型做流控，通道流量实时可视化以及细粒度的可视化报表）。

4、上网行为管控更有效：外发数据识别更精确，真正防泄密

- 具备业界最好的内容识别与管控技术，可以对多种外发途径的数据进行有效管控（如网盘上传附件控制、论坛上传附件控制、邮件外发附件审计与控制、IM 外发文件控制等）；
- 可以对 SSL 加密内容精准识别与控制（如邮件客户端收发加密邮件、加密论坛审计等）。

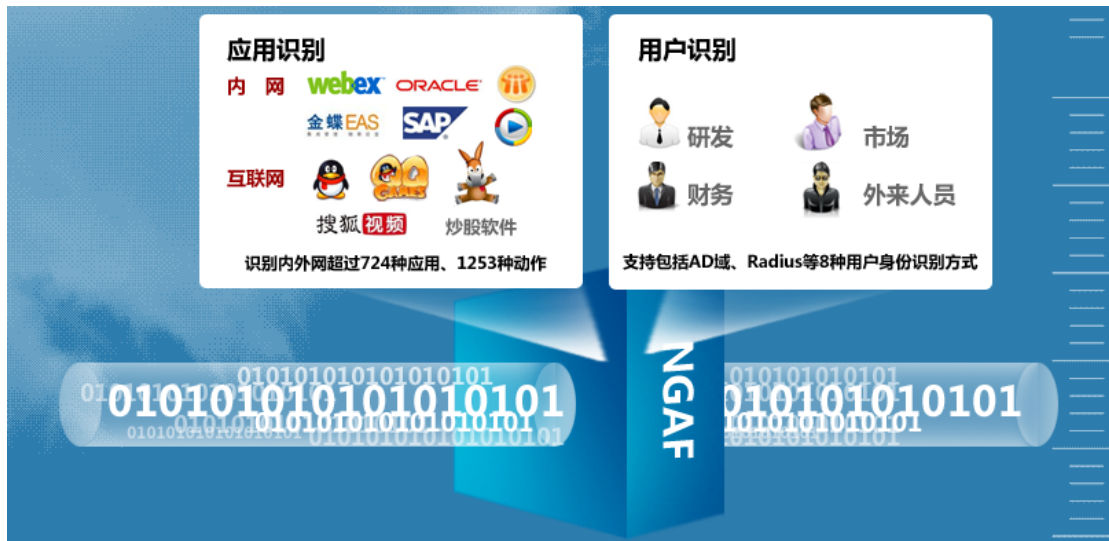
2.4.3.2 vAF 下一代防火墙

在构建分支 IT 基础架构中，安全是其中必不可少一环，内网业务、终端的安全防护是分支 IT 建设的重中之重。深信服下一代防火墙是面向应用层设计，能够精确识别用户、应用和内容，具备完整安全防护能力，能够全面替代传统防火墙，并具有强劲应用层处理能力的全新网络安全设备。

1、精细的应用安全访问控制

区别于传统的网络层防火墙，NGAF 具备 L2-L7 层的协议的理解能力。不仅能够实现网络层访问控制的功能，且能够对应用进行识别、控制、防护，解决了传统防火墙应用层控制和防护能力不足的问题。

（1）可视化的应用识别



图：智能的用户与应用识别能力

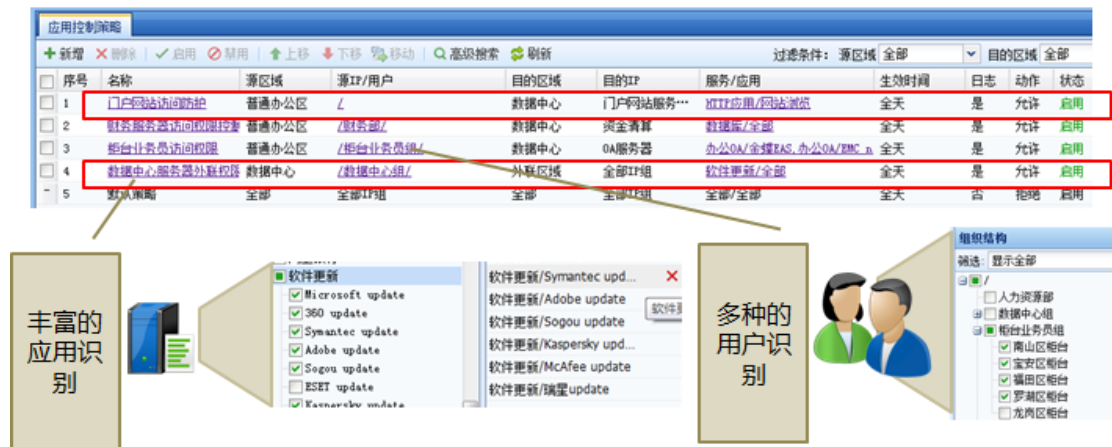
NGAF 具有卓越的应用可视化功能，通过多种应用识别技术形成国内最大的应用特征识别库，可精确识别内外网的采用端口跳跃、端口逃逸、多端口、随机端口的各类应用，为下一代防火墙实现用户与应用的精细化访问控制提供技术基础。

(2) 智能用户身份识别

NGAF 用户识别功能可以与 8 种认证系统（AD、LDAP、Radius 等）、应用系统（POP3、SMTP 等）无缝对接，通过单点登录的方式自动识别出网络当中 IP 地址对应的用户信息，并建立组织的用户分组结构。

(3) 面向用户与应用的控制策略

区别于传统防火墙的五元组访问控制策略，下一代防火墙可通过应用可视化功能与用户识别技术结合制定的 L3-L7 一体化应用控制策略，可以为用户提供更加精细和直观化控制界面，在一个界面下完成多套设备的运维工作，提升工作效率。



图：基于用户和应用的访问控制策略

2、全面的应用安全防护能力

能够防护 web 攻击，与 web 应用防火墙关注 web 应用程序安全的设计理念不同，深信服下一代防火墙 NGAF 关注 web 系统在对外发布的过程中各个层面的安全问题，为对外发布系统打造坚实的防御体系。

(1) 强化 web 攻击防护

深信服下一代防火墙 NGAF 采用攻击特征+主动防御相结合的双重防护模式，可有效保护 web 业务安全。攻击特征的防护模式有效结合了 web 攻击的静态规则及基于黑客攻击过程的动态防御机制，提供 OWASP 定义的十大安全威胁的攻击防护功能，有效防止常见的 web 安全威胁。如 SQL 注入、XSS 跨站脚本、CSRF 跨站请求伪造，敏感信息泄露等，主动模式可提供参数类型学习的主动防御和自定义参数防护等个性化配置，保护 web 系统免受网站篡改、网页挂马、隐私侵犯、身份窃取、经济损失、名誉损失等问题。

(2) 基于应用的深度入侵防御

NGAF 基于应用的深度入侵防御采用六大威胁检测机制：攻击特征检测、特殊攻击检测、威胁关联分析、异常流量检测、协议异常检测、深度内容分析能够有效的防止各类已知未知攻击，实时阻断黑客攻击。如，缓冲区溢出攻击、利用漏洞的攻击、协议异常、蠕虫、木马、后门、DoS/DDoS 攻击探测、扫描、间谍软件、以及各类 IPS 逃逸攻击等。

(3) 高效精确的病毒检测能力

NGAF 提供先进的病毒防护功能，可从源头对 HTTP、FTP、SMTP、POP3 等协议流

量中进行病毒查杀，也可查杀压缩包（zip, rar, gzip 等）中的病毒。同时采用高效的流式扫描技术，可大幅提升病毒检测效率避免防病毒成为网络安全的瓶颈。

(4) DOS/DDOS 攻击防护

NGAF 可防护基于数据包的 DOS 攻击、IP 协议报文的 DOS 攻击、TCP 协议报文的 DOS 攻击、基于 HTTP 协议的 DOS 攻击等，实现对网络层、应用层的各类资源耗尽的拒绝服务攻击的防护，实现 L2-L7 层的异常流量清洗。

3、独特的双向内容检测技术

区别于传统 DPI 技术的入侵防御系统，深信服 NGAF 具备深入应用内容的威胁分析能力，具备双向的内容检测能力为用户提供完整的应用层安全防护功能。



图：双向内容检测

(1) 可定义的敏感信息防泄漏

NGAF 提供可定义的敏感信息防泄漏功能，根据储存的数据内容可根据其特征清晰定义，通过短信、邮件报警及连接请求阻断的方式防止大量的敏感信息被窃取。深信服敏感信息防泄漏解决方案可以自定义多种敏感信息内容进行有效识别、报警并阻断，

防止大量敏感信息被非法泄露。（如：用户信息/邮箱账户信息/MD5 加密密码/银行卡号/身份证号码/社保账号/信用卡号/手机号码……）

（2）僵尸网络检测隔离

NGAF 独有的僵尸网络检测隔离功能，应用 NGAF 对外发流量的检测能够判断服务器或终端由于中了病毒木马向外发起的恶意流量。该功能融合了僵尸网络识别库，利用业界领先的僵尸网络识别检测技术对黑客的攻击行为进行有效识别，针对以反弹式木马为代表的恶意软件进行深度防护，可识别僵尸网络流量达 15 万条，并由深信服攻防团队实时更新。同时，深信服 NGAF 正在完善安全云平台，部署在全球各地的深信服下一代防火墙设备可以自动或手动上传可疑的应用流量到安全云平台，平台会自动分析，形成新的恶意软件识别策略下发到全球所有设备的规则库上。

（3）应用协议内容隐藏

NGAF 可针对主要的服务器（WEB 服务器、FTP 服务器、邮件服务器等）反馈信息进行了有效的隐藏。防止黑客利用服务器返回信息进行有针对性的攻击。如：HTTP 出错页面隐藏、响应报头隐藏、FTP 信息隐藏等

（4）用户登录权限防护

NGAF 可以针对特定的服务或者 web 页面提供登陆保护，通过发送短信验证码的方式针对敏感信息和应用提供强认证保护。用户访问到该页面或应用的时候需要经过短信的二次认证，增强敏感页面或应用的安全系数；该功能能够带来的价值：

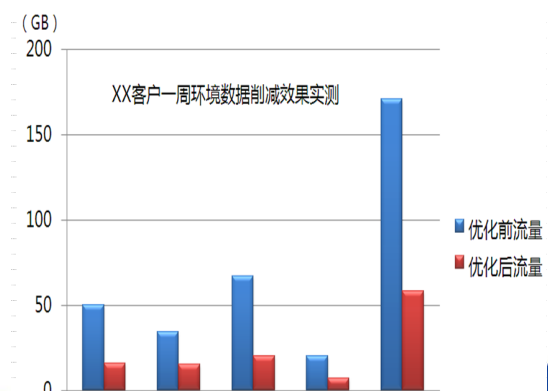
- 1、对重要的页面（如管理员页面）进行防护，防止通过社会工程、暴力破解拿到正常管理员的账号密码；
- 2、可实现敏感页面的双因子强认证提高安全性，防止敏感页面开放于公网或办公网；

2.4.3.3 广域网加速

1、访问提速

提高应用系统访问速度，保障 VPN 跨网访问质量；

- (1) 针对跨运营商或线路质量不好的网络高丢包高延时的情况，采用 HTP 技术进行优



化，大幅降低丢包率，并提升高延时下访问速度。

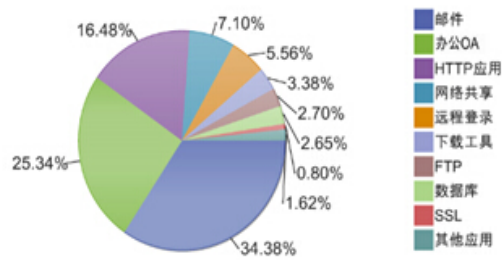
(2) 替换专线后的加速 VPN 线路上承载 OA、办公及视频会议等系统，通过应用加速技术实现对 OA 等核心系统的访问提速，针对视频会议在跨运营商线路上的优化，消除马赛克保障视频会议高效稳定流畅。

(3) 针对办公文件、财务数据等应用数据通过缓存压缩进行流量削减，降低带宽的压力，提高交付效率，提升带宽价值。

2、流量可视化管理

(1) 流量可视化：对当前线路的流量组成进行分析，再进行有效的管理。

(2) VPN 带宽保障、上网应用流控：对 VPN 通道内的应用系设置带宽保障流量整形，并对上网应用基于用户、应用、时间三元



素进行使用权限控制及流量分配，防止因 P2P、视频等无关工作应用挤占带宽，保障 VPN、上网可用性。

2.4.3.4 aSW 虚拟分布式交换机

虚拟分布式交换机是管理多台主机上的虚拟交换机的虚拟网络管理方式，包括对主机的物理端口和虚拟机虚拟端口的管理。

aSV 虚拟化平台提供的虚拟分布式交换机就是把分布在集群中多台主机的单一交换机逻辑上组成一个大的集中式交换机，减少每台虚拟交换机需要单独分别配置过程，同时为集群级别的网络连接提供一个集中控制点，使虚拟环境中的网络配置不再以主机为单位，简化虚拟机网络连接的部署、管理和监控，适合于大规模的网络部署。

虚拟分布式交换机可以保证虚拟机在主机之间迁移时网络配置的一致性，同时提供丰富的网络配置管理功能，端口动态绑定，静态绑定，IP 接入控制、虚拟机网络 Qos，实现网络资源统一管理，实时化网络监控。

2.4.4 深信服 aNet 的特色技术

2.4.4.1 所画即所得业务逻辑拓扑

所画即所得的业务逻辑呈现，是深信服超融合架构中，非常具有特色的技术功能，由于 aSV、aSAN 已将计算和存储的资源池拉通，结合 aNet 提供的网络资源池，从而为业务逻辑拓扑的搭建提供了各种元素，通过管理平面便可从资源池中调取相应的资源，即可呈现出不同的拓扑架构。当从管理页面，构建业务逻辑拓扑时候，整个超融合架构底层，会执行大量的动作和指令，并且根据业务逻辑拓扑进行底层真实的环境模拟，

从而屏蔽了底层的复杂性，方便 IT 管理人员可以更快速，简单，直观的方式构建数据中心各个业务所需的逻辑拓扑。

2.5 aSAN 存储虚拟化

2.5.1 存储虚拟化概述

aSAN 是深信服在充分掌握了用户对虚拟化环境存储方面的需求基础上，推出以 aSAN 分布式存储软件为核心的解决方案，aSAN 是基于分布式文件系统 Glusterfs 开发的面对存储虚拟化的一款产品，并作为 aBos 超融合架构中的重要组成部分，为云计算环境而设计，融合了分布式缓存、SSD 读写缓存加速、多副本机制保障、故障自动重构机制等诸多存储技术，能够满足关键业务的存储需求，保证客户业务高效稳定可靠的运行。

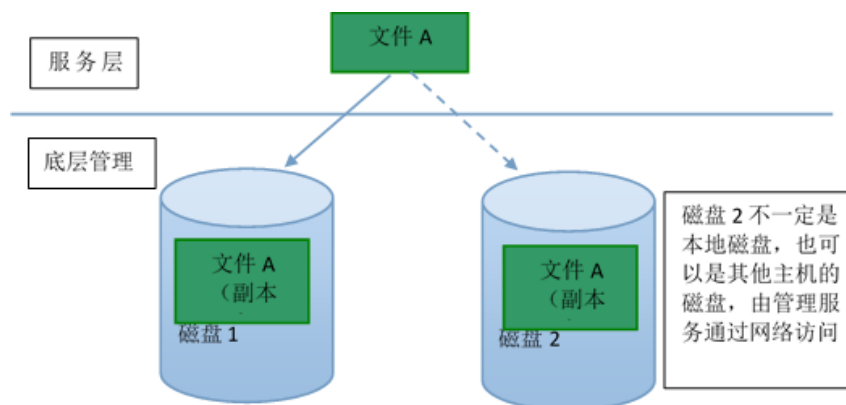
2.5.2 aSAN 技术原理

aSAN 基于底层 Hypervisor 之上，通过主机管理、磁盘管理、缓存技术、存储网络、冗余副本等技术，管理集群内所有硬盘，“池化”集群所有硬盘存储的空间，通过向 VMP 提供访问接口，使得虚拟机可以进行业务数据的保存、管理和读写等整个存储过程中的操作。

2.5.2.1 文件副本

由于下一节磁盘管理的策略与副本设置有直接管理，因此在讲解磁盘管理前，我们要先介绍文件副本技术。

所谓文件副本，即将文件数据保存多份的一种冗余技术。aSAN 副本颗粒度是文件级别。例如两个副本，即把文件 A 同时保存到磁盘 1 和磁盘 2 上。并且保证在无故障情况下，两个副本始终保持一致。



技术特点：

存储池可用空间=集群全部机械磁盘空间/副本数（同构情况），因此副本是会降低实际可用容量的。

底层管理的副本对上层服务是透明的，上层无法感知副本的存在。磁盘管理、副本分布由底层服务负责，副本颗粒度是文件级。

在没有故障等异常情况下，文件副本数据是始终一致的，不存在所谓主副本和备副本之分。

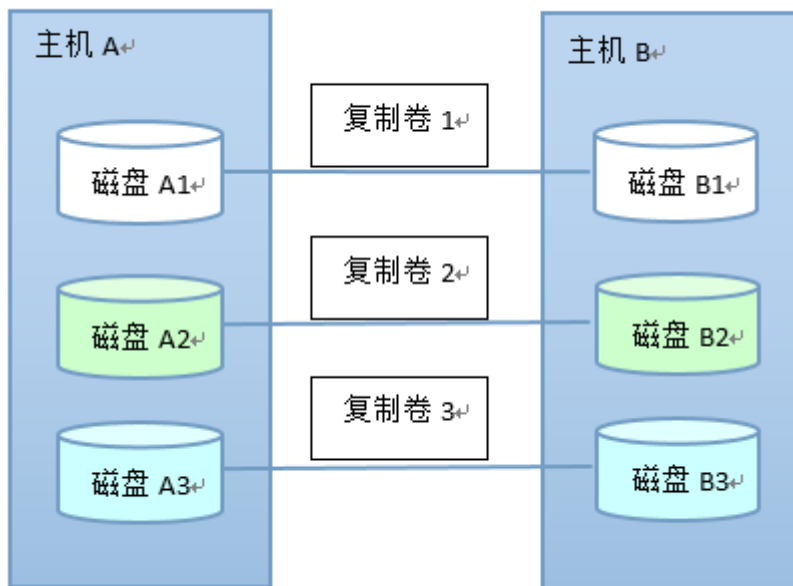
如果对文件 A 进行修改，如写入一段数据，这段数据会被同时写到两个副本文件。如果是从文件 A 读取一段数据，则只会从其中一个副本读取。

2.5.2.2 磁盘管理

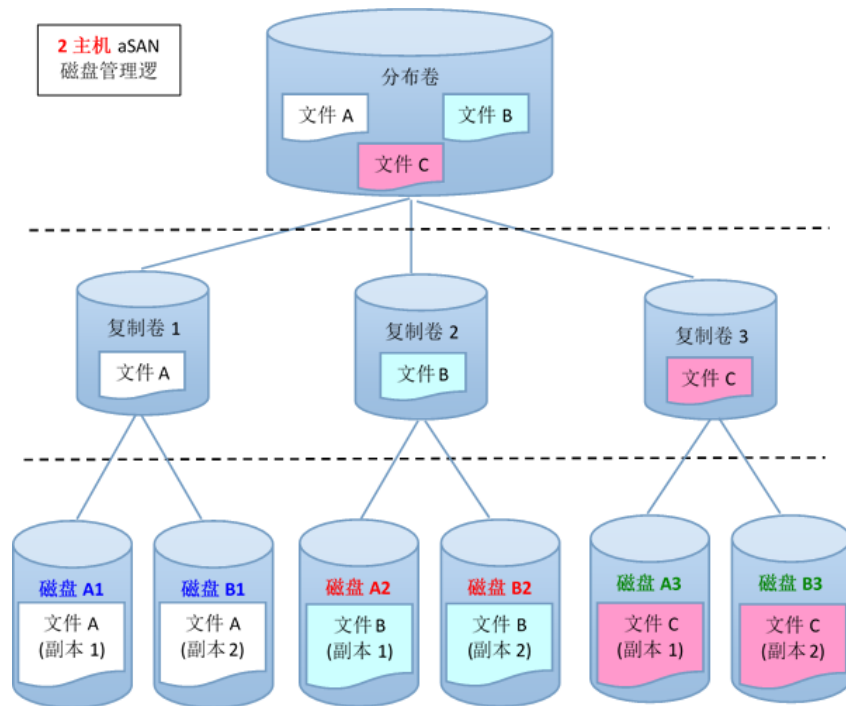
aSAN 的环境中，可以实现单台、两台主机节点来构建 aSAN，实现两副本的存储资源池，保证数据安全性。

在多主机集群下，采用**两个副本**副本组建 aSAN 的磁盘管理。若需要在保证主机故障而不影响数据完整性的目标，复制卷的磁盘组的每个磁盘都必须是在不同主机上。即需要做到跨主机副本。跨主机副本的关键在于复制卷磁盘分组算法。

以下面场景为列（两台主机，每台主机各三块磁盘组建两个副本）：



当构建两副本，并且两台主机磁盘数相同时。主机间的磁盘会一一对应组成复制卷。逻辑视图如下：

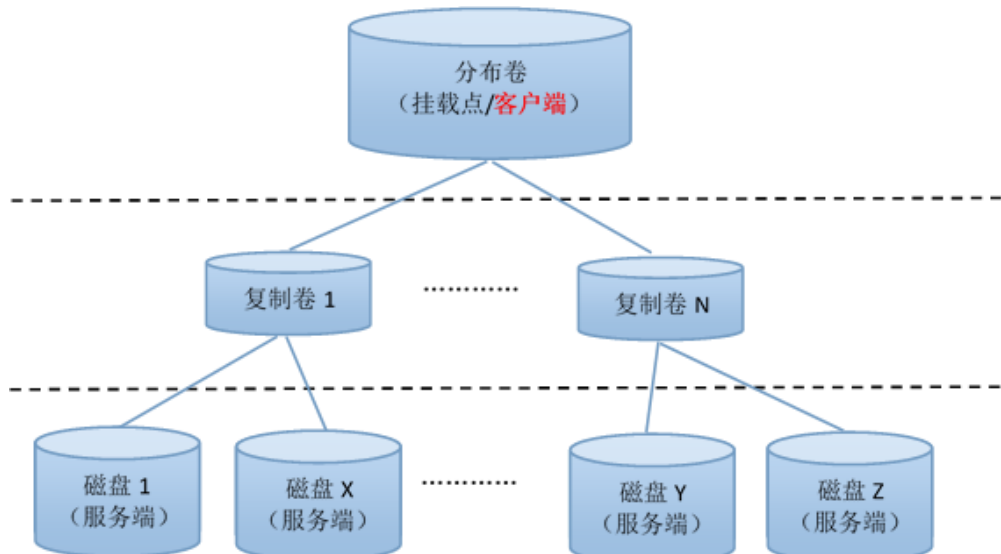


从逻辑视图上，可以看出来和前面提到的单主机逻辑视图并没有本质上的区别，只是最底层的磁盘分组时，保证了复制卷内下面的磁盘不在同一主机内，从而达到了文件跨主机副本的目标。

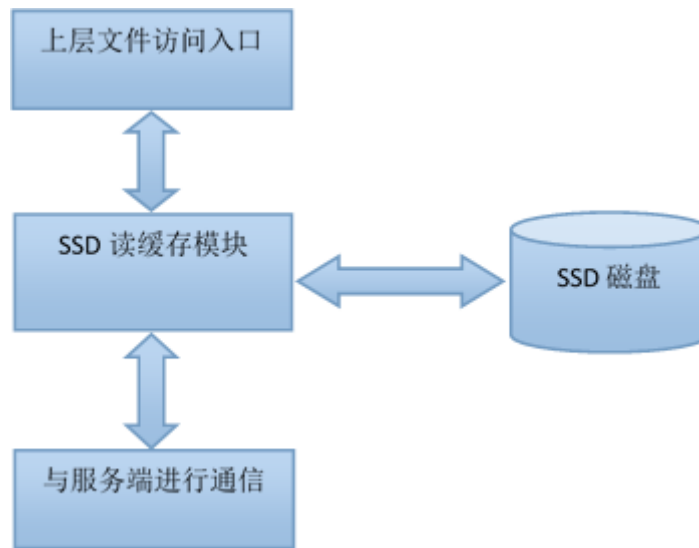
2.5.2.3 SSD 读缓存加速原理

在 aSAN 里面，会默认把系统内的 SSD 磁盘作为缓存盘使用，下面介绍 aSAN SSD 读缓存原理。

首先需要区分 aSAN 客户端和服务端概念。在 aSAN 里面，负责处理底层磁盘 IO 称为服务端；负责向上层提供存储接口（如访问的挂载点）称为客户端。aSAN SSD 读缓存工作在客户端，（注意：aSAN 的 SSD 写缓存则工作在服务端）。逻辑视图如下：



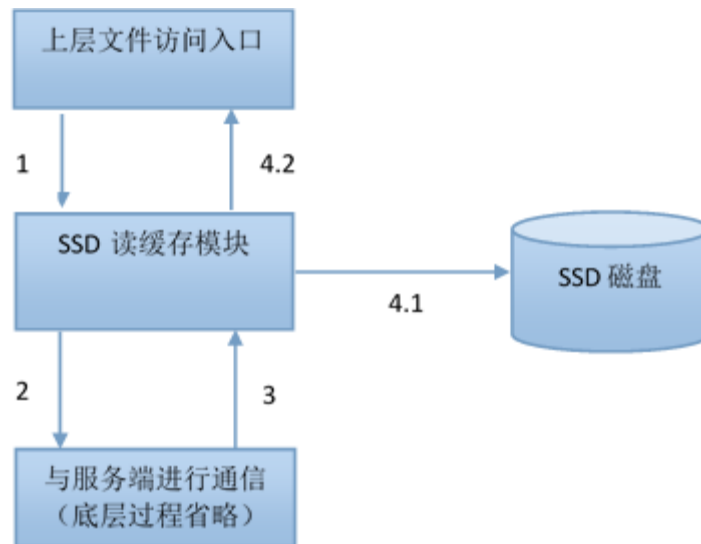
下面抛开底层的分布卷、复制卷、磁盘分组等概念，仅在客户端上理解 SSD 读缓存的原理。



SSD 读缓存的缓存颗粒度是按文件数据块缓存，不是文件整体。例如，A、B、C 三个文件，可以分别各缓存读过的一部分数据，没读过的部分不缓存。

简单地看，SSD 读缓存模块工作在文件访问入口和服务端通信层之间。所有对文件的 IO 动作都会经过 SSD 读缓存模块进行处理。下面分别针对首次文件读取、二次文件读取、文件写入 3 个过程说明工作流程。

● 首次文件读取



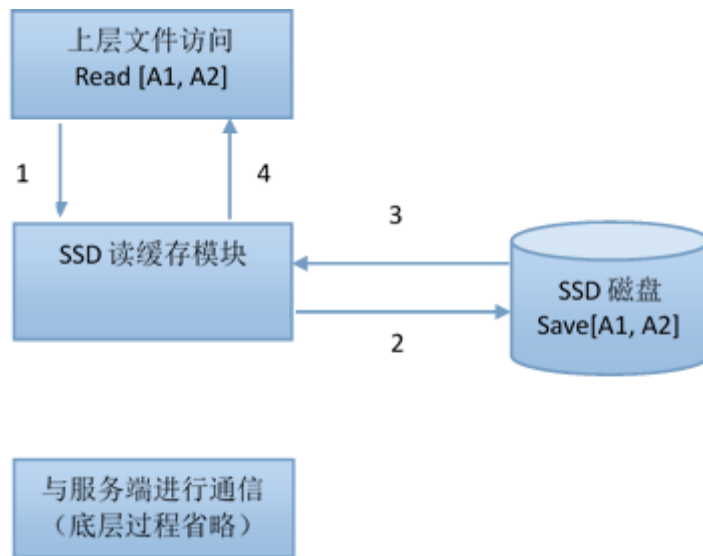
未缓存数据块的首次读操作步骤说明：

1. 从上层下来一个针对 A 文件的区间块 [A1, A2] 的读操作，由于该数据块是首次读取，没命中 SSD 读缓存。该读操作会直接传递到下去，进入流程 2。
2. [A1, A2] 的读操作继续传递到服务端，进行具体的读操作，完成后返回，进

入流程 3

3. 数据块[A1, A2]在流程 3 里面返回到 SSD 读缓存模块，进入流程 4
4. SSD 读缓存模块会把数据块[A1, A2]复制一份保存到 SSD 磁盘并建立相关索引，对应 4.1。原数据块[A1, A2]继续往上返回到上层响应读操作，对应 4.2。注意 4.1、4.2 是并发进行，因此这个缓存动作不会对原操作造成延时。
5. 至此，数据块[A1, A2]就被保存到 SSD 磁盘内，以备下次读取直接从 SSD 磁盘读取。

● 二次文件读取



针对已缓存数据块的二次读取步骤说明：

假设数据块[A1, A2]已经缓存到 SSD 磁盘内，

1. 从上层下来一个同样是对 A 文件的区间块 [A1, A2] 的读操作。
2. 由于该数据块[A1, A2]已经有缓存，在 SSD 读缓存模块里面命中索引，从而直接向 SSD 磁盘发起读出缓存数据块[A1, A2]的操作。
3. 缓存数据块[A1, A2]从 SSD 磁盘返回到 SSD 读缓存模块，进入流程 4
4. SSD 读缓存模块把缓存数据块[A1, A2]返回给上层。

至此，对缓存数据块[A1, A2]的重复读取直接在客户端返回，避免了服务端通信的流程，从而减少了延时和减轻了底层磁盘的 IO 压力。

● 文件写入

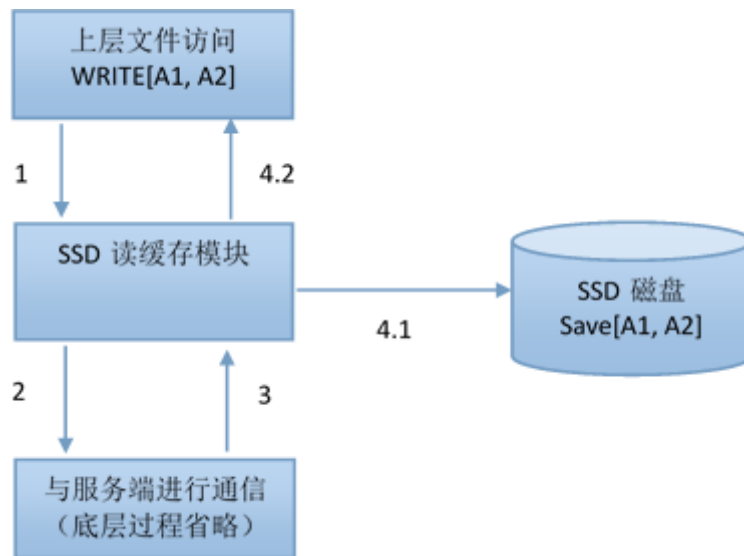
虽然当前 aSAN 实现的读缓存，但对于读缓存模块对于文件写入操作，也需要做相应的处理，以保证缓存的内容始终和底层磁盘一致，并且是最新的，但这个针对文件写入的处理并不是写缓存。

aSAN 读缓存模块对写操作进行处理实质是基于最近访问原则，即最近写入的数据

在不久的将来被读出的概率会比较高，例如文件共享服务器，某人传到文件服务器的文件，很快会其他人读出来下载。

aSAN 读缓存对写操作的处理从实现上分为首次写预缓存、二次写更新缓存。

■ 文件块首次写预缓存



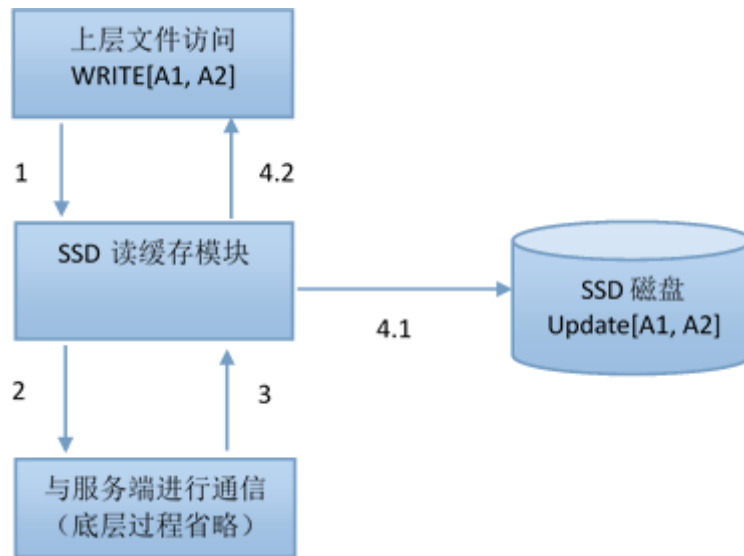
流程说明：假设数据块[A1, A2]是首次写入。

1. 写操作写来经过 SSD 读缓存模块。由于是写操作，SSD 读缓存会直接 PASS 到下层
2. 写操作一直传递到服务端，写入到底层磁盘，操作完成后会返回结果，进入流程 3
3. 返回结果经过 SSD 读缓存模块，如果返回结果是成功的，表示底层数据已经成功写入，则进入流程 4。如果返回结果是失败，则不会进入流程 4，而是直接返回结果到上层。
4. SSD 读缓存模块会把数据块[A1, A2]复制一份保存到 SSD 磁盘并建立相关索引，对应 4.1。原返回结果继续往上返回到上层响应读操作，对应 4.2。注意 4.1、4.2 是并发进行，因此这个缓存动作不会对原操作造成延时。

至此，数据块[A1, A2]的写入也会保存到 SSD 磁盘上，以备下次访问。下次访问的流程与二次文件读取流程相同，从而提升了下次访问数据的速度。

■ 文件块二次写更新缓存

SSD 读缓存文件块写更新是指对 SSD 读缓存已缓存的数据块进行更新的动作。

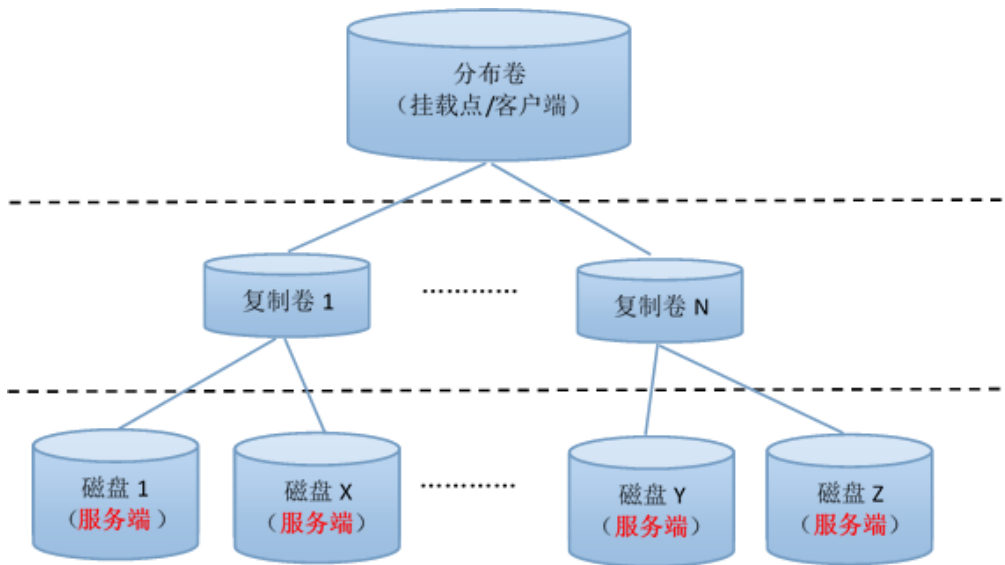


假设数据块[A1, A2]原来已经有缓存了，现在上层再次对 [A1, A2] 来一次写操作（例如更新内容）。

1. 写操作写来经过 SSD 读缓存模块，由于是写操作，SSD 读缓存会直接 PASS 到下层
2. 写操作一直传递到服务端，写入到底层磁盘，操作完成后会返回结果，进入流程 3
3. 返回结果经过 SSD 读缓存模块，如果返回结果是成功的，表示底层数据已经成功写入，可以更新 SSD 读缓存数据，进入流程 4。如果返回结果是失败，则不会进入更新流程。
4. SSD 读缓存模块会把数据块[A1, A2]复制一份更新到 SSD 磁盘并建立相关索引，对应 4.1。原返回结果继续往上返回到上层响应读操作，对应 4.2。注意 4.1、4.2 是并发进行，因此这个缓存动作不会对原操作造成延时。

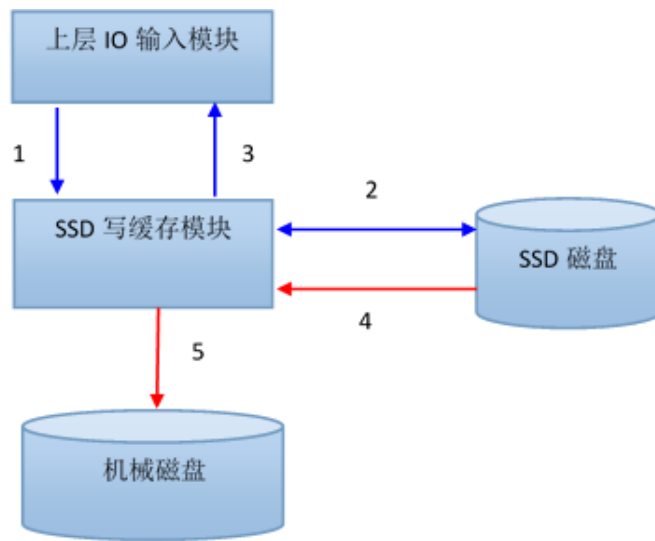
2.5.2.4 SSD 写缓存加速原理

SSD 写缓存工作在服务端。由于写缓存工作在服务端，也就是说在每个副本上都有写缓存，即 SSD 写缓存也是多副本的。即使有 SSD 磁盘突然损坏，也能在副本数范围内保证数据的安全。



● **SSD 写缓存模块结构**

SSD 写缓存原理是在机械硬盘上增加一层 SSD 写缓存层，见下图：



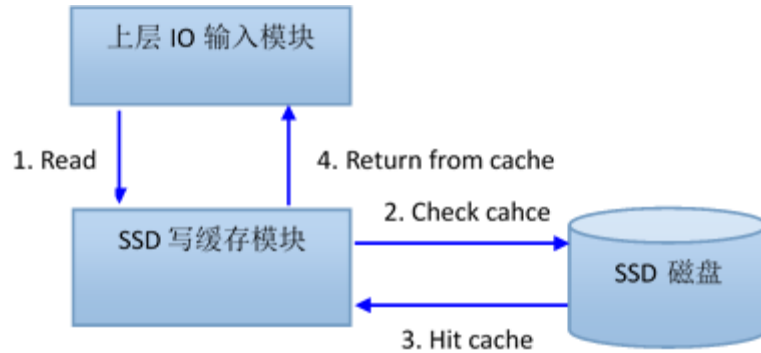
SSD 写缓存数据流分成蓝色和红色两部分。这两部分是同时在运行的，没有先后关系。蓝色部分是虚拟机有数据写入 SSD 缓存，红色部分是从 SSD 缓存读出数据回写到机械磁盘。流程如下：

1. 上层写入数据请求到达 SSD 写缓存模块
2. SSD 写缓存模块把数据写入到 SSD 磁盘，并获得返回值。
3. SSD 写缓存模块在确定数据写入 SSD 磁盘后，即立即返回上层模块写入成功
4. SSD 写缓存模块在缓存数据累计到一定量后，从 SSD 磁盘读出数据
5. SSD 写缓存把从 SSD 磁盘读出的数据回写到机械磁盘。

其中，第 4、5 步是在后台自动进行的，不会干扰第 1、2、3 步的逻辑。

● **SSD 写缓存数据读命中**

从 SSD 磁盘回写到机械磁盘是需要累积一定数据量后才会进行触发的。这时如果来了一个读数据请求，SSD 写缓存模块会先确认该读请求是否在 SSD 写缓存数据内，如果有则从 SSD 缓存内返回；如果没有则透到机械硬盘去读取。

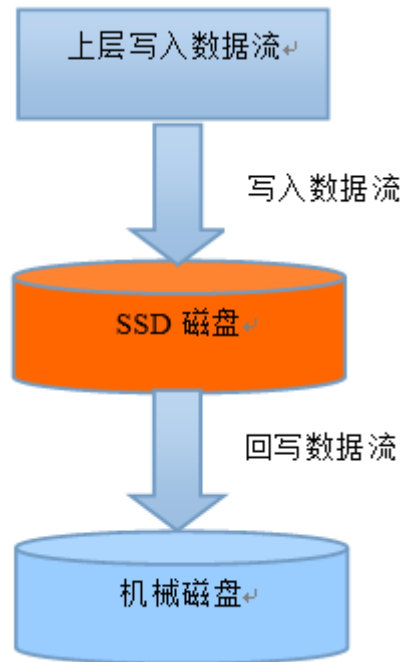


流程说明：

1. 上层下发读请求
2. SSD 写缓存模块先检查数据是否还在缓存内未回写
3. 命中缓存，返回数据（如果不命中缓存，则会返回从底层数据盘读取）
4. 向上层返回数据

● **SSD 写缓存写满后处理**

如果上层持续对 SSD 写缓存进行大量不间断的数据写入，直到 SSD 写缓存空间用完。这时的上次继续写入数据的速度就会下降至约等于写缓存回写机械盘的速度。



当 SSD 磁盘用满时会出现写入数据流速度 \leq 回写数据流速度。在虚拟机层面看，就是写入数据下降到机械盘速度。如果持续出现这种情况，说明 SSD 磁盘容量不足以应对业务 IO 写性能，需要增加 SSD 缓存盘解决。

- 当 SSD 磁盘故障或离线时的处理

如前文所说，SSD 写缓存工作于服务端，有多副本机制。在多主机多副本场景下，如果一个 SSD 磁盘损坏后，其他副本的 SSD 还正常情况下，对数据安全不会造成影响。一旦 SSD 离线超过 10 分钟，缓存数据就视作失效，进入副本修复流程。由于所有数据都是被 SSD 接管的，因此如果是误拔出 SSD 硬盘，需要在 10 分钟内插回来，否则会认为该副本数据全部需要重建。

2.5.3 aSAN 存储数据可靠性保障

副本修复是指当某个磁盘出现离线再上线后，保存在上面的文件副本可能是旧数据，需要按照其他在线的文件副本进行修复的一个行为。典型的情况是主机短暂断网，导致副本不一致。

通过采用副本快速修复技术，即对于短暂离线的副本，只修复少量差异数据，从而避免了整个文件进行对比修复，达到快速修改的目的，同时，aSAN 对业务 IO 和修复 IO 做了优先级控制，从而避免了副本修复 IO 对业务 IO 的影响。

2.5.4 深信服 aSAN 功能特性

2.5.4.1 存储自动精简配置

自动精简配置 (Thin Provisioning) 是一种先进的、智能的、高效的容量分配和

管理技术，它扩展了存储管理功能，可以用小的物理容量为操作系统提供超大容量的虚拟存储空间。并且随着应用的数据量增长，实际存储空间也可以及时扩展，而无须手动扩展。一句话而言，自动精简配置提供的是“运行时空间”，可以显著减少已分配但是未使用的存储空间。

如果采用传统的磁盘分配方法，需要用户对当前和未来业务发展规模进行正确的预判，提前做好空间资源的规划。在实际中，由于对应用系统规模的估计不准确，往往会造成容量分配浪费，比如为一个应用系统预分配了 5TB 的空间，但该应用却只需要 1TB 的容量，这就造成了 4TB 的容量浪费，而且这 4TB 容量被分配了之后，很难再被别的应用系统使用。即使是最优秀的系统管理员，也不可能恰如其分的为应用分配好存储资源，而没有任何的浪费。根据业界的权威统计，由于预分配了太大的存储空间而导致的资源浪费，大约占总存储空间的 30% 左右。

aSAN 采用了自动精简配置技术有效的解决了存储资源的空间分配难题，提高了资源利用率。采用自动精简配置技术的数据卷分配给用户的是一个逻辑的虚拟容量，而不是一个固定的物理空间，只有当用户向该逻辑资源真正写数据时，才按照预先设定好的策略从物理空间分配实际容量。

2.5.4.2 aSAN 私网链路聚合

aSAN 的私网链路聚合是为了提高网络可靠性和性能设置而提出的。使用 aSAN 私网链路聚合不需要交换机上配置链路聚合，由存储私网负责链路聚合的功能，使用普通的二层交换机，保证正确的连接即可。

传统的链路聚合是按主机 IP 进行均分，即每两台主机间只能用一条物理链路。而 aSAN 私网链路聚合采用按照 TCP 连接进行均分，两台主机间的不同 TCP 连接可使用不同物理链路。在保障可靠性的同时，还达到了更加充分的利用所有链路资源的能力。

2.5.4.3 数据一致性检查

aSAN 采用一致性复制协议来保证多个副本数据的一致性，即只有当所有副本都写成功，才返回写入磁盘成功。正常情况下 aSAN 保证每个副本上的数据都是完全一致，从任一副本读到的数据都是相同的。如果某个副本中的某个磁盘短暂故障，aSAN 会暂时不写这个副本，等恢复后再恢复该副本上的数据；如果磁盘长时间或者永久故障，aSAN 会把这个磁盘从群集中移除掉，并为副本寻找新的副本磁盘，再通过重建机制使得数据在各个磁盘上的分布均匀。

3 深信服 aBos 一体机核心价值

3.1 高稳定性

深信服 aBos 超融合架构，通过软件定义分支数据中心的方式，将计算、存储、网络包括安全形成一个大的资源池，可以灵活快速的构建数据中心里的业务系统。虚拟机的热迁移技术、虚拟机的 HA 技术、存储的两副本技术保证了数据的安全性。传统分支 IT 基础架构网络设备需要双机部署，aBos 一体机以较低的成本进行集群部署即可满足业务可靠性。

3.2 简化分支机构 IT

深信服 aBos 一体机仅需一台服务器就可以解决安全防护、上网行为管理、下一代防火墙、新建销售办公业务系统等问题，简化了小型机构采购及部署多台不同功能、品牌设备的流程，不仅节省 IT 投入成本也大幅降低了运维成本，实现分支机构“ZERO IT”。

3.3 简化运维、集中管理

通过我们的配置向导，现在不再需要专业管理员出差去支持分支 IT 部署和运维，日常运维通过 BBC 管理平台也更为直观高效。aBos 一体机解决方案实现 IT 集中化，集中管理 BBC 可以部署在总部数据中心或者云端，可满足实时信息汇总、安全策略下发、日志集中管理、分级管理、远程接入分支维护等。

3.4 灵活部署、扩展性好

深信服 aBos 一体机采用超融合架构，用户可按需使用虚拟上网行为管理（vAC）、虚拟应用层防火墙（vAF）、虚拟广域网优化（vWOC）、虚拟 SSL VPN（vSSLVPN）等资源服务。用户可根据业务需求灵活组合 NFV 虚拟网络设备，若随着业务发展有新的网络功能需求时只需 license 激活即可使用无需重新购买和部署替换硬件设备。并可通过可视化的 WEB 管理平台即可管理所有虚拟设备，所画即所得的业务逻辑网络拓扑可实时监控网络流量，排查网络问题几分钟就可以解决。

4 超融合架构最佳实践

aBos 一体机可以应用到中小型网络数据中心建设、集团分支组网 IT 基础架构建设，尤其替代传统网关设备，满足基础路由、安全防护、上网审计、广域网优化，服务器虚拟化、存储虚拟化的需求。

以下为深信服 aBos 一体机三个实际应用案例分享

● 完美集团

项目背景：完美（中国）有限公司成立于 1994 年，是由马来西亚完美资源有限公司在中国投资设立的侨资企业。拥有四大生产基地、八家控股子公司的现代化企业集团，实现年销售过百亿人民币的目标。目前，在全国各省、自治区和直辖市设立了 34 家分支机构、6 家办事处、万余家服务网点。

众多的服务网点运维工作成为了 IT 运维中心最大的困扰。运维难度大、故障多、排障难成为了完美集团 IT 运维中心的“三座大山”。

解决之道：完美集团整合分支网络架构，在每个分支机构部署 aBos 一体机，实现网络基础路由以及安全审计功能。

方案价值：完美集团采用了深信服分支机构一体化解决方案后，解决了运维难题的同时节省了分支机构运维成本及 IT 基础建设成本。实现集中管理，分支无需专业人员投入，总部进行策略下发以及接入分支进行故障排查等。

● 申银万国期货

项目背景：申银万国期货有限公司系申银万国证券股份有限公司的控股子公司。随着申万期货的业务范围不断扩张，申万期货准备在全国各地部署 20 余家轻型营业部，加快现有业务的发展。为了保障营业部高效、稳定的运行，轻型营业部在快速安全接入、网络安全、数据安全以及安全审计变得尤为重要。同时广域网的运维和管理的要求也在不断的提升，从网络安全、应用安全、数据优化到业务数据安全的要求都在不断的提高。

解决之道：在每个营业部出口处部署 1 台深信服分支一体机 aBos，总部数据中心和 IDC 数据中心之间部署广域网优化设备。实现分支与总部之间的快速安全的加速 VPN 组网，同时实现分支到数据中心链路互备。

方案价值：申万采用深信服 aBos 一体机解决方案后，解决分支网络管理难问题，简化分公司网路架构。分支无需专业 IT 管理人员，总部部署 BBC 集中管理平台。分公司 aBos 一体机开机即用，总部直接下发安全策略、流控策略等即可。后期可以远程接入 aBos 一体机进行故障定位，总部运维人员可监控分支安全告警、网路设备状态。

- **甘肃省公安厅交通警察总队**

项目背景：目前甘肃省公安厅交警队有 40 人的外包运维团队，但是如何对运维团队进行合理的管理，使其成为具备强大的运维能力、具有高效机制的团队一直困扰着甘肃公安厅交警队。要求对运维办公区进行安全防护以及行为审计，满足安全规范、合规需求。

解决之道：深信服为交警总队提供一套综合解决方案，改造方案充分考虑到安全防护和上网行为审计功能。通过在运维办公区出口部署 aBos 一体机设备，集成的 NGAF、AC 功能模块的整体解决方案，形成 L2-L7 层立体安全防护体系，实现内网安全防护、流量控制、安全审计的效果，确保内网办公高安全和高可用。

方案价值：

- 1、方案整合运维办公区网络设备，一体化方案很好满足区域办公的所有网络建设需求，实现内网 IT 网络建设改造。
- 2、满足 82 号令，实现上网行为审计，规范化运维外包人员上网行为。完善的流控策略保证非核心业务的带宽，提高工作效率。
- 3、完善的内网 APT 防护、IPS 防护机制，保证内网 PC 主机安全，同时限制外发文件等流量。合理的区域隔离，保护单位数据安全。



深圳市南山区学苑大道 1 0 0 1 号南山智园 A1 栋
邮编：518055

Add: Block A1, Nanshan iPark,
No.1001 XueYuan Road, Nanshan District,
Shenzhen 518055, P.R.China

产品咨询热线：400-806-6868

Email:master@sangfor.com.cn