紫光集团 核心企业 | H3C 数字化解决方案领导者

互联网技术专刊

新华三集团

新华三集团

北京总部
北京市朝阳区广顺南大街8号院 利星行中心1号楼
邮编:100102

杭州总部
杭州市滨江区长河路466号
邮编:310052

www.h3c.com deindex.h3c.com

Copyright © 2022新华三集团 保留一切权利
免责声明：虽然新华三集团试图在本资料中提供准确的信息，但不保证本资料的内容不含有技术性误差或印刷性错误，
为此新华三集团对本资料中信息的准确性不承担任何责任。新华三集团保留在没有任何通知或提示的情况下对本资料的内容进行修改的权利。

CN-170X30-20220105-BR-HZ-V1.0

目录

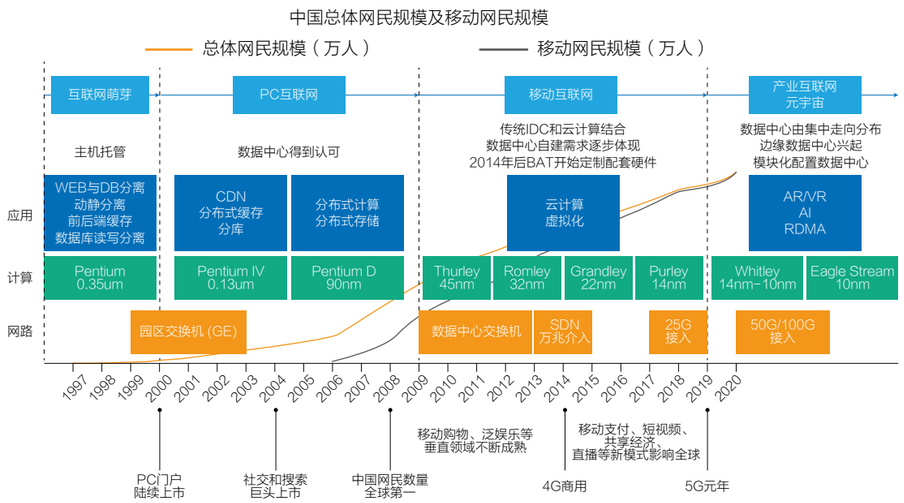
CONTENTS

刊首语

新华三集团互联网系统部技术总监 陈明

日前，由中国互联网协会组织编撰的《中国互联网发展报告（2021）》正式发布。《报告》显示，截至 2020 年底，中国网民规模为 9.89 亿人，互联网普及率达到 70.4%，特别是移动互联网用户总数超过 16 亿；5G 网络用户数超过 1.6 亿，约占全球 5G 总用户数的 89%；数字经济持续快速增长，信息技术与实体经济加速融合，规模达到 39.2 万亿元，总量跃居世界第二。我国互联网行业创新能力不断提升，数字经济蓬勃发展，网络治理逐步完善，为网络强国建设提供有力支撑。

回顾中国互联网的发展史，其实也是一部 ICT 技术的演进史：



1999 年 7 月，中华网在美国纳斯达克上市，2000 年，新浪、网易、搜狐均在美国上市，门户时代成为中国互联网的启蒙阶段。在这个互联网的萌芽期，互联网的应用架构在传统架构的基础上不断演进，以满足用户数量的快速提升，网络和计算等 ICT 基础设施则基本沿用传统架构。

随着进入 PC 互联网时代，互联网应用架构开始向分布式架构演进。分布式架构的广泛应用使数据中心内的流量模型发生重大变化，对网络设备的转发性能、缓存大小、表项空间等提出了更高的要求，专门应用于数据中心场景的数据中心交换机应运而生。

2009 年，随着微博、微信的相继发布，标准着中国互联网进入移动互联网时代。在 4G 通信技术的强力支撑下，智能化设备全面普及，接入互联网的门槛大幅降低，移动支付、手机健身、网络租车、移动视频、移动阅读……海量应用覆盖了百姓生活的方方面面。移动互联网以其泛在、连接、智能、普惠等突出优势，有力推动了互联网和实体经济深度融合，成为创新发展新领域、公共服务新平台、信息分享新渠道。在移动互联网时代，互联网的应用架构开始向着虚拟化、云计算、云原生的方向演进。

随着 5G 技术的普及，互联网又一次来到了新的关键发展节点。2019 年，有些业界专家认为，互联网将进入“产业互联网”时间。2021 年，有些业界专家将“元宇宙”定义为下一代互联网。元宇宙融合区块链、5G、VR、AR、人工智能、物联网、大数据等前沿数字技术，让每个人都可以摆脱物理世界中现实条件的约束，从而在全新数字空间中成就更好的自我，实现自身价值的最大化。在这个阶段，前沿的技术有望实现融合应用，区块链创造数字化的资产，智能合约构建智能经济体系，物联网让物理世界的现实物体向数字世界广泛映射，人工智能成为全球数字网络的智慧大脑并创造“数字人”，AR 实现数字世界与物理世界的叠加，5G 网络、云计算、边缘计算正在构建更加宏伟的数字新空间。其实无论是“产业互联网”还是“元宇宙”，强调的均是促进数字经济与实体经济实现更深层次的融合，从而助力“百行千业”全面转型升级，为实体产业开辟全新的发展空间。

从“移动互联网”到“元宇宙”的演进，由此也带动了 ICT 架构的重大变革：

在计算领域，呈现出以下趋势：

- 平台多元化： ARM/RISC-V 架构通过占领手机，物联网等终端设备，开始进入数据中心市场。
- 进入数据为中心的计算时代：需要一个快速的网络提升微服务之间的通信效率，同时卸载 host CPU 对流量处理的开销。因此，异构计算快速增长（智能网卡、DPU、IPU）
- 硬件设备 API 化：云管理平台（IaaS）通过 RestAPI 对设备（CPU、GPU、FPGA、AI、NIC 等）继续更加细致的管理，最好可以对每个设备进行独立操作（远程替换、升级、资源分配）
- 边缘服务器需求加速：预计 2029 年边缘服务器将达到 700-900 亿美元规模，将占据服务器市场规模的 50% 左右。

在网络领域，呈现出以下趋势：

- NASS（Network As A Service ）：构建一种服务化的网络，让应用更加容易的调度网络资源。应用的视角不再看到具体的 IP，路由策略，更关注服务的状态，限流，熔断监控等。
- 高带宽、低延时：带宽持续演进，规模日益扩大，低延时网络赋能计算与存储
- 芯片为王：设备形态简单化，使用单芯片的盒式设备构建大型网络成为趋势，建设成本大幅降低。
- 分布式云网络：边缘计算是一种分布式网络基础设施，构建一张分布式云网络，实现云 – 边协同、边 – 边协同。

新华三集团从上世纪 90 年代就开始积极参与互联网行业的基础设施建设。多年来，新华三密切跟踪互联网行业的需求及技术发展趋势，不断创新，不断超越，与互联网用户共同成长，产品经受了互联网环境下最苛刻的考验，成为中国互联网行业最主要的基础设施供应商。

网络与计算是互联网数据中心最重要的基础设施。面对日益庞大和复杂的业务需求，互联网行业迫切需要为数据中心注入新血液，为业务发展增添动力。新华三拥有全球领先的研发体系和创新团队，成熟的产品和解决方案覆盖各类应用需求，并以深刻的本土洞察力，打造出中国互联网企业最需要的自主创新的产品和解决方案。

在互联网行业，新华三集团秉承开放、合作、共赢的理念。新华三已经由设备提供商转变为互联网企业的战略合作伙伴。多年来，新华三在高性能网络、混合云、专有云、软件定义网络、NFV 网络功能虚拟化、可视化运维、人工智能等前沿技术领域与互联网企业开展了深入的技术交流与合作。未来，新华三愿意继续保持与互联网客户的深入合作，共同推动中国互联网的发展与进步。

新华三集团：数字化解决方案领导者

- 新华三（H3C）是紫光集团核心企业，业务覆盖“芯－云－网－边－端”全产业链
- 拥有芯片、计算、存储、网络、5G、安全、终端等全方位的数字化基础设施整体能力
- 提供云计算、大数据、人工智能、工业互联网、信息安全、智能联接、边缘计算等在内的一站式数字化解决方案，以及端到端的技术服务
- 新华三是 HPE® 服务器、存储和技术服务的中国独家提供商

13，000+
名员工

12，700+
件专利申请
（截至2021年11月底）

368 亿万
销售收入
（2020 年度）

50%
研发人员占比

14%
研发投入占比

5,000 万 +
台设备在线运行

市场份额持续领先

市场份额领先

第一

- 2021Q1-Q3，中国企业网交换机市场份额第一，38.1%
- 2021Q1-Q3，中国企业级WLAN市场份额第一，29.9%
- 2021Q1-Q3，中国Wi-Fi 6市场份额第一，32.4%
- 2021Q1-Q3，中国刀片服务器市场份额第一，45.1%
- 2021H1，中国分布式存储块存储市场份额第一，28.4%
- 2017年-2021年，连续五年中国网络管理软件市场份额第一
- 2016年-2020年，连续五年中国SDN（软件）市场份额第一
- 2016年-2020年，连续五年中国云管理平台市场份额第一
- 2015年-2020年，连续六年国产品牌服务器虚拟化市场份额第一
- 2021H1，中国IT统一运维软件市场份额第一，13.3%
- 2021H1，中国网络管理服务市场份额第一，3.9%
- 2021H1，中国支持服务市场份额第一，6.6%
- 2021H1，中国硬件部署与支持服务市场份额第一，11.2%
- 2021年，城轨云基础设施市场份额第一，62.9%

第二

- 2021Q1-Q3，中国企业网市场份额第二，35.9%
- 2021Q1-Q3，中国以太网交换机市场份额第二，36.9%
- 2021Q1-Q3，中国数据中心交换机市场份额第二，36.8%
- 2021Q1-Q3，中国园区交换机市场份额第二，36.9%
- 2021Q1-Q3，中国路由器市场份额第二，10.0%
- 2021Q1-Q3，中国企业网路由器市场份额第二，31.1%
- 2021H1，中国SD-WAN市场份额第二，16.6%
- 2021H1，中国SD-WAN基础架构市场份额第二，22.7%
- 2021Q1-Q3，中国服务器市场份额第二，17.3%
- 2021Q1-Q3，中国x86服务器市场份额第二，17.2%
- 2021Q1-Q3，中国non-x86服务器市场份额第二，21.4%
- 2021Q1-Q3，中国存储市场份额第二，12.2%
- 2021H1，中国备份一体机市场份额第二，11.7%
- 2021Q1-Q3，中国UTM防火墙市场份额第二，23.7%
- 2021H1，中国超融合市场份额第二，19.0%
- 2021H1，中国分布式存储市场份额第二，16.8%
- 2021H1，云系统软件（Cloud OS）市场份额第二，21.7%
- 2021H1，中国IT咨询服务市场份额第二，5.5%
- 2021H1，中国网络咨询和集成服务市场份额第二，3.2%
- 2021H1，中国IT外包服务市场份额第二，4.7%

第三

- 2021Q1-Q3，中国全闪存储（AFA）市场份额第三，9.7%
- 2021Q1-Q3，中国安全内容管理市场份额第三，6.8%
- 2021Q1-Q3，中国入侵防御保护（IDP）市场份额第三，15.3%
- 2021Q1-Q3，中国应用交付（ADC）市场份额第三，12.5%
- 2021H1，中国软件定义计算（SDC）市场份额第三，15.6%
- 2020年，紫光位居中国政务云基础设施市场份额第三，13.2%
- 2020年，紫光位居中国政务数据治理解决方案市场份额第三，5.7%

★ 数据来源：IDC（截至 2021Q3），计世资讯

持续创新成果显著

连续十年，发明专利授权量位居浙江省第一名！

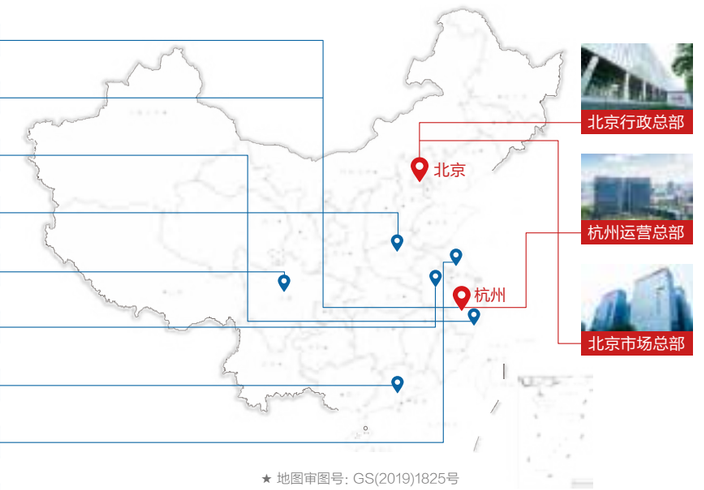
连续九年，荣列浙江省高新技术企业创新能力百强榜首！

90%	12,700+	4
发明专利占比	专利申请量	平均每个工作日专利申请量
国家级高新技术企业	国家科学技术进步奖	知识产权示范企业
国家企业技术中心	国家重点新产品奖	CMMI5 级认证厂商
国家规划布局内重点软件企业	浙江省科技进步奖	知识创新能力评价 AAA 级企业

★专利数据截至 2021 年 11 月底

产业布局覆盖全国

- 新华三技术·杭州
- 新华三人工智能·杭州
- 新华三制造物流·海宁
- 新华三大数据·郑州
- 新华三云计算·成都
- 新华三信息安全·合肥
- 新华三智能终端·广州
- 新华三工业互联网·苏州
- ……



★ 地图审图号：GS(2019)1825号

双 总部

8 大基地

9 大产业

50+ 个销售与服务机构

助力“双碳”目标达成，推动“双碳”经济发展

新华三是“双碳”战略的践行者和推动者。我们以持续创新推进产品与解决方案绿色变革； 依托数字化解决方案，我们深度赋能城市与行业重构发展模式，助力区域协同，实现全面的提质增效和绿色发展的目标；此外，我们更以自身转型促进节能降耗，履行时代使命。

- 技术创新 打造绿色产品及方案
- 使能重点领域 实现“双碳”目标

- 助力区域协同 推动“双碳”经济腾飞
- 全面推进自身节能减排 践行碳中和使命

数据时代 DCN 网络架构设计

互联网系统部 王京

网络是 IT 基础设施重要的组成部分，是联接所有 IaaS 层资源对上提供服务的基础。在数据时代，云计算、大数据和人工智能的核心是数据，网络就是承载数据流动的高速公路。

从过去以严谨、规范著称的金融行业数据中心，再到现在引领技术潮流的互联网公司数据中心，数据中心网络近十余年的变化可谓是日新月异。

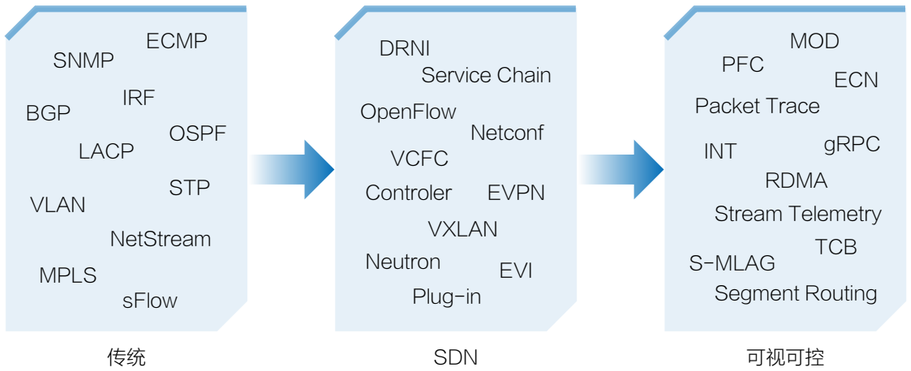


图 1：网络技术快速更迭

大量的网络技术如雨后春笋般涌现，一个新技术往往还没有规模化的应用和落地，就被另一个更优的新技术所代替。层出不穷的协议和技术就像过眼烟云一般，如果再过十年，回看这段网络发展的高速上升期，相信每一名网络从业者都会有很多感悟。

如今，云原生技术发展迅速，已经有超过 200 个项目，应用的开发、发布、运行、运维也发生了改变。大量的应用系统采用云原生技术所构建，容器作为业务负载的最小单元，其特点是敏捷、一致性和具备极强的复制扩展能力。集群由数量众多的容器构成，数量级远高于 VM。同时，更细粒度的资源分配机制和保证可靠性的分布策略，导致业务容器以及各种分布式系统组件间的跨 Node 通信和交互更加频繁，这些都依赖于外部网络提供可靠的端到端转发，同时对流量控制和可视化提出了更高的要求。

其次，随着大数据和人工智能技术的普及，基于此的推荐系统、图像搜索和识别、语音交互、人脸识别、机器翻译等被广泛应用。而且，大数据和人工智能已成为企业经营管理和应对市场竞争的重要武器，海量的数据被存储下来用于分析挖掘。从数据处理、模型训练（机器学习 / 深度学习）到上线服务，每个环节都依赖于强大的算力和海量的数据，对计算资源和存储资源的消耗与日俱增，促使数据中心建设向大规模、超大规模演进（截至 2021 年第三季度末，全球超大规模数据中心达到 700 个），与之配套的网络规模也越来越大，网络自动化和智能运维成为刚需。

最后，不得不提的是长、短视频、直播、VR/AR 等视频流媒体近两年的爆发式增长，而且渗透到了新闻、教育、购物、社交、出行、游戏娱乐等各个领域，用户规模庞大、使用时长高，根据相关报告分析，到 2022 年视频流媒体将占到全球互联网流量的 82%。伴随着 5G 终端的快速普及，用户对高质量视频、低延迟观看体验的期望不断提高，又进一步推升了对网络带宽的消耗。

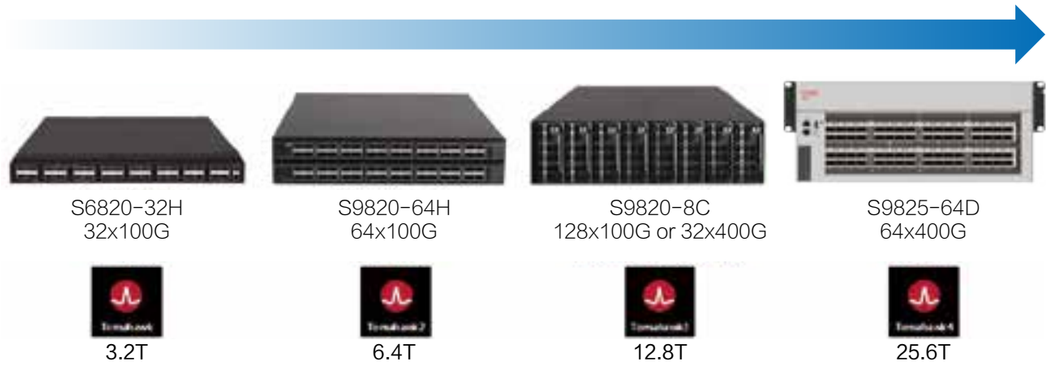
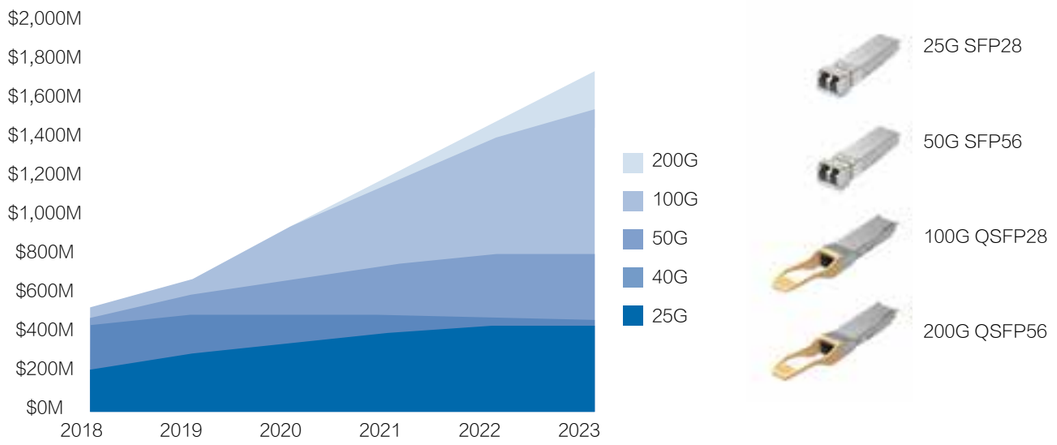


图 2：H3C 基于 Tomahawk 系列芯片的四代产品演进

面对业务需求侧的变化趋势以及网络技术的快速发展，数据中心网络设备的迭代速度也随之加快。当前，数据中心交换机不到 2 年就更新一代产品，且每一代新产品的推出都提供近乎翻倍的性能，更高的吞吐、更大的表项、更多的特性，产品在网络中的角色定位也更有针对性。



Source: Crehan Long-range Forecast - Ethernet Adapter forecast, January 2019

在服务器端网卡和光模块整体产业环境的推动下，数据中心接入链路带宽从 10G->25G->50G->100G->200G 演进，互联链路带宽从 40G->100G->200G->400G 演进。主力场景从 2018 年至今的 25G 接入 +100G 互联组合，在 2021 年开始过渡到 100G 接入 +400G 互联组合。GPU 场景将从 100G 接入过渡到 200G 接入。

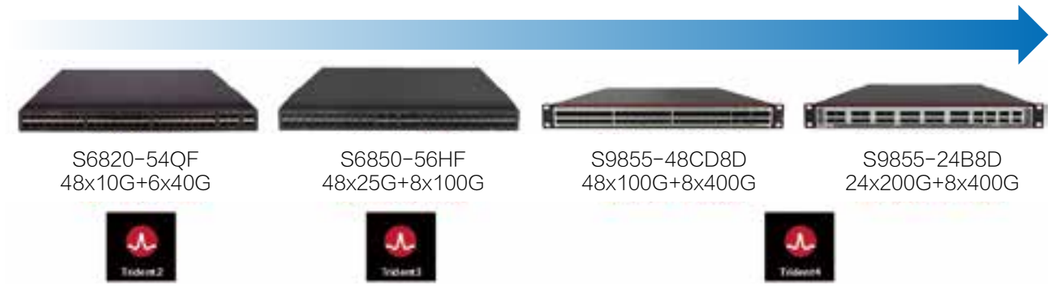


图 4：H3C 基于 Trident 系列芯片的产品演进

综合前述背景，再来看网络架构。DCN 网络架构的选择是受到业务需求、当前技术条件、设备成本、管理成本、人力投入等诸多因素的影响。没有一种架构可以驾驭所有客户的场景和需求，需要全面考量和平衡后进行选择。

第一种：适用于中小型、中型规模的数据中心

两级 Clos 架构是应用最早、最普遍的网络架构，现如今依然是很多行业客户的首选。整网设备只有两种角色，数据转发路径短，跨 Leaf 一跳可达，路径和时延具有很强的一致性。统一的接入方式也给上线部署和水平扩展方面带来了很大的便利条件，例如 BGP 协议的部署，策略的控制，日常维护和问题排查等。非常适合体量中小，运维人员编制少的企业。

两级 Clos 架构对 Spine 交换机的性能和可靠性要求很高，一般采用数据中心框式核心交换机产品。基于可变信元转发和 VoQ 调度机制，保证 Spine 设备内部的严格无阻塞交换，分布式大缓存的配置在应对流量突发等方面具备天然的优势。框式核心交换机有独立的控制平面、转发平面和支撑系统，而且采用冗余设计，整个系统在可靠性上远高于盒式交换机。

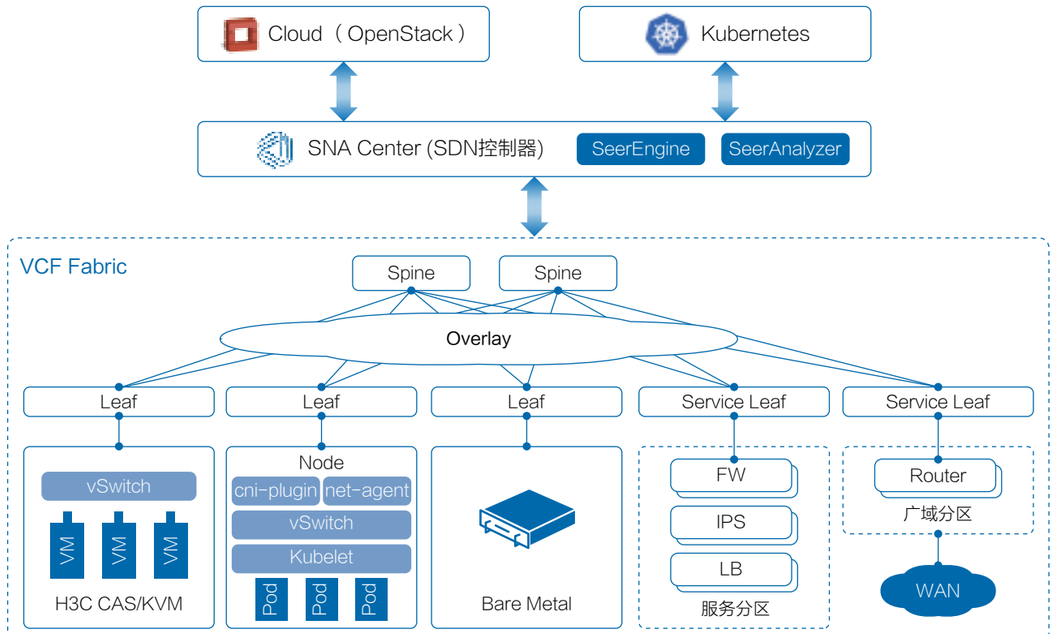


图 5：H3C AD-DC 应用驱动数据中心解决方案

两级Clos架构在和商用SDN控制器方案的适配上更成熟，结合 SDN 控制器可快速构建基于 EVPN 的网络 Overlay 方案，降低东西向和南北向服务链的部署难度，满足云场景下网络对 VM、裸金属、容器等全形态计算资源联动的需求。

另外，该架构也同样适用于大型企业在各地部署的汇聚机房和边缘机房，用于构建边缘计算网络，缓解主干网络压力和降低访问时延。

两级 Clos 架构在和商用 SDN 控制器方案的适配上更成熟，结合 SDN 控制器可快速构建基于 EVPN 的网络 Overlay 方案，降低东西向和南北向服务链的部署难度，满足云场景下网络对 VM、裸金属、容器等全形态计算资源联动的需求。

另外，该架构也同样适用于大型企业在各地部署的汇聚机房和边缘机房，用于构建边缘计算网络，缓解主干网络压力和降低访问时延。

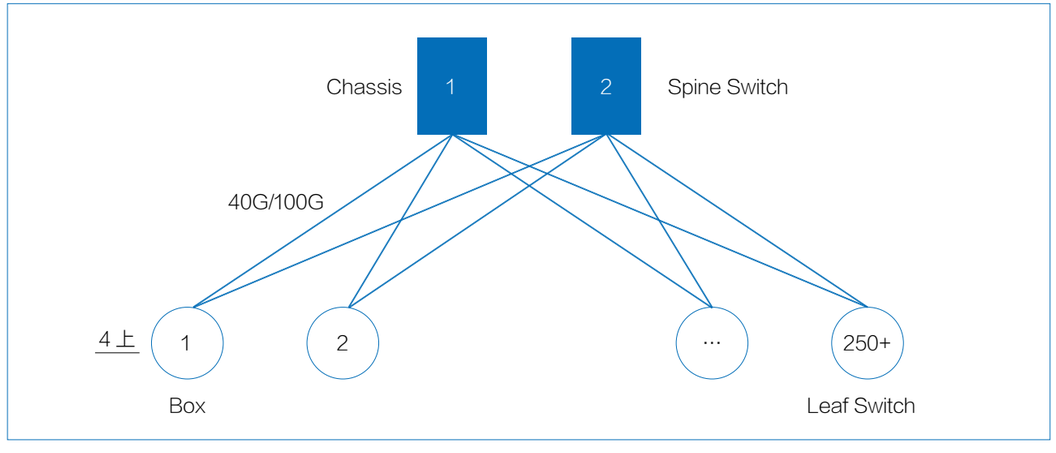


图 6：两级 Clos 双 Spine 示例

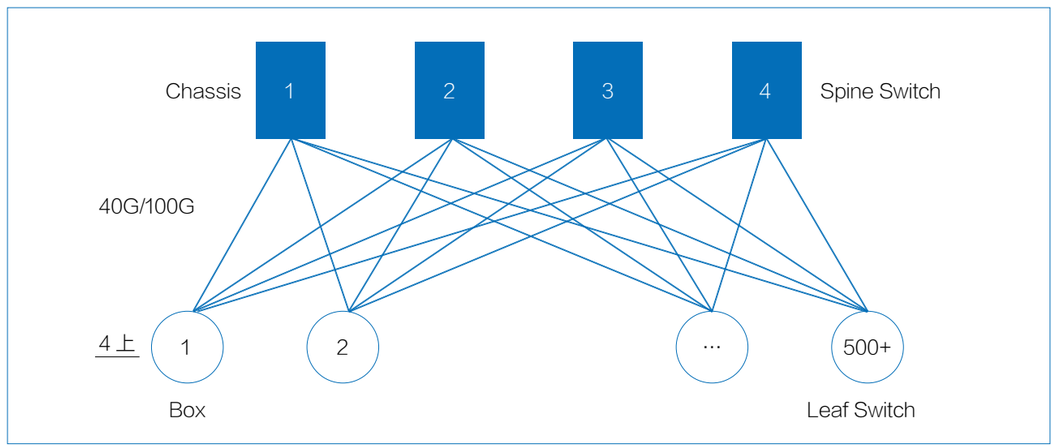


图 7：两级 Clos 四 Spine 示例

Spine 采用 2 台或 4 台框式核心交换机，Leaf 交换机每台 4 上行，在保证 3:1 收敛的情况下（10G Leaf 4*40G 上行，48*10G 下行；25G Leaf 4*100G 上行，48*25G 下行），可支撑的服务器（双上行）规模分别为 5000+ 台和 10000+ 台。

由拓扑可见，两级 Clos 架构的网络规模也就是水平扩展能力，是受到 Spine 设备所能提供的端口总数限制的（设备台数 * 单机提供的端口数）。由于 Leaf 交换机上行端口数量是固定的（通常 4~8 个），因此 Spine 层交换机的台数也是受限的，不能持续增加。

图 8：H3C 数据中心框式核心交换机的演进

当前，框式核心交换机支持的业务板槽位数最多是 16 个，历史上曾经达到过 18 个（注：在十一年前的 2009 年，H3C 推出的初代旗舰数据中心核心交换机 S12518，具备 9 个交换网板和 18 个业务板槽位，是迄今为止的槽位巅峰）。在 1U 左右的空间内，单槽位支持的面板端口数普遍最多是 48 个。这样，整机所能提供的最大面板端口数为 768 个。

在物理空间有限的条件下，如何能进一步提升 Spine 交换机的接入能力，将两级 Clos 架构的网络规模扩展到极致，满足特定场景的需求，就需要用到端口拆分方案。

图 9：单板 + 光纤配线箱

端口拆分是采用更高速率的端口拆分成多个低速端口使用，变向提高单槽位的端口密度，增加整机的接入能力。例如，在 10G 组网时代，采用更高速率的 40G 业务板，通过将 1 个 40G 端口拆分成 4 个 10G 端口使用。这样，一块 36 端口 40G 业务板可以拆分出 144 个 10G 端口，是一块 48 端口 10G 业务板端口密度的 3 倍。在 2021 年开始规模部署的 400G、200G 设备上，这种端口拆分方案依然在沿用，来解决诸多场景下的网络互联需求。

第二种：适用于中型、大型规模的数据中心

两级 Clos 架构所支撑的服务器规模一般小于 20000 台，三级 Clos 架构的引入解决了两级 Clos 架构在网络规模上的瓶颈。三级 Clos 架构在两级 Clos 架构的中间增加了一级汇聚交换机（Pod Spine），由一组 Pod Spine 交换机和其下连的所有 Leaf 交换机一起组成一个 Pod，通过 Spine 层交换机将多个 Pod 互连组成整个网络。增加 Pod 的数量即可实现网络的水平扩展，大幅提升了网络的扩展能力。同时，以 Pod 为单位进行业务部署，在适配多种业务需求、提供差异化服务、隔离性等方面，三级 Clos 架构更具灵活性。

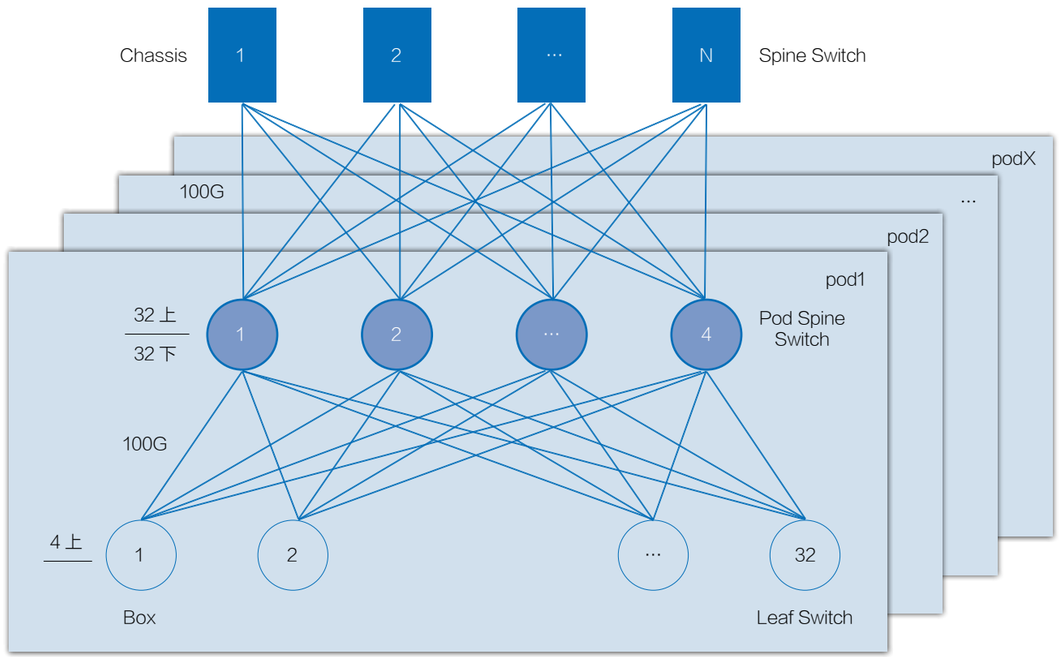


图 10：三级 Clos 示例 A

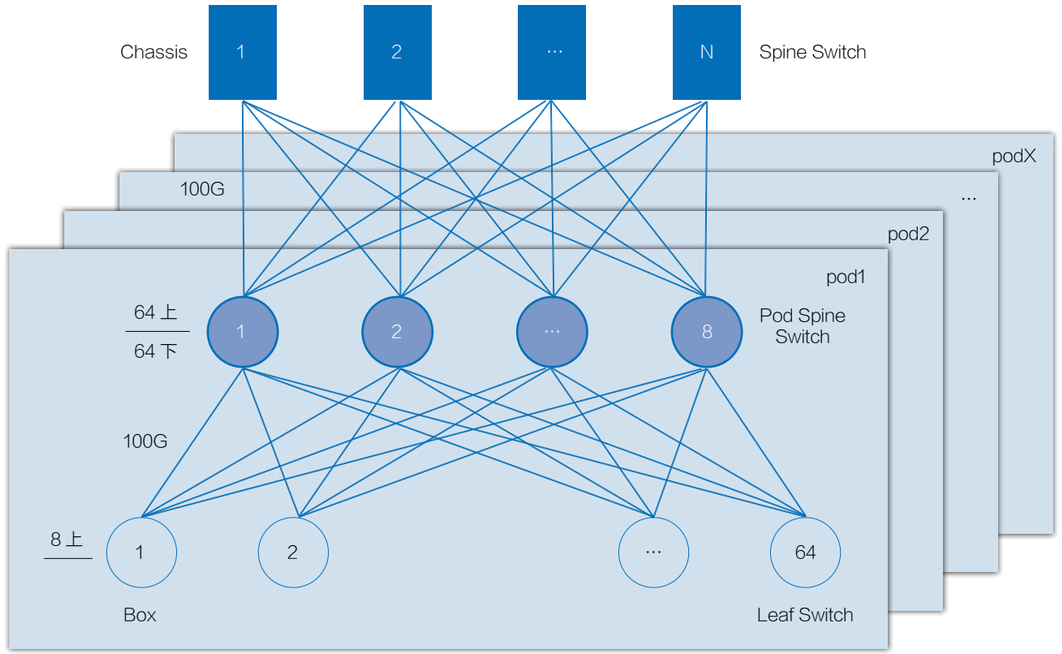


图 11：三级 Clos 示例 B

三级 Clos 架构在每个 Pod 内部，Pod Spine 采用 4 台或 8 台高密 100G 盒式交换机，Pod Spine 半数端口用于上接 Spine，半数端口用于下接 Leaf，Leaf 交换机每台 4 上行或 8 上行，典型场景如下：

A 场景：Pod Spine 采用 4 台 64 口 100G 盒式交换机（S9820-64H），Leaf 交换机每台 4 上行，在保证 Pod 内 3:1 收敛的情况下（25G Leaf 4*100G 上行，48*25G 下行），单 Pod 可支撑的服务器（双上行）规模为 768 台。

B 场景：Pod Spine 采用 8 台 128 口 100G 盒式交换机（S9820-8C），Leaf 交换机每台 8 上行，在保证 Pod 内 1.5:1 收敛的情况下（25G Leaf 8*100G 上行，48*25G 下行），单 Pod 可支撑的服务器（双上行）规模为 1536 台。在保证 1:1 收敛的情况下（25G Leaf 8*100G 上行，32*25G 下行），单 Pod 可支撑的服务器（双上行）规模为 1024 台。

由于高密汇聚交换机 Pod Spine 的引入，Spine 层的框式核心交换机突破了个位数限制，可以部署数十台，Spine 层框式核心交换机提供的总端口数可用于连接数十个 Pod，整个网络可以支撑服务器规模超过 10 万台。

另外，通过调整 Pod 内 Pod Spine 交换机的上、下行端口比例，可以灵活定义每个 Pod 的收敛比，在满足不同业务需求的同时还有助于降低成本，避免不必要的浪费。

第三种：适用于大型、超大型规模的数据中心

基于盒式设备的多平面组网架构，是当前头部互联网公司采用的最新架构，用于组建大规模和超大规模的数据中心网络。其架构最早源于 Facebook 的 F4，由于构建该网络的两代交换机 6-pack 和 Backpack 都是基于多芯片（12 颗）设计的，一台设备 12 个控制平面在管理和部署上有诸多不便，而且成本也较高。F4 演进到 F16 后，得益于芯片能力的提升，用于构建 F16 的交换机 Minipack 采用单芯片设计，功耗、成本和技术门槛大幅降低，方案也更加成熟，自此这种架构开始被国内互联网公司引入。

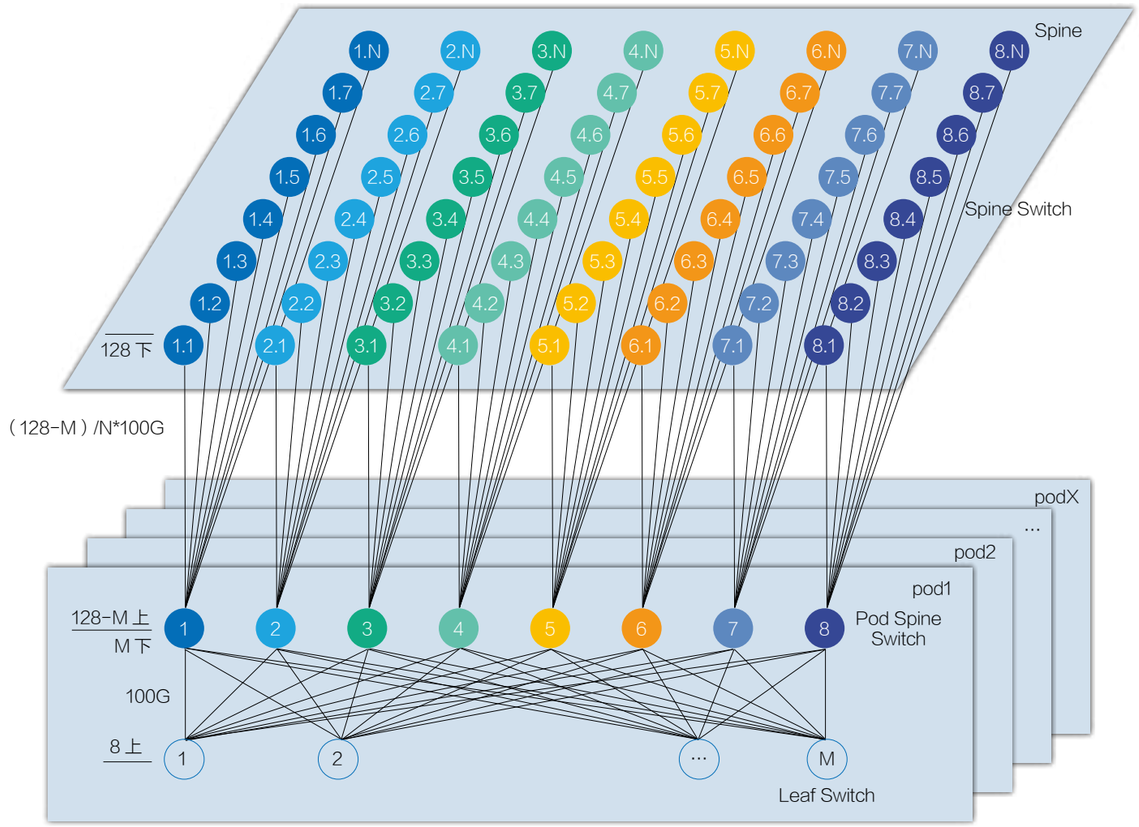


图 12：基于盒式设备的多平面组网示例

《Introducing data center fabric, the next-generation Facebook data center network》 和《Reinventing Facebook's data center network》两篇原文对该架构进行了详细的阐述。相比三级 Clos 架构，基于盒式设备的多平面组网架构将 Spine 层的框式交换机也替换为了盒式交换机，这样，全网各层设备均由盒式交换机组建。在设备连接上，不同于三级 Clos 架构中每台 Pod Spine 都需要和所有 Spine 层交换机全互联，在新架构中将 Spine 层交换机分成多组（组数与每个 Pod 中 Pod Spine 交换机数量一致），每组中的 Spine 交换机构成一个平面（如图 12，Spine 层分成 8 个平面，用不同颜色进行了区分），每个 Pod 中的 Pod Spine 交换机只需要和对应平面中的 Spine 交换机全互联。这样，整个 Spine 层可以连接更多的 Pod，构建出超大规模的数据中心，支撑数十万级别的服务器，且随着盒式交换机性能的提升，该架构还可以持续的提升容量空间。

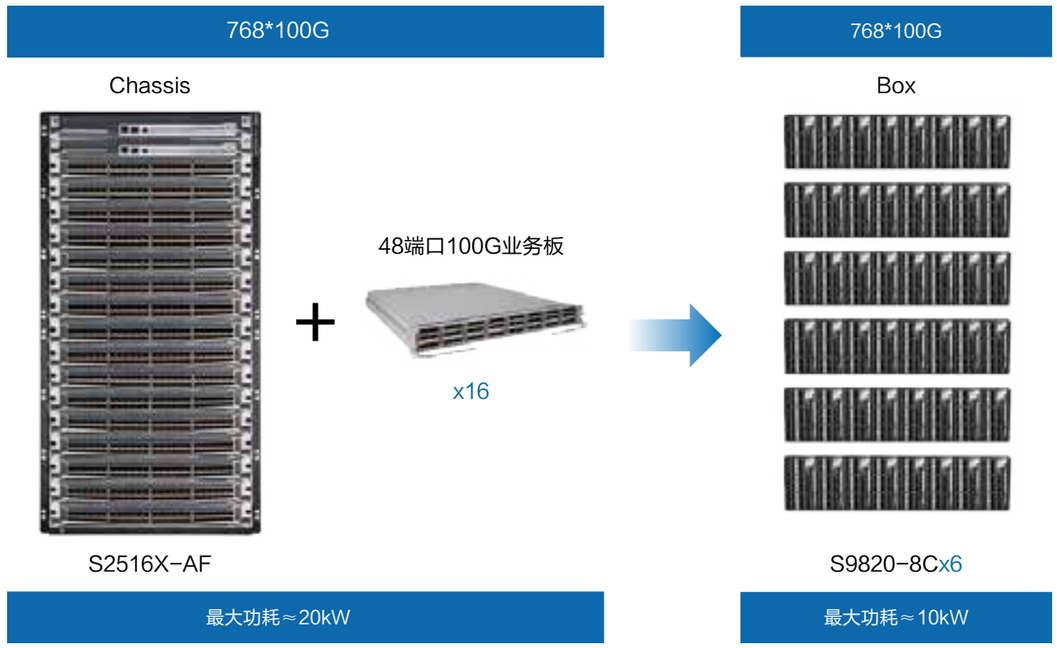


图 13：Chassis & Box

一台满配 48 口 100G 业务板的核心框式交换机 S12516X-AF 与 6 台 128 口 100G 的盒式交换机 S9820-8C，都可提供相同数量的 100G 端口（768 个），但是采用盒式交换机方案在成本、功耗和散热上具有明显的优势，同时也免去了以往框式核心交换机对机柜空间和配电的特殊要求。

由于 Spine 和 Pod Spine 采用了相同的设备，功能特性一致，转发延迟一致，非常便于一些新功能的开发并在全网去部署应用。并且，整个网络从 100G 组网向 200G、400G 组网以及后续更高速组网的演进上能保持同步。另外，得益于单芯片设计，采用盒式交换机组建的整个 Spine 层在转发延时上要明显低于采用框式设备，进一步降低了跨 Pod 的访问延迟。

当然，该架构也引入了新的问题，Spine 层设备的数量比采用框式交换机时的数量翻了多倍，而且盒式交换机的单机可靠性比框式核心交换机要低，这给网络管理和日常运维带来了很大的挑战。配套的管理平台、监控平台等都要能够适应这种变化，对网络运维团队也提出了更高的要求，如完善的人员分工，丰富的运维经验，有较强的技术能力和平台的开发能力等，通过对整个网络的把控来规避和降低设备和网络故障对业务的影响。

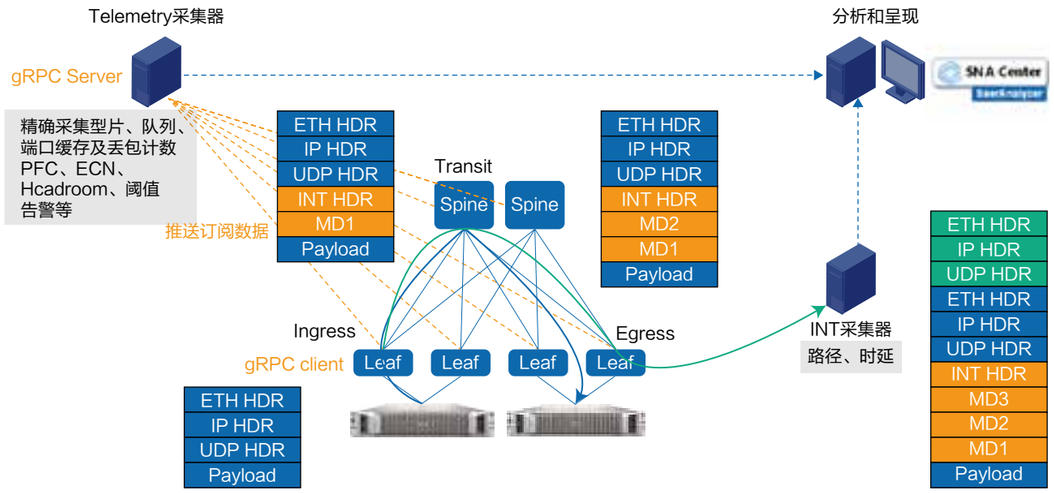


图 14：gRPC+INT

以上我们分别介绍了三种当前最典型的 DCN 网络架构。如何才能更好、更高效的驾驭这些网络，就离不开网络可视化技术。网络可视化技术不但能够完成端到端的流量监控、风险预警、协助故障排查。通过数据积累和分析，还能够用来指导和优化数据中心的网络架构设计（模型、收敛比、POD 规模等），是一种非常重要的技术手段。

网络可视化技术越来越主动、高效和智能。例如，通过 gRPC 可以更实时、高精度的从设备中采集所需的各种信息。通过 INT 或 Telemetry Stream 可以获取业务数据在网络中转发的路径和延迟。通过 TCB 可以对设备 MMU 进行监控，获取队列丢包的时间、原因、被丢弃报文。通过 MOD 可以检测报文在设备内部转发过程中发生的丢包，并能捕获到丢包原因及丢弃报文的特征。通过 Packet Trace 可深入转发逻辑，模拟报文在芯片中的转发，确定问题根因等。

利用大数据和 AI 技术将网络中采集到的各类信息进行存储、分析和预判，实现从数据训练模型，用模型洞察网络。

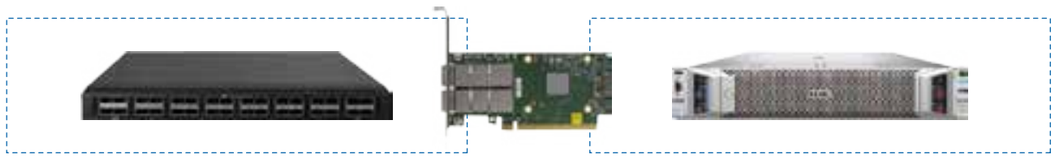


图 15：Smart NIC

未来，智能网卡将是 DCN 网络中重要的组成部分，具备可编程能力的智能网卡在释放 CPU 资源、实现高性能转发的同时，还拥有隧道封装 / 解封装、虚拟交换、加解密、RDMA 等功能，随着业务场景和需求的增加，越来越多的数据平面功能将由智能网卡来完成，打破了基于服务器或交换机各自实现的局限性，有望做到性能、功能和灵活性的完美平衡。智能网卡将接替 Leaf 交换机作为 DCN 网络的最末端，由此，网络架构、协议部署、可视化技术等也会因智能网卡的引入而发生改变，更利于端到端的性能优化和服务保障、端到端的探测和监控、SRv6 等新技术的应用，未来的 DCN 网络将会更上一个台阶，为不断丰富的上层业务提供更稳定、更高效、更灵活的网络服务。

50G/200G/400G 高速以太网技术

互联网系统部 刘佳豪

前言

以太网是目前应用最为普遍的局域网技术，从 1983 年 IEEE 802.3 标准诞生后的几十年时间里，以太网速率从 10M、100M 慢慢发展到了 GE、10GE 以及 100GE/400GE。目前在数据中心内部服务器还是以 10GE、25GE 网卡为主，网络架构相对应也主要是 10GE-40GE、25GE-100GE 的互连模型。随着业务的不断升级，带宽需求越来越大，25GE 服务器已经难以满足业务发展，因此后续 DCN 将逐渐进入 50GE 接入以及 100GE 接入时代。本文我们将一起探索 50G/200G/400G 高速以太网技术，主要为大家介绍标准协议技术发展以及光模块的发展。

50G PAM4 调制技术

以太网技术发展到 100G 时，物理层调制码型还主要是 NRZ 技术，但是 100G 之后这种调制方式面临着带宽提升带来成本激增以及调制方案充分度不足的问题。因此在 IEEE P802.3bs 工作组制定 400G 标准时有厂家提出用 PAM4 替代 NRZ，随之获得通过。之后的 200GE/50GE 标准也采用了该调制技术，因此可以说本文谈到的三个以太网标准均基于 PAM4 技术。

接下来我们来看一下 PAM4 技术对比 NRZ 技术在哪些地方做了改进：

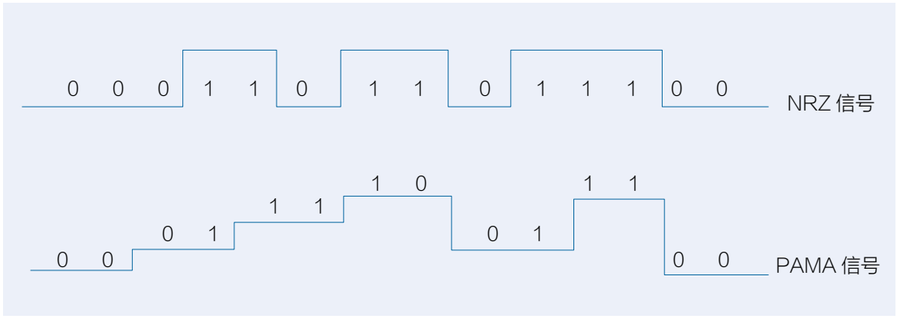


图1：NRZ 与PAM4 信号波形对比

如图 1 所示，波形图的主要参数有两个：比特周期 T 和幅度 H。信号传输所需带宽与比特周期相关，比特周期越短，所需带宽就越大。另外就是信噪比（SNR），信噪比与振幅相关，振幅越小，噪声敏感性越大。

NRZ 信号分别用高、低两种电平信号表示 1、0 两种数字逻辑信号。我们把 NRZ 的比特周期看做一个完整的 T，其幅度看做一个完整的 H。

再来看 PAM4 的信号波形，PAM4 信号用四种电平信号表示 00、01、10 以及 11 四种数字逻辑信号。PAM4 中“PAM”指的是脉冲幅度调制，“4”表示 4 级脉冲调制。顾名思义，PAM4 对比 NRZ 其比特周期是一致的，但是在幅度层面每个电平信号的幅度变成了 H/3。

以上就是两种技术的主要区别，其实简单来理解就是 NRZ 信号每个时钟周期可以传输 1bit 的逻辑信息，PAM4 信号每个时钟周期可以传输 2bit 的逻辑信息。因此，在同样波特率条件下，PAM4 信号传输效率是 NRZ 信号的两倍，同时还可以有效降低成本，以上这些优势却仅仅以牺牲噪声敏感性作为代价。所以，基于单通道 50G PAM4 技术的 400GE/200GE/50GE 可以很好得适应业务发展需求。

50GE 以太网技术

50GE 的相关标准是由 IEEE P802.3cd 工作组制定的，于 2016 年 5 月正式启动，其标准基线完成于 2016 年 9 月。IEEE 802.3cd 标准覆盖了 50GBASE-LR/KR/SR/FR PMD 子层，所有子层均采用了 50G PAM-4 技术。本小节我们重点讲解一下 50GE PMD 子层技术基本情况。

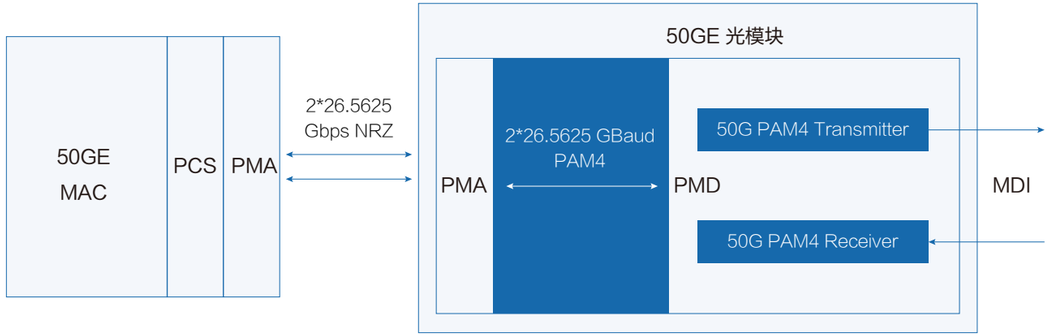


图2 50GBASE-LR 架构

我们先对上图所示架构中的一些主要组成部分做一个简单介绍：

PCS 子层负责对信号编 / 解码、加 / 去扰、Alignment 插入 / 去除、排序处理和控制等，PCS 子层还具备 FEC（前向纠错）的功能。

PMD 子层负责和物理传输媒介的接口，一个以太网速率存在多个不同的 PMD 子层，分别适配不同的物理接口。

PMA 子层负责适配 PCS 和 PMD 子层，提供映射、复用解复用、时钟恢复等功能。

MDI 代表了物理媒介，比如光纤、电缆等。

PMA 子层将两路 26.5625Gbaud NRZ 信号与一路 26.5625Gbaud PAM-4 信号进行双向转换。发送方向时，经过 PMA 完成速率和码型转换后的数据在 PMD 子层进行电光转换，由 Transmitter 发送到 MDI。接收方向时，PMD 子层从 Receiver 收到的数据并进行光电转换，然后发送给 PMA 进行速率和码型转换，以上就是 50GBASE-LR 架构解析。IEEE 802.3cd 定义的 50G 标准有以下几种：

50GBASE-CR，通过双绞铜线屏蔽电缆的一个通道（2 条同轴双绞线）传输 50 Gb / s，至少达到 3 m；

50GBASE-FR，在一个波长（总共 2 根光纤）上的 50 Gb / s 串行传输，可通过单模光纤电缆进行操作，最大可达 2 km；

50GBASE-KR，通过电背板的一个通道传输 50 Gb / s，在 13.28125 GHz 时总插入损耗小于 30 dB；

50GBASE-LR，在一个波长（总共 2 根光纤）上以 50 Gb / s 的速率进行串行传输，可通过单模光纤电缆进行操作，并达到至少 10 km；

50GBASE-SR，通过一个通道（总共 2 根光纤）的 50 Gb / s 传输速率，可通过多模光纤电缆进行操作，最大可达 100 m。

以上就是 50G 以太网技术的相关情况，50G 技术是 400G/200G 以太网技术的基础，所以本小节我们重点介绍了 50GBASE 标准的架构及组成。本小节主要想让大家了解 50G PAM-4 技术是如何应用在 50G 以太网中的，了解清晰之后，我们接来下重点看一下 400G 以太网技术以及 400G 光模块的发展情况。

400GE 以太网技术

400GE 标准基线是由 IEEE P802.3bs 工作组于 2015 年初制定的。随后，400GE 接口的多种 PMD 子层规格以及 AUI 电接口陆续提出来。

其中，IEEE 802.3bs 定义了传输距离 70-150m 的 SR16、传输距离为 500m 的 DR4、传输距离为 2km 的 FR8 以及传输距离为 10km 的 LR8。IEEE 802.3cm 定义了传输距离为 100m 的 SR8、SR4.2。100G Lambda MSA 定义了传输距离为 2km 的 FR4、传输距离为 10km 的 LR4。OIF 定义了传输距离为 80km 的 ZR。这些规格里边只有 SR16 仍采用 NRZ 的调制技术，其他规格均采用了 PAM4 调制。不同规格对应的电口、光口速率如图 3 所示。

距离	名称	速率	
		电口	光口
70-150m	SR16	16*25G	16*25G
100m	SR8	8*50G	8*50G
100m	SR4.2	8*50G	8*50G
500m	DR4	8*50G	4*100G
2km	FR8	8*50G	8*50G
10km	LR8	8*50G	8*50G
2km	FR4	8*50G	4*100G
10km	LR4	8*50G	4*100G
80km	ZR	DP-16QAM	

图 3 400G 接口的相关标准

1. 100m 传输距离主流标准是 SR8：

SR8 400G 光模块，该模块光口采用了 26.5625 Gbaud PAM-4 调制，所以单通道可以传输 50G，8 条通道传输 400G。下图是 400G SR8 光模块的内部通道图示：

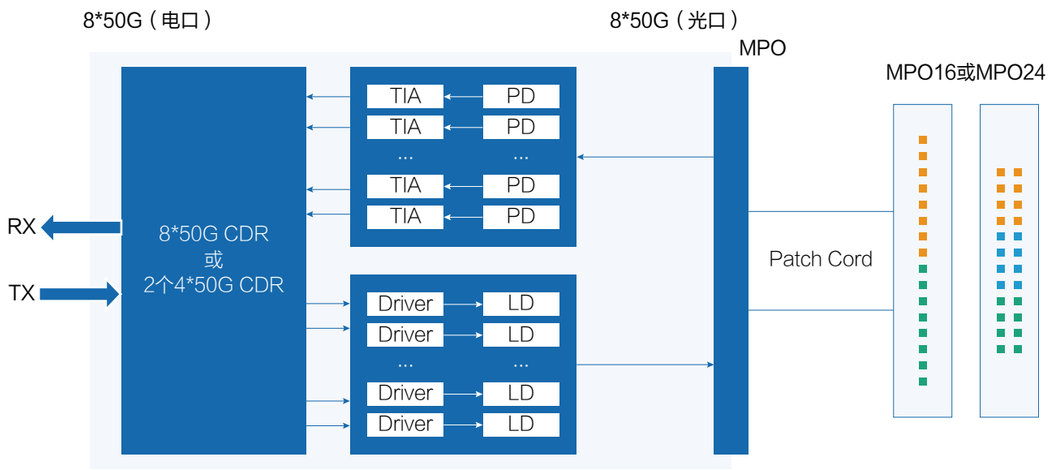


图 4 400G SR8 传输通道

从图 4 可以看到：400G SR8 光模块光口采用 16 芯光纤或者 24 芯光纤的 MPO 连接器，光纤传输通道均是 8 收 8 发，单通道 50G，可以实现 400G 以太网速率在 100m 距离内的传输。

2. 500m 传输距离主流标准是 DR4：

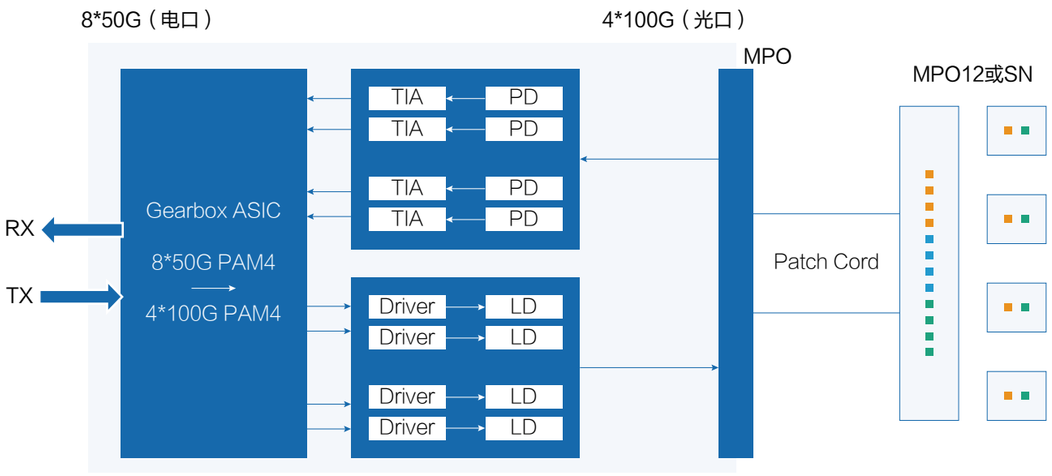


图 5 400G DR4 光模块及通道

400G DR4 光模块光口采用了 53.125 Gbaud PAM-4 调制，单通道可以传输 100G，4 通道可以传输 400G。从图 5 的通道图示来看，该模块采用了 12 芯光纤的 MPO 连接器，4 收 4 发，单通道传输 100G，可以实现 400G 以太网速率在 500m 距离内的传输。同时，也可以将 400G 一分四为 4 个 100G，与 4 个 100G 模块互连。在 AWS 的数据中心中，就大量存在 400GBASE-DR4 一分四的部署场景。

3. 2km/10km 传输距离主流标准是 FR4/LR4：

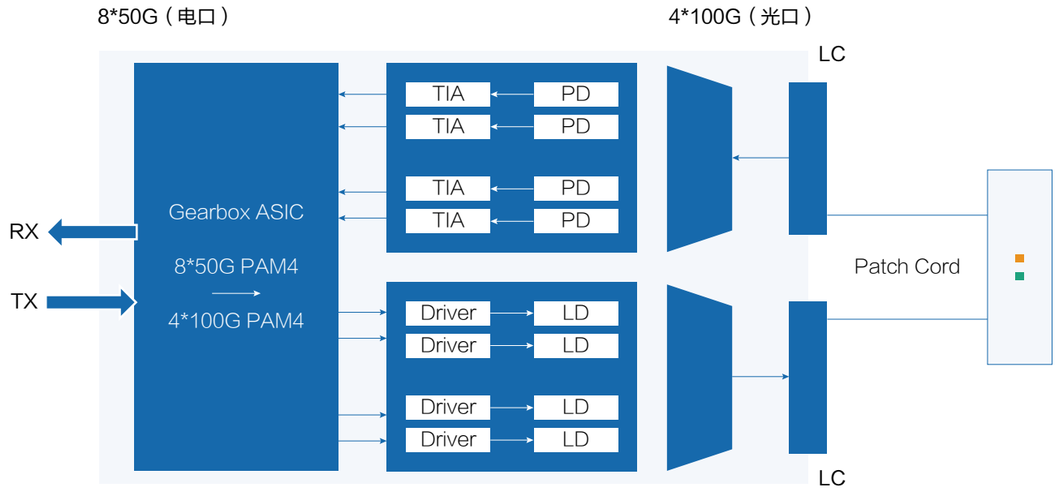


图 6 400G FR4/LR4 光模块及通道

400G FR4/LR4 光模块光口同样采用了 53.125 Gbaud PAM-4 调制，单波道可以传输 100G，4 波道可以传输 400G。从图 6 的通道图示来看，该模块采用了 LC 连接器，通过波分复用技术将 4 种不同波长的通道在一根光纤中传输，这样通过一收一发两根光纤便可以实现 400G 以太网速率在 2km/10km 距离内的传输。

200GE 以太网技术

200G 高速以太网技术在标准制定进度层面与 400G 极为相近，同样都由 IEEE P802.3bs 工作组所主要覆盖，同时个别标准例如 200GBASE-CR4、200GBASE-SR4 等由 IEEE P802.3cd 工作组制定。与此同时，在光模块架构上，200G 光模块与 400G 光模块也极为相似。200G 目前主要的标准有 200GBASE-SR4、200GBASE-DR4、200GBASE-FR4 以及 200GBASE-LR4，传输距离分别为 100m、500m、2km 以及 10km。

200GBASE-SR4 与 400GBASE-SR8 架构上类似，同样采用了 26.5625Gbaud PAM-4 调制，唯一区别的地方在于 200GBASE-SR4 是 4 条通道，而 400GBASE-

SR8 是 8 条通道，因此 200GBASE-SR4 光模块采用 12 芯的 MPO 连接器即可。

200GBASE-DR4 与 400GBASE-DR4 通道图示（见图 5）相似，只不过 200GBASE-DR4 光模块采用了 26.5625Gbaud PAM-4 调制，这样话单通道可以传输 50G，4 通道可以传输 200G。同样的，该模块也采用了 MPO 连接器与光纤互连。

200GBASE-FR4 与 400GBASE-FR8 类似，均采用了 26.5625Gbaud PAM-4 调制，唯一不同的地方就在于 200G 模块是在一根光纤中跑了 4 种不同波长的波道，而 400G 则是在一根光纤中存在 8 个波道，传输距离可以达到 2km。

同理，200GBASE-LR4 与 400GBASE-LR8 类似，也采用了 26.5625Gbaud PAM-4 调制，通过 4 种不同波长的波道在一根光纤中传输，可以达到 10km 的传输距离。

目前 200GBASE-FR4 光模块在 Google 与 Facebook 的数据中心中都有大规模部署。

不同类型光模块在 DCN 组网中的应用

现阶段大型互联网数据中心内部组网架构可以用下图来表示，自下到上依次是 Server 层、TOR 层、Leaf 层、Spine 层以及 Core 层，如下图所示。

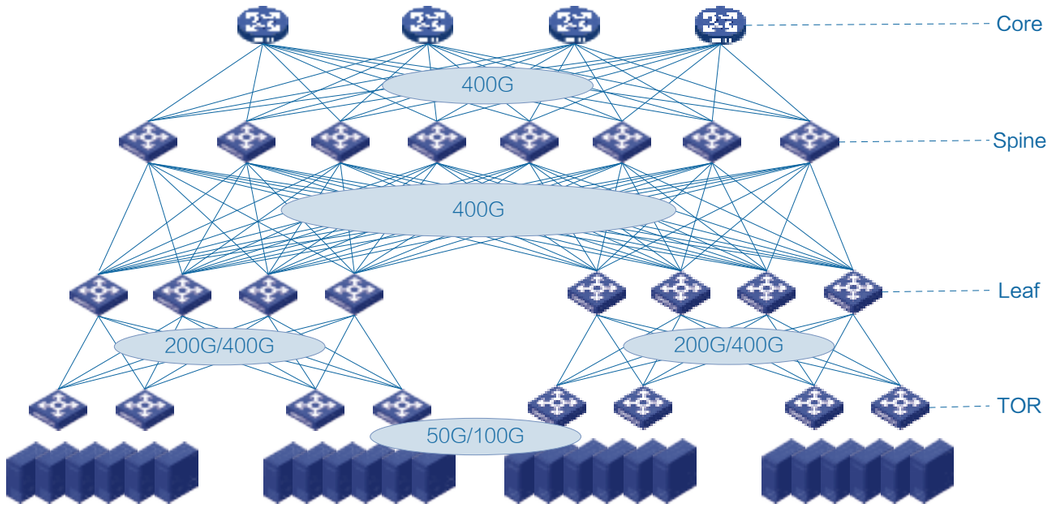


图 7 互联网数据中心组网模型

后续随着 IDC 规模的增大，互连网络的以太网速率也会随之升级，我总结了不同网络模型下各个层级的互连网络所主要使用的光模块或线缆，见图 8。

	100G 网络	200G 网络	400G 网络	800G 网络
DCI	100G/200G ACO/DCO	400G ZR/ZR+/openZRQSFPDD/CFP2/OSFP		800G ZR
Spine to Spine	100G QSFP28LR4	400G QSFPDD LR4/LR8		800G PSM4/FR4
Leaf to Spine	100G QSFP28 CWDM4	200G QSFP56 FR4	400G QSFPDD DR4/FR4	
TOR to Leaf	100G QSFP28 SR4	200G QSFP56 SR4	400G QSFPDD SR8	800G PSM8/4
Server to TOR	25G SFP+ DAC/AO CACC/AEC	QSFP28 AOC 一分二	100G DSFP/NGSFP	200G AOC

图 8 不同组网模型下光模块使用情况

针对 Server to TOR 这一层，我为大家做了进一步的展示，图 9 展示了业内一些高速率组网案例中所主要应用的光模块及线缆情况：

	SFP28(25G)	QSFP28(50G)	DSFP(100G)
相关标准	SFP28 MSA	QSFP28 MSA	DSFP MSA，18 年 9 月发布，起始是为了 5G 应用
信号速率	电信号 1*25G，光信号 1*25G	电信号 4*25G，光信号 4*25G	电信号 2*50G，光信号 2*50G(短距) 或 1*100G(中长距)
封装形式	单排金手指，850nm 激光器和光电探测器	4 通道 850nm 激光器阵列和光电探测器阵列	单排金手指，在 SFP28 金手指基础上增加了两个管脚
市场概况	25G 光模块在目前市场上已经大量出货	100G QSFP28 模块已经大量出货，被大量应用在互联网 IDC 的 100G 核心网	2*10G 的 DSFP 模块目前市场上已有发货，主要应用在无线前传；2*50G 的 DSFP 模块还在调研，开发难度不是很大
总结	随着 25G 接入成为市场主流，25G 光模块已经成为市场的中坚力量	50G Server 是一个过渡方案，在 T 用户和 A 用户的组网中，采用 100Gb 一分二 AOC 线缆用于 Server-TOR 互连	相对容易升级到 112G PAM4，在速率升级和跟 5G 共用产业链上有一定优势。与此同时，DSFP AOC 线缆已规模应用于 Z 用户与 K 用户的 Server-TOR 互连

25Gb SFP28 AOC 线缆

100GbE to 2x50GbE (QSFP28 to 2xQSFP28)

100Gb DSFP AOC 线缆

图 9 Server ⟷ TOR 互连演进

从图中不难发现，AOC 线缆是 Server to TOR 的主要互连手段，但是考虑到用电成本，国外一些大型互联网公司偏向于大规模应用 DAC 电缆。这就需要根据不同数据中心的实际环境进行多方位的考量并决策。

展望

针对互联网数据中心的组网模型发展，新华三始终放眼未来，站在行业前端，为互联网客户提供构建 IDC 组网所需的全套产品及方案。

“50G/200G/400G 高速以太网技术”写到这里就告一段落了，本文主要给大家介绍了三种以太网技术标准的发展以及相配套的光模块发展情况，主要想让大家对后续数据中心的发展有一个大概的思路。

后续，随着电路速率发展到一定程度，会遇到工艺瓶颈，届时硅光技术将逐步登上历史舞台。这将给光模块、交换机等一系列网络单元带来巨大变革。后续我们将针对硅光技术、CPO 技术等进行下一步探索，由于篇幅原因，不在本文一一叙述，也给大家自主学习网络技术提供一个大致方向。

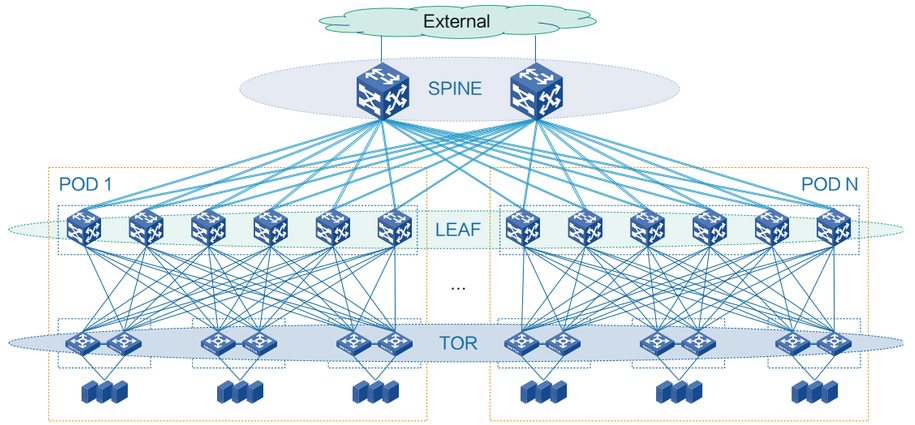
大型数据中心网络路由设计与优化

互联网系统部 杨德

前言

某些运营数据中心网络，支持超过 10 万台服务器。在本文中，这样的数据中心被称为“大型（large-scale）”，以区别于小型基础设施。这种规模的环境有一组独特的网络需求，强调操作的简单性和网络的稳定性。我们在这里，总结一下使用 BGP 作为唯一的路由协议设计和运行大型数据中心的操作经验。

通常，大型数据中心网络总体架构是这样的。



实验和大量测试表明，外部 BGP（EBGP）[RFC4271] 非常适合作为此类数据中心应用程序的独立路由协议。 这与更传统的 DC 设计形成对比，后者可能使用简单的树形拓扑，并依赖于跨多个网络设备扩展二层域，RFC7938 也提出了将 EBGP 路由协议应用于大规模数据中心的建议。

有别于 OSPF、ISIS 等链路状态协议，BGP 是一种距离矢量路由协议，因此 BGP 的扩展性更好。在中小型的数据中心组网时，使用 BGP 和使用 ISIS、OSPF 等链路状态协议性能区别不大，但是在超大型数据中心的网络中，应用 BGP 的性能会更优。OSPF、ISIS 等链路状态协议需要在网络内传递大量的 LSA，路由信息生成过程是先完成 LSA 信息同步，再计算生成路由信息。在网络部分节点发生变动或者网络割接升级时，会引起大量 LSA 的传递。而距离矢量路由协议 BGP 不存在这样的问题，BGP 节点间直接通告路由，在网络扩展和割接升级时的网络稳定性更好。

综合各种因素，我们总结了如下几个基本需求。

需求 1	通过添加更多相同类型的链接和网络设备，而不需要升级网络元素本身，选择可以“水平（scale-in）”扩展的拓扑。
需求 2	定义由众多网络设备供应商支持的一组有限的软件特性 / 协议。
需求 3	选择一种路由协议，该路由协议在编程代码复杂度和易于操作支持方面都非常简单。
需求 4	尽可能减少设备或协议问题的故障范围。
需求 5	允许进行一些流量工程设计，最好使用内置协议机制，通过明确控制路由前缀下一跳来进行。

BGP 路由协议网络设计与优化

为什么选择 BGP 作为路由协议

一般情况下，会优先选择单个路由协议以减少复杂性和相互依赖性。虽然在这种情况下通常依赖 IGP，有时在与 WAN 相连的设备上添加 EBGP 或在整个内部使用内部 BGP（IBGP），但这里建议使用仅 EBGP 设计。尽管 EBGP 是 Internet 上几乎所有域间路由所使用的协议，并且得到了供应商和服务提供商社区的广泛支持，但出于多种原因，它通常未被部署为数据中心内的主要路由协议。

为什么要选 EBGP 呢？最初人们认为它是这样的。

- BGP 被认为是“仅 WAN，仅协议”，并不经常用于企业或数据中心应用。
- 与 IGP 相比，BGP 被认为具有“慢得多”的路由收敛。
- 大规模 BGP 部署通常利用 IGP 进行 BGP 下一跳解析，因为 IBGP 拓扑中的所有节点均未直接连接。
- BGP 被认为需要大量的配置开销，并且不支持邻居自动发现。

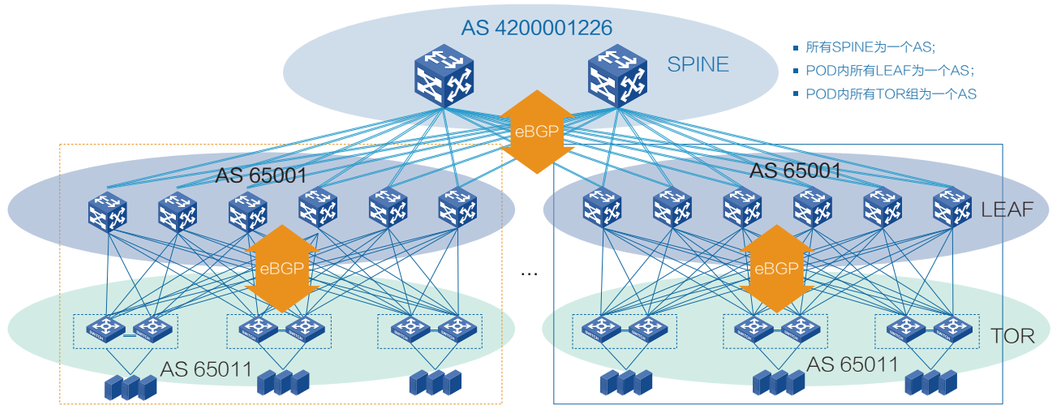
但是，使用 BGP 协议进行设计，同样存在如下优势。

- BGP 在其协议设计的一部分中具有较少的复杂性与大多数链路状态 IGP 相比，内部数据结构和状态机更简单。例如，BGP 没有实现邻接关系形成，邻接关系维护和 / 或流控制，而是仅依靠 TCP 作为基础传输。 这满足了需求 2 和需求 3。
- 与链路状态 IGP 相比，BGP 信息泛滥开销更小。这样，BGP 更好地满足了需求 3 和需求 4。 还值得一提的是，所有广泛部署的链路状态 IGP 均具有定期刷新路由信息的特性，而 BGP 不会使路由状态失效。
- BGP 支持第三方（递归解析）下一跳。通过与应用程序“控制器”建立对等会话，可以将多路径控制为基于 ECMP 或基于应用程序定义的路径的转发，该对等会话可以将路由信息注入系统，从而满足需求 5。
- 使用定义明确的自治系统编号（ASN）分配方案和标准的 AS_PATH 循环检测，可以控制“ BGP 路径搜寻”，而复杂的不需要的路径将被忽略。
- 使用定义明确的自治系统编号（ASN）分配方案和标准的 AS_PATH 循环检测，可以控制“ BGP 路径搜寻”，而复杂的不需要的路径将被忽略。
- 使用最少的路由策略实现的 EBGP 配置更容易对网络可达性问题进行故障排除。因此，BGP 满足需求 3。

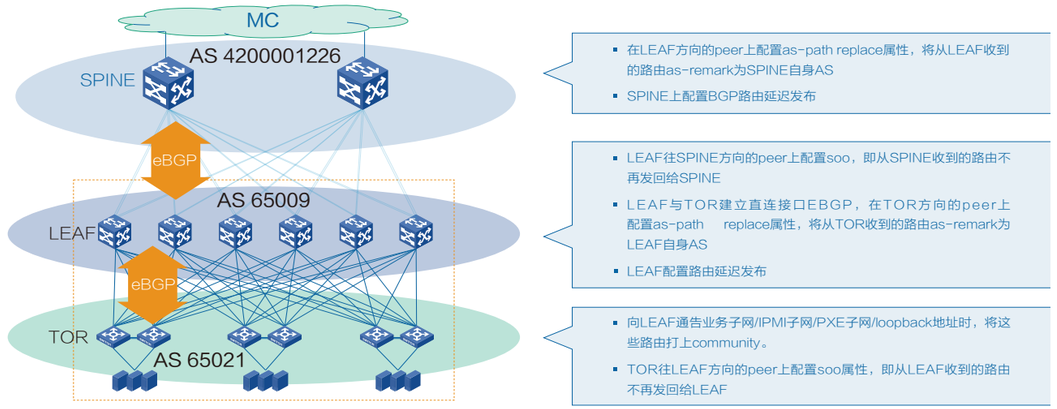
Underlay 路由规划的三种方案

方案一，按 SPINE/LEAF/TOR 角色进行 AS 划分

方案一（拓扑）



方案一（BGP 配置）

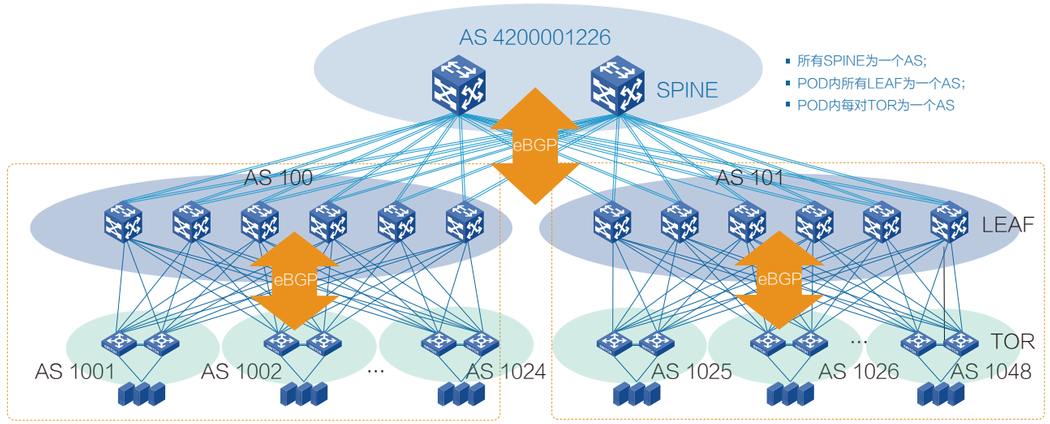


这种组网情况下，SPINE/LEAF/TOR 每层只与其他层级建立 BGP 邻居，相同层级内部相同角色的设备间可以不建立邻居，减少了配置工作量和产生的路由表项数量，可以从全局角度更好地把握整体网络的规划，降低网络管理运维的复杂度。

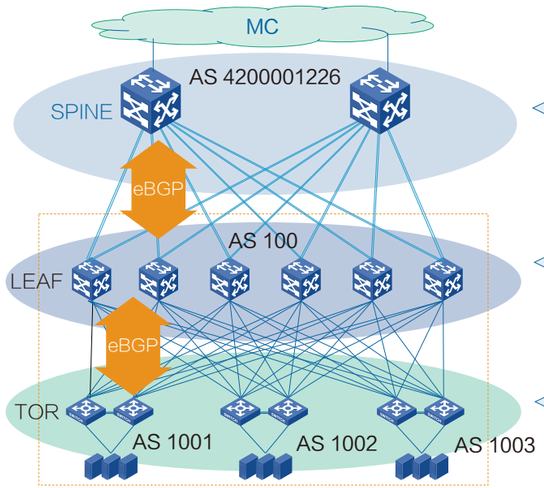
这种组网情况比较适用于 TOR 下联服务器功能性比较统一，服务器集群主要体现大规模计算能力和分布式计算能力，对 AS 外部整体表现一致性服务的特点。

方案二，SPINE 核心单独划分 AS，POD 内全部 LEAF 单独划分 AS，TOR 成对划分 AS

方案二（拓扑）



方案二（BGP 配置）



- 在LEAF方向的peer上配置as-path replace属性，将从LEAF收到的路由as-remark为SPINE自身AS
- SPINE上配置BGP路由延迟发布

- LEAF往SPINE方向的peer上配置soo，即从SPINE收到的路由不再发回给SPINE
- LEAF配置路由延迟发布

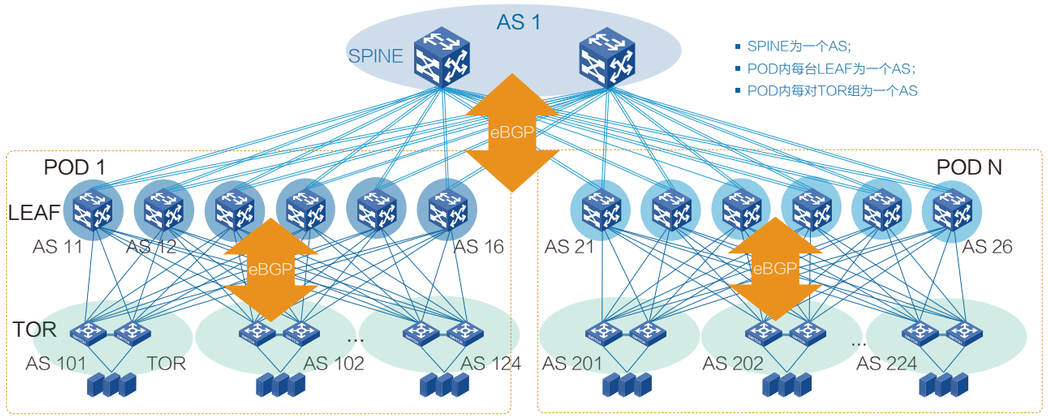
- 向LEAF通告业务子网/IPMI子网/PXE子网/loopback地址时，将这些路由打上community 。

此类组网情况下，SPINE 只与 LEAF 层建立简单的 BGP 邻居，SPINE/LEAF 相同层级内部相同角色的设备间可以不建立邻居，TOR 层面需要每组均与 LEAF 层建立邻居，并在 LEAF 层面汇总并同步所有 TOR 层的路由表项，统一协调跨三层流量的寻路、转发。

这种组网模型比较适用于每 TOR 下的服务器集群功能独立性较强，但在转发层面具有简单的路由可达一致性特点，无须建立特定 TOR 区域之间的独有连接，即无需形成个性化的独有转发通道需求，虽然 LEAF 层表项数量有所增加，但管理运维复杂度也不算高。

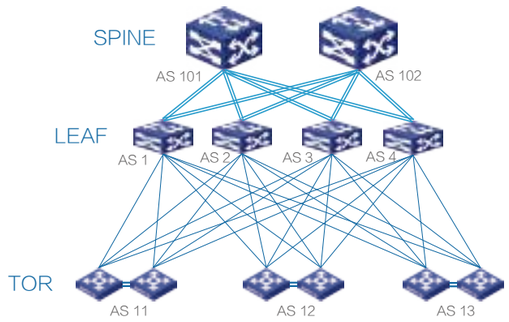
方案三，SPINE 核心单独划分 AS，每 POD 内每 LEAF 单独划分 AS，TOR 成对划分 AS

方案三（拓扑）



方案三（BGP 配置）

方案三需要用到的重要 BGP 属性



主要属性

- balance as-path-relax：用来配置内容不同但长度相同的 AS_PATH 属性的路由能够形成 BGP 负载分担
- balance as-path-neglect：用来配置不同 AS_PATH 属性的路由能够形成 BGP 负载分担

配置进行 BGP 负载分担的路由条数	balance [ebgp | eibgp | ibgp] number	缺省情况下，不会进行 BGP 负载分担
配置不同 AS_PATH 属性的路由能够形成 BGP 负载分担	balance as-path-neglect	缺省情况下，不同 AS_PATH 属性的路由之间不能形成 BGP 负载分担
配置内容不同但长度相同的 AS_PATH 属性的路由能够形成 BGP 负载分担	balance as-path-relax	缺省情况下，内容不同但长度相同的 AS_PATH 属性的路由不能形成 BGP 负载分担

此类组网情况下，SPINE 需要与每 LEAF 层设备均建立相应的 BGP 邻居，每 LEAF 层设备需要与每组 TOR 建立 BGP 邻居，且相同层级内部相同角色的设备间也需要建立邻居，TOR 层面需要每组设备均与 LEAF 层每个节点建立邻居，且相同层级内部相同角色的设备间也需要建立邻居。LEAF 层面需要汇总并同步所有 TOR 层的路由表项。

这种组网模型同样比较适用于每 TOR 下的服务器集群功能独立性较强，但在转发层面具有复杂的跨三层互联需求，一般可能会在应用层面上频繁建立、释放特定 TOR 区域之间的连接通道。此种场景中，对 LEAF 层设备的表项要求较高，建立 / 释放路由表项会较频繁，设备负载较高，整体管理运维复杂度也比较高。

SR Policy 技术及实践

互联网系统部 吴银怀

数字经济的高速发展，使得承载数字经济的互联网基础设施建设如火如荼，迅速成为热点话题。作为数字经济基础设施的重要数字管道，互联网数据中心园区及城域网 / 广域网在数字经济时代和 SDN 软件定义网络的理念一同被推上了风口浪尖。

互联网数据中心广域 / 城域网覆盖范围广、接入用户多、承载业务复杂，近年人工智能、短视频、公有云、大数据等应用的爆发，进一步推动数据中心广域网 / 城域网在数据流量、网络规模、网络带宽、服务质量等方面发展。新兴业务爆发与广域 / 城域网带宽、费用、可靠性、健壮性等之间矛盾，使互联网数据中心广域网 / 城域网流量调度成为必然，在 SDN 软件定义网络集中控制理念的加持下，基于段路由 Segment Routing 的统一流量调度方案相比传统 MPLS TE 调度方案优势明显，并得到广泛关注和部署。但初始 SR 技术在隧道接口体系下，依然存在很多短板，流量调度方案亟需被拯救：

SR-TE 隧道接口体系

SR-TE 最初使用隧道接口的概念以支持简单 SR-TE 功能从而满足大多数用户的需求，引流方式沿用用户比较习惯的 RSVP-TE 方式，导致 SR-TE 在性能、扩展性、标签支持、负载均能、新功能等方面存在短板：

- 隧道接口和引流动作割裂，引流方式麻烦且设备性能压力较大，无法大规模部署；
- 隧道需预先配置，隧道终点不明确的情况下，需要部署全网状隧道，可扩展性很低；
- 隧道接口和 RSVP-TE 电路算法导致只能使用 Adj-SID 路径，无法使用 Prefix-SID 路径，导致 IP ECMP 能力无法施展拳脚，并且 Segment 列表长度过长导致部分设备无法支持；
- 隧道与路径强相关，为一对一关系，导致 E/UCMP 必须配置多个不同隧道接口，配置繁琐且扩展性差；
- 隧道接口占用设备的逻辑资源，设备能支持的 SR-TE 数量有限；
- 不支持新的 SR 功能如 Flex-Algo、Performance Measurement 等。
- SR Policy 是全新设计的一套 SR-TE 体系架构，完全不同于传统的基于隧道接口的实现方式。基于 SR Policy 的 SR-TE 已得到业界的广泛接受，在数据中心流量调度中得到广泛的应用。

SR Policy 技术解析

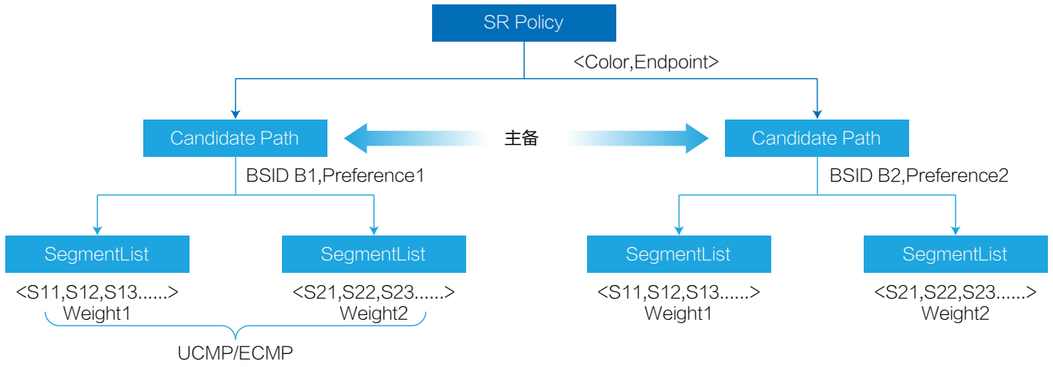
SR-TE Policy 提供了灵活的转发路径选择方法，满足用户不同的转发需求。当 Segment Routing 网络的源节点和目的节点之间存在多条路径时，合理利用 SR Policy 选择转发路径，不仅可以方便管理员对网络进行管理和规划，还可以有效地减轻网络设备的转发压力。SR Policy 完全抛弃了隧道接口的概念，SR Policy 通过解决方案 Segment 列表实现流量工程意图。即本质上 SR Policy 不是隧道接口而是 Segment 列表，Segment 列表对数据包在网络中的任意转发路径进行编码，列表中的 Segment 可以支持 IGP Segment、IGP Flex-Algo Segment、BGP Segment 等。

SR Policy 以二元组（头端给定）或三元组标识，即头端 BSID、颜色 Color、端点 End-point，具体解析请自行查阅资料。

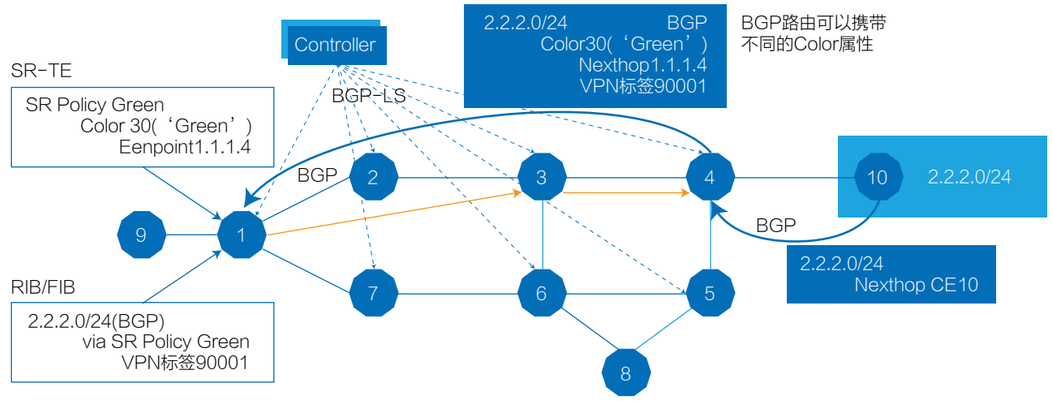
相比 RSVP-TE 的隧道接口的引流方式，SR Policy 不仅能实现自动引流，基于前缀或前缀流分类进行引流，简化多路径 ECMP，还可支持针对 IP 优化的 SR 原生算法，最大化利用 ECMP 和最小化 Segment 列表长度，可支持 Flex Algo、基于链路和 SR Policy 的性能测量（时延、丢包、存活等）。

SR Policy 引流方案的核心路由协议为 BGP，对业务路由进行染色实现自动生成 SR Policy 和自动引流至 SR Policy，可实现按需下一跳、自动引流等。SR Policy 具有至少一条候选路径，其最高 Preference 值的有效候选路径为活动候选路径。Preference 为候选路径的优先级，用于在通过 SR-TE Policy 转发流量时，选择候选路径。

候选路径可由操作员或控制器计算，也可以由应用指定。每条候选路径可以通过不同的方式学到，例如本地配置、NETCONF、PCEP 或者 BGP。SR Policy 的活动路径根据候选路径的有效性和偏好值来选择，候选路径的来源不影响选择过程。每条候选路径具有至少一个 SID 列表，SID 列表中的 SID 为转发路径上各个节点到下一跳的 SID。候选路径由一个 SID 列表或者多个带权重的 SID 列表组成。每个 SID 列表（SegmentList）具有关联的负载均衡权重，流量在 SID 列表间根据权重进行负载均衡：



SR Policy 根据候选路径的有效性和偏好值来选择活动路径，候选路径具有 BSID（Binding-SID）属性，BSID 为入节点的 SID，会被弹出压入 SegmentList。头节点将 BSID 绑定到 SR Policy，并将 SR Policy 作为 BSID 标签的 MPLS 重写条目安装在转发平面中。即，入向标签：BSID；标签操作：弹出 BSID，压入 SegmentList<S1，S2>；出口：出接口和下一跳。那么如何通过 SR-TE Policy 引流：



节点 4 将 BGP 路由 2.2.2.0/24 着色为“绿色”（对应的颜色值为 30 ），当头端节点 1 收到此业务路由后，发现此路由与本地配置的 SR Policy GREEN（“绿色”，节点 4) 匹配（假设此 SR Policy 表示从节点 1 到节点 4 的低时延路径，图中黄色路径），因此在本地转发表中把此 BGP 前缀的下一跳设置为 SR Policy GREEN 的 BSID，则去往 2.2.2.0/24 的流量会被自动引导至低延迟 SR Policy GREEN 。

总结：出口 PE 节点通告 BGP 业务路由或者入口 PE 接收路由时，对路由进行着色， 当头端节点接收到已着色业务路由，对比 BGP Color 团体属性和下一跳与 SR Policy 的颜色和端点，如果相同 BGP 安装此路由，将其解析到 SR Policy 的 BSID。因此 SR-TE Policy 解决方案的核心为 BGP 路由，关键在于新增的 Color 属性，通过对业务路由进行着色实现 Policy 和自动引流至 SR Policy。

H3C Comware v9 网络操作系统 SR Policy 配置：

- segment-routing // 进入 Segment Routing 视图
- traffic-engineering // 创建 SR-TE
- policy policy-name // 进入 SR-TE Policy 视图
- binding-sid mpls mpls-label // 手工配置 BSID
- color color-value end-point ipv4 ipv4-address // 配置 Color 和 End-point

H3C Comware v9 网络操作系统候选路径配置和引用 SID List：

- candidate-paths //SR-TE Policy 视图下，创建 SR-TE Policy 候选路径并进入
- preference preferenc-value // 配置候选路径优先级并进入，默认未配置 SR Policy 候选路径优先级。不同的优先级代表不同的候选路径。
- explicit segment-list segment-list-name [weight weight-value] // 为指定优先级的 SR Policy 候选路径配置显式 SID List（前提是要创建 SID List）。

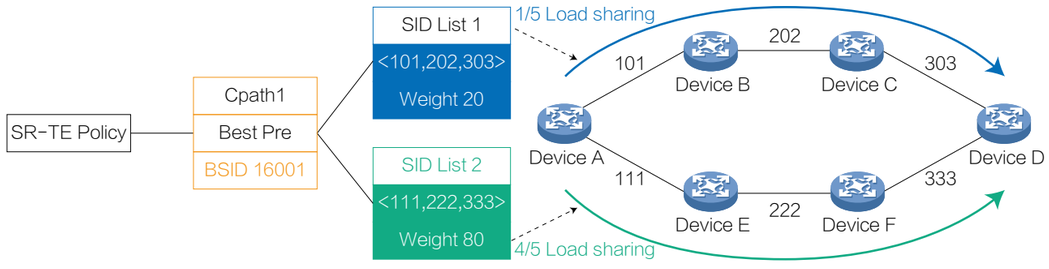
SID List 配置：

- segment-routing // 进入 Segment Routing 视图
- traffic-engineering // 创 v 建 SR-TE，并进入
- segment-list segment-list-name // 创 建 SID 列表，并进入
- index index-number mpls label label-value // 添加节点

设备是如何通过 SR Policy 选路和转发流量的呢?

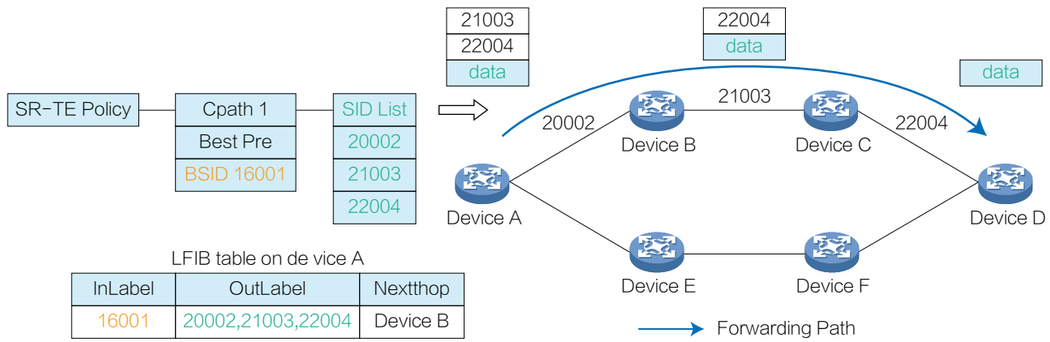
设备收到栈顶标签为 BSID 的报文后，根据 BSID 选择对应的有效 SR Policy 转发流量。BSID 匹配到多个有效 SR Policy，优先选择先配置 BSID 的 SR Policy，并优先选择 SR Policy 中优先级最高的有效候选路径转发流量。优先级最高的有效候选路径各个 SID 列表间基于权重对转发的流量进行负载。

SR Policy 转发选路过程示意：



根据 BSID 选择有效的 SR Policy 转发流量，再选取优先级取值最大的有效候选路径 Best Pre 转发流量。该候选路径中有两个有效的 SID 列表：SID List 1 和 SID List 2，其权重分别为 20 和 80。通过该 SR Policy 转发流量时，SID List 1 和 SID List 2 转发的流量占比分别为 20% 和 80%。

SRPolicy 流量转发过程示意：



LFIB table on de vice A

InLabel	OutLabel	Nextthop
16001	20002,21003,22004	Device B

Device A 收到栈顶标签为 16001 的报文后，查找 LFIB（Label Forwarding Information Base,标签转发信息库）表判断该标签为 BSID，需要根据 SR Policy 转发该报文。Device A 根据 LFIB 表获取出标签和下一跳（Device B），将 BSID 弹出并压入 SID 列表的标签栈 {20002, 21003, 22004}，然后将报文发送给 Device B。其中，20002 为 Device A 到达 Device B 的 SID；21003 为 Device B 到达 Device C 的 SID；22004 为 Device C 到达 Device D 的 SID。Device B 收到报文后查找 LFIB 表，根据入标签获取到下一跳为 Device C，将报文发送给 Device C。Device C 收到报文后查找 LFIB 表，根据入标签获取到下一跳为 Device D，将报文发送给 Device D。Device D 收到报文后，如果报文携带标签，则查找 LFIB 表转发报文；如果报文未携带标签，则查找 IP 转发表转发报文。

MP-BGP 定义了 IPv4 SR Policy 地址族，新增 SR Policy NLRI（Network Layer Reachability Information，网络层可达性信息），即 SR-TE Policy 路由（包含 SR-TE Policy 的相关配置，包括 BSID、color、Endpoint、Preference 和 Weight 等）。将 SR Policy 路由发布给对等体后，对等体也可以根据 SR Policy 转发流量。

设备为 BGP 路由添加 Color 扩展团体属性，之后如果设备收到匹配该路由的报文，则会 Color 属性中的 Color 值查找具有相同 Color 值的 SR Policy 转发报文。否则通过最优路由转发报文。

本质上 SR Policy 是作为路由引入 BGP 路由指导转发，配置 BGP 发布和引入 SR Policy 路由：

- router id router-id // 配置全局 Router ID。
- bgp as-number [instance instance-name] // 启动 BGP 实例并进入 BGP 实例
- peer { group-name | ipv4-address [mask-length] } as-number as-number // 配置对等体。
- address-family ipv4 sr-policy // 创建 BGP IPv4 SR-TE Policy 地址族并进入。
- peer { group-name | ipv4-address [mask-length] } enable // 本地路由器与指定对等体 / 对等体组交换 BGP SR Policy 路由。默认本地路由器不能与对等体或对等体组交换 BGP SR Policy 路由。
- import-route sr-policy // 将 SR Policy 路由引入 BGP 路由。

还可以通过配置控制 BGP SR Policy 路由的优选和发布（BGP IPv4 SR Policy 地址族视图下）：

- peer { group-name | ipv4-address [mask-length] } next-hop-local // 配置向对等体 / 对等体组发布路由时，将下一跳属性修改为自身的地址。
- peer { group-name | ipv4-address [mask-length] } allow-as-loop [number] // 配置对于从对等体 / 对等体组接收的 BGP 消息，允许本地 AS 号在该消息的 AS_PATH 属性中出现及出现的次数。
- peer { group-name | ipv4-address [mask-length] } preferred-value value // 为从对等体 / 对等体组接收的路由分配首选值，默认为 0.
- peer { group-name | ipv4-address [mask-length] } route-limit prefix-number [{ alert-only | discard | reconnect reconnect-time } | percentage-value] * // 配置允许从对等体 / 对等体组接收的路由的最大数量。默认不限制从对等体 / 对等体组接收的路由数量。
- peer { group-name | ipv4-address [mask-length] } reflect-client // 配置本机作为路由反射器，对等体 / 对等体组作为路由反射器的客户机。默认不配置。
- peer { group-name | ipv4-address [mask-length] } prefix-list ipv4-prefix-list-name { export | import }// 为对等体 / 对等体组设置基于 IPv4 地址前缀列表的路由发布和接收过滤策略。默认不对发布和接收的路由信息进行过滤。
- peer { group-name | ipv4-address [mask-length] } route-policy route-policy-name { export | import }// 对来自对等体 / 对等体组的路由或发布给对等体 / 对等体组的路由应用路由策略。默认不对等体 / 对等体组指定路由策略。
- peer { group-name | ipv4-address [mask-length] } advertise-community // 向对等体 / 对等体组发布团体属性。默认不向对等体 / 对等体组发布团体属性。
- peer { group-name | ipv4-address [mask-length] } advertise-ext-community // 配置向对等体 / 对等体组发布扩展团体属性。默认不向对等体 / 对等体组发布扩展团体属性。
- 其他配置如维护 BGP 会话、统计转发流量、显示和维护等为常用熟知功能，均不一一展开。

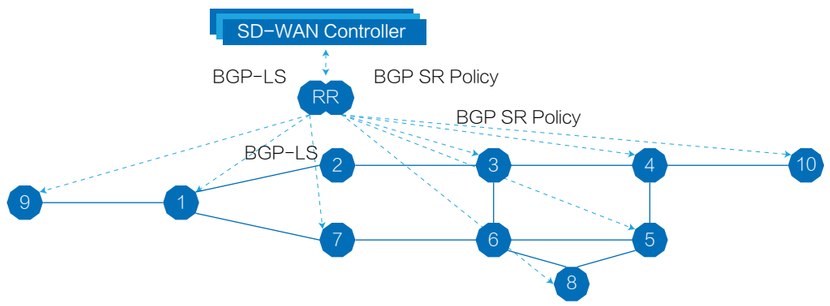
传统的 IGP 协议通过 cost 值等计算最短路径，数据按最短路径传输。若数据都走最短路径显然会对路径造成极大的性能压力，那么可以通过一些约束条件使报文走非最短路的其它路径。而 SR 中，每个 IGP Prefix-SID 除了和 Prefix 关联外，还与路径算法关联，比如基于 IGP Cost 值的 SFP 算法或基于 IGP Cost 值的严格 SFP 算法，IGP Prefix-SID 格式定义中包含了对于算法的定义，Prefix-SID 格式如下：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
类型								长度								标志位								算法							
SID/索引/标签（可变）																															

对于算法的定义共计 255 位，算法 0~ 算法 127 是 IETF 定义的算法，包括基于 IGP Cost 值的 SFP 算法或基于 IGP Cost 值的严格 SFP 算法，分别称为算法 0 和算法 1，默认参与算法 0 计算。算法 128~ 算法 255 由操作员定义，称为灵活算法（Flex-Algo）。

Flex-Algo 计算规则用三元组表示，即 Metric Type，Calc-Type，topo-constraints），分别表示链路指标约束，计算算法约束，拓扑约束。所以 Flex-Algo 可以用单个 Segment 提供带有约束条件的动态计算路径，降低节点压入标签深度的要求，并且其 TI-LFA 路径也遵循相同的约束条件和优化目标；同时沿途中间节点失效不会影响 Flex-Algo Prefix-SID 的可用性。

在流量调度方案中，控制器的角色也必不可少，控制器 Controller 为调度系统的大脑：



可以通过 Controller 部署业务，基于约束条件计算路径；通过 Netconf 实现整网自动化配置的下发和运维信息的数据获取，比如全网节点 / 邻接标签分配；利用 gRPC 进行速率和丢包监控，比如接口出入速率、每端口每队列的转发信息、丢包；通过 SNMP 获取 SR-TE 隧道数据统计；利用 Syslog 实现设备业务事件的信息监控。Controller 和 RR 之间部署 BGP LS 进行拓扑采集，通告隧道监控状态；Controller 和 RR 之间、RR 和引流首节点之间部署 SR Policy 地址族下发 Policy、部署 BGP-LS 地址族上报 Policy 状态。

SR Policy 体系适用于 SRv6，在 SRv6 完全基于 IP 的框架下，SR Policy 这个完全基于 IP 优化的体系更能发挥其所长。H3C SR Policy 在国内互联网行业巨头客户已经落地部署，如某大型云服务商使用 H3C S12500R 云计算核心路由器搭建了多个 DC 园区之间的 MAN 网络，SR Policy 使其摆脱了 SR-TE 隧道接口体系的束缚，实现在 DC 园区骨干网络设备中部署 SR-TE 策略的功能，用户流量按指定 SLA 精细化调度到不同 SR-TE 路径，进行多级哈希，提高了数据中心园区骨干网的带宽利用率，增强了数据中心园区骨干网络的可靠性、容错率及健壮性，获取更高的投资回报比，并且帮助其在网络精细化运营的道路上迈进了坚实的一大步。

SRv6 技术及应用场景

互联网系统部 于伦

产生背景

随着互联网业务的快速发展，单 DC 已经无法满足需求，业务云化和分布式 IDC 建设，以及 DC 间互访流量的爆发式增长，使得 DCI 逐渐成为网络建设的核心；DCI 骨干网也是互联网公司发展到一定阶段的必然选择，当前互联网行业头部玩家的 DCI 骨干网正在由 100G 逐步迈向 400G，DCI 骨干网也已经向多平面的方向演进。DCI 流量呈现类型多、突发性强和不均衡的特点，为了更经济有效利用 DCI 带宽资源，差异化和精细化调度是 DCI 的关键需求。

智能调度是新一代广域网的一个关键能力，对应用质量的保障、带宽资源的优化非常重要。现有的 MPLS 及 RSVP-TE 等流量工程技术虽然可以满足应用对带宽的差异化保障需求，但存在协议种类多、部署复杂、管理困难、可扩展性差等问题，无法满足新一代广域网所要求的动态部署、灵活调度、快速、可扩展等方面的要求。

在此背景下，SRv6 技术作为重构广域网的核心技术，通过自动部署、集中控制、智能调度及可视化等手段，加速网络交付，优化应用体验，提高带宽利用率，简化网络运维，满足云计算对广域网的最新需求。越来越受到互联网客户的关注。

SR 技术介绍

SegmentRouting（后续简称 SR）是一种源路由协议，也称为分段路由协议，由源节点来为应用报文指定路径，并将路径转换成一个有序的 Segment 列表封装到报文头中，路径的中间节点只需要根据报文头中指定的路径进行转发，除源节点外，其它节点无需维护路径状态。

SegmentRouting 有如下四个优点：

• 简化了控制协议

它只采用 IGP，统一了控制协议，不再像 MPLS 那样在 IGP 的基础还要 LDP、RSVP-TE 等协议，降低了运维的复杂度。SR 只需要设备通过 IGP 路由协议对 SR 的扩展来实现标签分发和同步，或者由控制器统一负责 SR 标签的分配，并下发和同步给设备 SR 技术的控制平面如下图所示：

图 4 SR 技术逻辑图

良好的扩展性

以前实现路径编程 (流量工程，TE) 时一般采用 RSVP-TE，网络中的每个节点都要感知到每条路径的状态，协议的消耗很大，限制了 TE 隧道的规格，难以部署和维护。SegmentRouting 路径编程则是在头结点进行，海量的路径都是依赖于有限的表示链路和节点的 Segment 的组合，网络中间节点几乎不感知路径状态，具备很高的扩展性。

SR 可以复用已有的 MPLS 和 IPv6 转发平面，网络设备不做改动或者进行小的修改就可以支持对 SegmentRouting 的转发，如：在 MPLS 网络中，Segment 就是 MPLS 标签，路径就是标签栈；在 IPv6 网络中，Segment 就是 IPv6Address，路径就是封装在一个 routingextensionheader 中的 IPv6Address 列表。

SegmentRouting 技术特点示意图

可编程性好

SegmentRouting 中的 Segment 非常类似于计算机的指令，通过对 Segment 的编排可以实现类似于计算机指令的功能。具备非常好的灵活性，可以非常灵活地建立满足不同需求的路径，释放网络的价值。

更可靠的保护

SegmentRouting 能提供 100% 网络覆盖的快速重路由（FastReRoute）保护，解决了 IP 网络长期面临的技术难题，能够在高可扩展性的前提下，又可以达到完全的可靠性保护。

SRv6 技术介绍

SR 技术可以直接使用 IPv6 的数据平面，基于 IPv6 路由扩展头进行扩展，即 SRv6 技术。将 IPv6 地址作为 SID 对报文进行转发。SRv6 仅针对 IPv6 路由扩展头进行处理，对报文其他部分没有任何影响，因此使得 SR 与现有 IPv6 网络完美兼容，实现极简的转发模式。

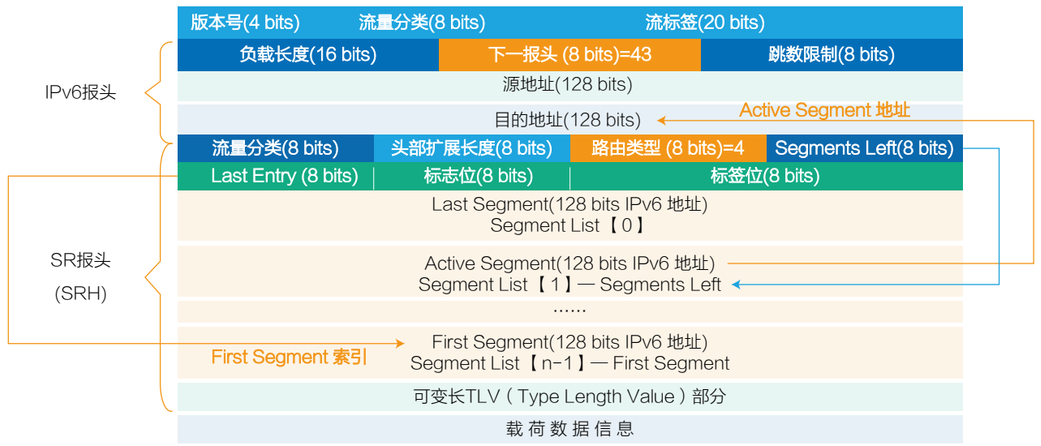
SRv6 不仅继承了 SR 的优点，还具备标签空间数量无限、全网唯一、任意点可达的优点（IPv6 地址特点）。进而可以实现只要地址可达，可以任意点接入，任意点之间互联。SRv6 具有的独特优势，使其成为下一代 DCI 网络的核心技术，成为业界研究的热点。随着全球各大主流互联网公司、运营商对 SD-WAN 及 SR 技术的探索实践及商业部署，并进行持续完善和功能提升，SRv6 相关技术及解决方案也日趋成熟。

SRv6 的数据平面

如下图 SRv6 报文头部结构中的 IPv6 部分，“下一报头字段”（图中标红部分）表示报文后面封装的协议类型，常用协议字段：（1）IPv4—[4]；（2）TCP—[6]；（3）UDP—[17]；（4）Routing—[43]；

SRv6 报文的“下一报头字段”固定为 43 号协议 Routing 使用了 IPv6 的头部扩展字段。在 RFC2460 标准中对 IPv6 的 43 号协议 Routing 的 IPv6 的扩展字段进行了定义：

- 扩展流量分类（NextHeader）：标识紧跟在 SRH 后封装的报文头类型，如 TCP、UDP 等；
- 头部扩展长度（NdrExtLen）：SRH 头所占用长度，且任意 IPv6 设备均可跳过该头部；
- 路由类型（RoutingType）：可定义多种路由头部类型，SRH 类型为 4；
- 段指针（SegmentsLeft）：到达目的节点前仍然应访问的中间节点数，若其为 0 则忽略扩展头部。



SRv6 报文头部结构图

标签采用反向编码的方式，最新的标签编码为 0，即 Segmentlist [0]。第一个标签则是 Segmentlist[n-1]，如图中 FirstSegment，即假设当前 SRv6 标签数量为 4，则最开始执行的标签序号为 3，最后一个执行的标签序号为 0。SRv6 组网中当前网络节点执行的标签编号为 SegmentLeft 指针。设备通过读取 SegmentLeft 指针得知当前应读取的 SegmentList 值，并将对应的标签地址复制到 IPv6 目的地址中再由设备进行转发。SRH 报头中的可变长 TLV 字段可添加一些非规则的附加信息，如安全相关的 HMAC 等。

SRv6 转发流程

通过这样只增加扩展头的方式，与原有 IPv6 转发平面平滑融合。如下图中从 R1 到 R6 的报文规划经过橙色线所指示的路径，如下图在 IPv6 转发平面下处理流程如下：

- 首节点（R1）：根据路径计算结果，确定 SRv6 的报文路径为 R2，R4，R6 并封装在 SRH 报文头，用 IPv6 地址栈的方式指明中间需经过的路径为 R2、R4、R6，同时初始化 SegmentsLeft 指针为 2，并将 SegmentsLeft 指针对应的 SegmentList【2】地址（R2）复制到 R1 外层 IPv6 目的地址，并查路由表转发到 R2。
- SR 路径涉及转发节点（R2,R4）：如果发现 IPv6 头中目的地是本节点，则查看 SRH 头，如果仍有下一个待处理 Segment，则将外层 IPv6 头中目的地址改为 SegmentsLeft 指针对应的 SegmentList【1 或 0】地址，并查路由表转发。此外当 SegmentList=0 时，则设备忽略 SRH 扩展头部内容，如在下图中 R4 网络节点的 SRH 标灰部分。
- 非路径涉及转发节点（R5）：直接根据 IPv6 报文头中的目的地址进行 IPv6 报文进行查表转发，不处理 SRH 扩展报头内容。
- 尾节点（R6）：去掉 SRH 头，按照普通 IPv6 报文进行转发。

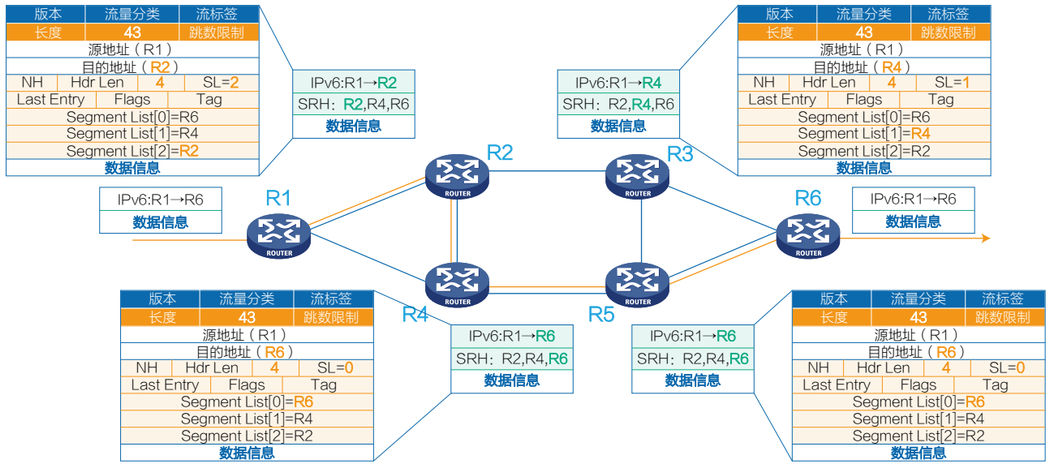


图 11SRv6 转发流程图

从上述流程可以看出，对于非 SR 路径涉及的转发节点，完全无需支持 SegmentRouting，仅支持标准的 IPv6 转发即可。采用这种方式，对网内不支持 SegmentRouting 的设备可以进行无缝兼容。

SRv6Segment 格式

SRv6 标签 SID 通常采用一个 128bits 的 IPv6 地址格式，编写规则也遵循 IPv6 地址规则。SID 一共分为两部分：Locator 和 Function，如下图 SRv6Segment 格式示意图：

- Locator：路由定位功能，占用较多比特位用于进行标签路由从而找到对应设备；
- Function：占用较少比特位用于定义所属设备收到该标签后的操作；
- Argument（可选）：占用尾部部分用于本地 Function 参数，如定义报文流和服务等信息等。

图 12SRv6Segment 格式示意图

SID 需在开启 SRv6 功能的设备上进行配置，设备可以灵活的进行标签长度的分配，并且标签的操作仅本地有效。

SRv6 技术亮点

• SRv6 技术兼容性更强

使能 SRv6 BE 功能，中间节点仅需支持 IPv6 转发，中间设备可以不支持 SRH 识别能力，只要路径上标签转换点支持 SRv6 即可，路径上其他设备只要支持 IPv6 即可；

• SRv6 路径可以按需制定路径

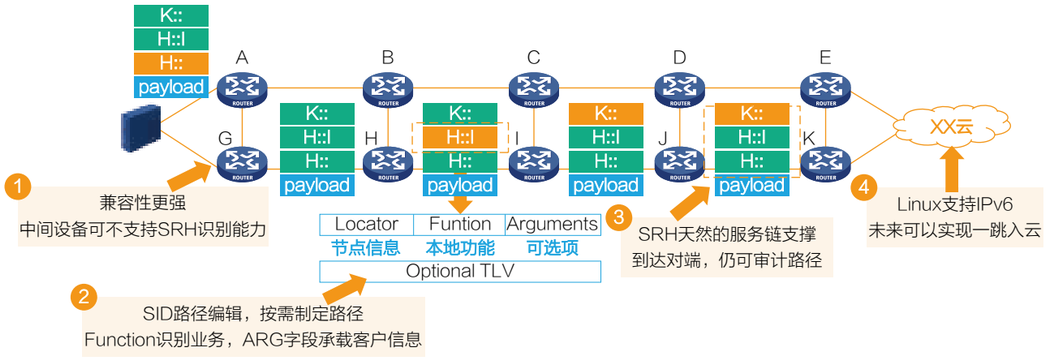
实际应用中，SID 中本地 Function 字段用来做业务识别，Arguments 字段承载客户信息，可以根据 Function 和 Arguments 字段执行相关操作。

• SRH 天然服务链支撑

针对路径中可能串行的防火墙不支持 SRH 识别的使用场景，报文在经过路由器时，路由器将报文的 SRH 字段拆去，引流到旁挂的防火墙处理，防火墙处理完的报文再回到路由器，路由器再将相关的 SRH 字段添加上，由此实现，报文到达对端之后，路径依然可以审计。

• 一跳入云

Linux 自 4.10 版本后支持 SRv6，并借助 FD.IO 或优化 DPDK 来完善 SRv6 功能。基于 SRv6 在 Overlay 网关上利用 FD.IO 及 Underlay 的 SLA 信息生成 SRH，Underlay 只需要执行简单 END 操作，不需要作为基于流的 SR-TE 头端，无须对 Underlay 进行大面积改造；同时 Overlay 和 Underlay 交互的是 SLA 路径信息，边界清晰。最重要的是 Overlay 的 SLA 交给应用来负责，中间链路设备只需支持 IPv6 即可，实现上云路径端到端跨域免配置、业务分钟级开通，让业务应用入网即入云，从而实现一跳入云。



SRv6 Policy

传统的流量工程都是一维的：或者是基于目的地，或者是基于业务等级。随着云业务的发展，基于流的流量工程越来越成为必备能力（例如为同一租户的不同应用提供不同的SLA，又或者是实现 Overlay 和 Underlay 交接时）。与传统的一维流量工程不同，基于流的流量工程粒度划分是二维的，即目的地 + 业务等级。

SRv6Policy 设置转发策略时的三要素包括：SRv6Policy 颜色、SRv6Policy 端点和和转发等级。其中颜色和端点标识了 SRv6Policy，转发等级则是 SRv6Policy 的一个重要属性。两台设备之间可以建立多组 SRv6Policy，每一组 SRv6Policy 对应着一组业务目的地网段，不同组的 SRv6Policy 可以采用相同的端点（不需要额外的 loopback 地址），只需要为不同目的地设置不同的颜色即可；为同一组中的多条（子）SRv6Policy（端点相同但颜色不同）赋予不同的转发等级，一条 SRv6Policy 对应着一个业务等级。所以 SRv6Policy 体系下是采用不同颜色 + 不同（子）SRv6Policy 的方式实现基于流的流量工程。

SRv6Policy（SegmentRoutingPolicy，段路由策略）提供了灵活的转发路径选择方法，满足用户不同的转发需求。当 SegmentRouting 网络的源节点和目的节点之间存在多条路径时，合理利用 SRv6Policy 选择转发路径，不仅可以方便管理员对网络进行管理和规划，还可以有效地减轻网络设备的转发压力。

SRv6 VPN

传统 VPN 网络中通过部署 LDP/RSVP-TE 等标签分发协议，在公网中建立虚拟专用通信网络。这种方式部署复杂，维护成本较高。通过在公网部署 SRv6 VPN 可以解决上述问题。

SRv6 VPN 是通过 SRv6 隧道承载 IPv6 网络中的 VPN 业务的技术，控制平面采用 MP-BGP 通告 VPN 路由信息，数据平面采用 SRv6 封装方式转发报文。租户的物理站点分散在不同位置时，SRv6 VPN 可以基于已有的服务提供商或企业 IP 网络，为同一租户的不同物理站点提供二层或三层互联。根据 VPN 业务种类，SRv6VPN 可以分为：L3VPN 业务（IPL3VPN over SRv6 和 EVPN L3VPN over SRv6）和 L2VPN 业务（EVPN VPWS over SRv6 和 EVPN VPLS over SRv6）。

SRv6 VPN 技术具有如下优势：

- **智能控制：**SRv6 基于 SDN 架构设计，跨越了应用和网络之间的鸿沟，能够更好地实现应用驱动网络。 SRv6 中转发路径、转发行为、业务类型均可控。
- **部署简单：**SRv6 基于 IGP 和 BGP 扩展实现，无须使用 MPLS 标签，不需要部署标签分发协议，配置简单。在 SRv6 网络中，不需要大规模升级网络设备，就可以部署新业务。在 DC（Data Center，数据中心）和 WAN（广域网）中，只需网络边界设备及特定网络节点支持 SRv6，其他设备支持 IPv6 即可。
- **易于实现：**VPN 等新业务 SRv6 定义了多种类型的 SID，不同 SID 具有不同的作用，指示不同的转发动作。通过不同的 SID 操作，可以实现 VPN 等业务处理。日后，用户还可以根据实际需要，定义新的 SID 类型，具有很好的扩展性。
- **简化维护：**仅需要在源节点上控制和维护路径信息，网络中其他节点不需要维护路径信息。

互联网技术详解
大话 H3C Comware v9 容器操作系统

互联网系统部 吴银怀

容器作为一种轻量化的虚拟化技术，共享 Linux 操作系统内核，直接使用服务器硬件资源，以更少资源获取更高性能，实现容器间的隔离和资源分配控制，达到应用之间的相互隔离并提供可移植性和版本控制等目的，解决开发与运维环境不一致的问题。

基于分层文件系统 Rootfs 和联合文件系统技术，容器封装可以运行于各种主要 Linux 发行版本，具有灵活度更高、应用部署快的特点，带来应用开发、部署和管理方式的巨大飞跃。

常见的经典容器管理应用主要有 Docker、LXC 等。Docker 对容器底层技术（Namespace、CGroups、UnionFS 等）进行了巧妙的包装，极大的简化了容器的使用，使得容器技术迅速普及，应用越来越广泛和深入。

H3C Comware v9 操作系统平台集成了 Docker，并添加了资源限制相关的功能，用于支持用户运行第三方应用。

Comware v9 容器操作系统平台简介

作为支持 H3C 全系列多种网络设备运行的网络操作系统，H3C Comware V7 网络操作系统历史悠久，基于 Linux 内核，具有统一化、模块化、标准化、虚拟化、分布式、高可用、继承性和开放性等特点。

在 Docker 容器如火如荼发展的大背景下，H3C Comware 网络操作系统进一步发展和完善，基于原生 Linux 内核，完美支持容器化，并且在模块化、轻量化、高可靠、高性能、自动化、虚拟化、可视化、继承性、开放性、可编程等方面相比 Comware V7 平台进一步增强：

容器化，可通过容器采用独立软件的形式发布 Comware 的某一软件功能，业务间独立部署，并增加了第三方容器网络对外互通的支持；

模块化，Comware v9 采用分层设计，支持模块 / 粒度化，实现独立模块的软件故障封闭及修复，增强系统的稳定性；

轻量化，利用数据库和消息机制实现模块间松耦合，功能模块可灵活裁剪，使得系统轻量化，最终实现可以独立发布的轻量化网络功能；

高可靠，支持 NSR 等技术，且 Comware v9 可利用共享数据库技术进行单机 ISSU 升级，

高性能，集成 DPDK，采用业界新的用户态转发技术，提升 CPU 转发性能，相比同硬件条件下 v7 平台 CPU 转发效率成倍提升；

自动化，既可支持 Zero-touch 部署，也支持 Python/TCL/Bash 等脚本工具和 Puppet、Chef 等自动化工具，提供 Netconf 等接口，支持云计算场景下网络设备自动化上线；

虚拟化，可将 FW、IPS、AV 等网络功能打包为 NFV 组件作为容器应用提供服务，实现真正的网络功能虚拟化。

继承性，完美继承 Comware v7 平台功能特性及命令行，平滑适配用户习惯；

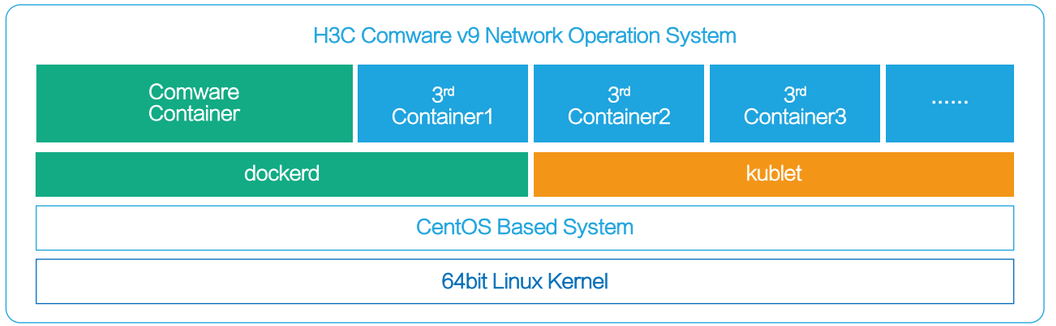
可视化，支持基于 gRPC（Google Remote Procedure Call，Google 远程过程调用）/GNMI（gRPC Network Management Interface gRPC 网络管理接口）、INT（IN-BAND Telemetry， 带内遥测）、ERSPAN 的 Telemetry 技术，并可支持基于 BCM 厂商硬件平台的硬件 Telemetry Streaming 及 TCB、MOD 技术；

开放性，基于原生 Linux 内核，支持 x86 平台，能以容器的方式部署扩展网络功能及第三方应用；

可编程，Comware V9 平台可提供丰富的可编程手段，支持在设备中安装第三方 RPM 软件包，支持容器管理和编排支持如 Docker CE 和 k8s，支持 Native Linux Container Image-NLCI， 支持 H3C SDK 软件开发工具包 - 即 CSDK 可编程环境等。

Comware v9 容器操作系统架构

H3C Comware V9 网络操作系统基于标准 Linux 内核及定制的 CentOS 发行版基础系统；集成了 Docker Community 版本及 Kubelet（随版本升级而更新），支持 Docker 原生命令，比如 image、ps、pull、run 等，其架构逻辑如下：

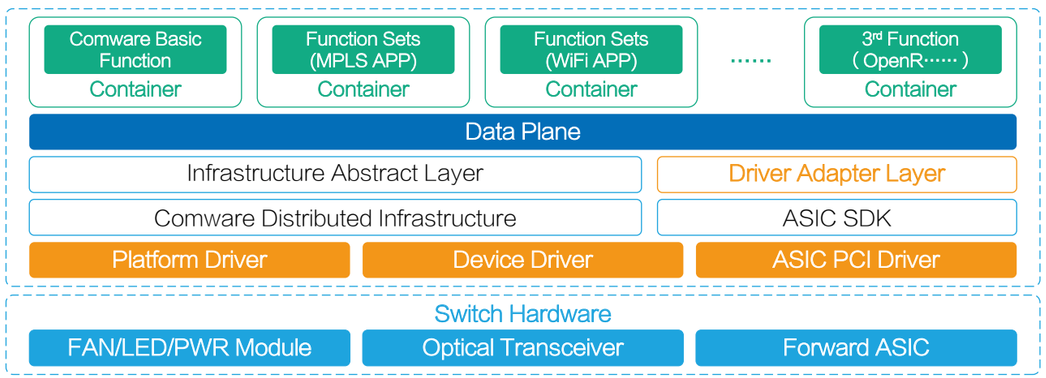


Comware 命令行提供了 Docker Client 执行入口和 Kubelet 启动入口，Docker 及 Kubelet 命令受 Comware 用户认证体系及 RBAC 管理，只有被授权用户才可以访问，保证系统安全。

通过 Comware 命令行直接执行 DDocker Client 命令（即 docker 命令），进入 Comware Shell 系统视图执行 Docker 相关命令，也可以通过 Comware 命令行执行 Kubelet 使能和去使能命令来启动和关闭 kubelet 服务；

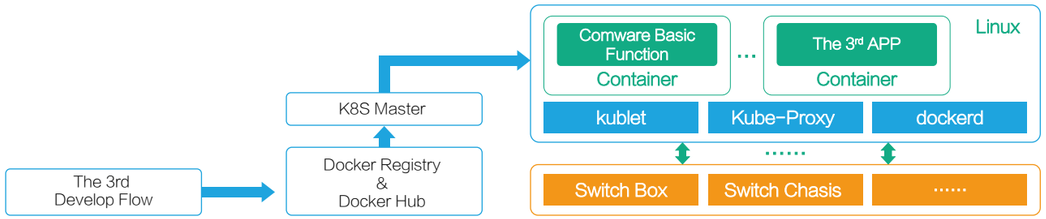
Comware 对容器的保存和恢复做了特殊的实现，可通过执行 save 命令保存容器，设备可借助保存的容器文件自动恢复容器，通过指定容器重启策略 always 实现自动重启。

Comware V9 的容器化架构，为 Comware 以及第三方应用提供独立的运行环境，将其容器架构进一步细化如下：



Comware 容器和用户创建的容器均运行在 Linux 内核之上，直接使用物理设备的硬件资源，利用 Linux namespace 和 cgroups 等技术来实现容器间的隔离和资源分配控制。

Comware V9 集成了 K8S 和 Docker daemon，Comware 的基本功能运行在 Comwre 容器中，可作为一个节点接受 Kubernetes Master 的调度和管理，支持容器镜像形式发布虚拟化产品，并通过 K8S 对设备上运行的第三方容器进行大规模、集群化部署和集中式管理，不需要进行 Comware V7 将第三方软件源码通过交叉编译环境编译过才能运行在 Comware 的操作，因此管理员可将设备作为类似服务器的产品进行部署和管理，实现应用快速发布。



H3C Comware V9 支持用户在 H3C 物理设备上部署第三方应用，包括基于 RPM（Red-Hat Package Manager，RPM 软件包管理工具）技术打包的应用和基于 Docker 容器化的应用。

根据第三方应用的运行环境不同，第三方应用可以分为两类，不同种类的第三方应用的部署方式不同。

运行在 Comware 中的第三方应用

如开源应用、用户开发的应用、H3C 基于开源软件二次开发集成在 Comware 的 Docker/Kubelet 应用、H3C 基于开源软件二次开发但没集成在 Comware 软件包中的应用。

Docker 和 Kubelet 集成在 Comware 软件包中，跟随 Comware 软件包一起安装和升级；其他的运行在 Comware 中的第三方应用用户可以使用 RPM 来安装、卸载、查看。

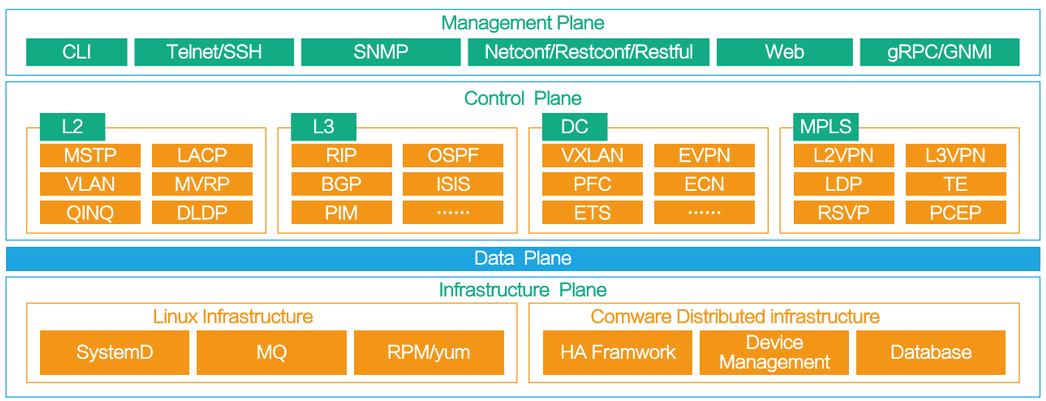
运行在 Comware 中的第三方应用和 Comware 中支持的特性一样，共享系统资源。

运行在容器中的第三方应用

用户需要先创建、启用容器，这样的容器称为第三方应用容器。

有些容器本身就是一个独立的第三方应用，有些容器中支持单独安装第三方应用。用户可以在第三方应用容器内使用应用提供的接口来启动和停止这些应用，停止、删除容器会导致容器中运行的所有应用被停止和删除。容器和 Comware 之间互相隔离。

以上是从容器架构角度对 Comware v9 的剖析。从业务架构角度讲，Comware v9 操作系统平台由四个平面组成：



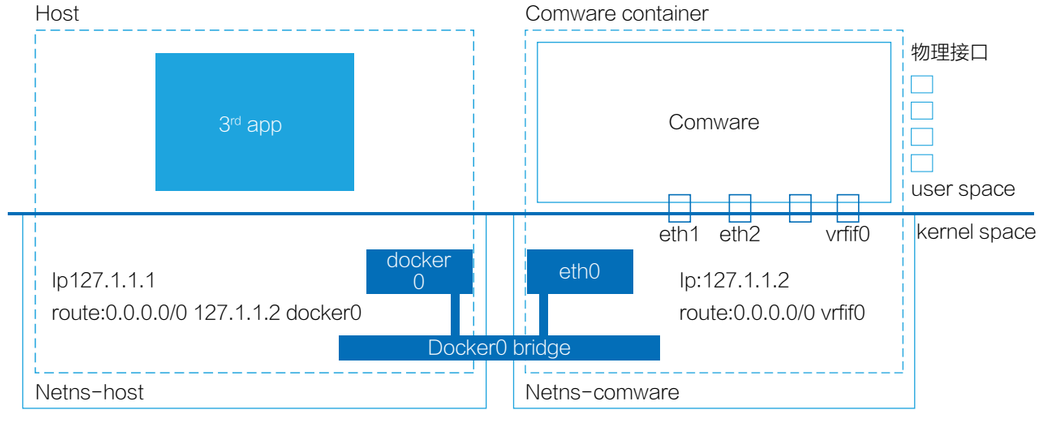
基础设施平面，在操作系统的基础上提供业务运行的软件基础，包括操作系统基础服务和业务支撑功能。基础服务功能是与业务无关的各种软件功能，包括 Linux 操作系统的各种基本功能、C 语言库函数、数据结构操作、标准算法等。业务支撑系统是整个系统业务运行的基础，为 Comware 各业务提供软件和基础设施，后面提到的各种系统架构中涉及的基础功能均在这部分提供，可以作为逻辑架构中的基本功能应用容器而独立存在。

- **数据平面**，提供数据报文转发功能，包括本地报文的收发，即 IPv4/IPv6 协议栈、socket、基于各层转发表的数据转发功能等。Comware V9 的数据平面，支持基于 DPDK（Data Plane Development Kit，数据平面开发套件）的转发模型。
- **控制平面**，运行路由、MPLS、链路层、安全等各种路由、信令和控制协议，生成各种转发表项以控制数据平面的转发行为。Comware V9 的控制平面主要可以分为 Layer2、Layer3、DC、MPLS 等几类业务。
- **管理平面**，对外提供设备的管理接口，如 CLI、Telnet、SSH、SNMP、HTTP、Netconf、Restful 和 gRPC 等。通过管理平面，实现人机交互，对 Comware V9 进行设置、监控、管理。

可以通过独立的容器运行 Comware 的基本功能，如库函数、数据库、基础算法等， 并通过增加容器来运行扩展功能，比如数据平面报文的转发功能、控制平面的 BGP 功能 /OSPF 功能 / 安全协议功能 /MPLS 功能、管理平面的 Telnet 功能等，每个功能都可以作为单独的容器应用被发布、部署和管理。再比如交换机上使用无线插卡，无线功能可封装成一个独立的容器镜像文件，运行在独立的容器中，运行时依附 Comware 基本功能，又可独立升级而不受其他功能影响。

Comware 可为 Host 应用（dockerd 及 kubelet）和第三方容器提供网络支持，运行在 host 或者第三方容器中的应用程序，可以与 Comware 中的应用通信（东西向流量），或者通过 Comware 与外界通信（南北向流量）。

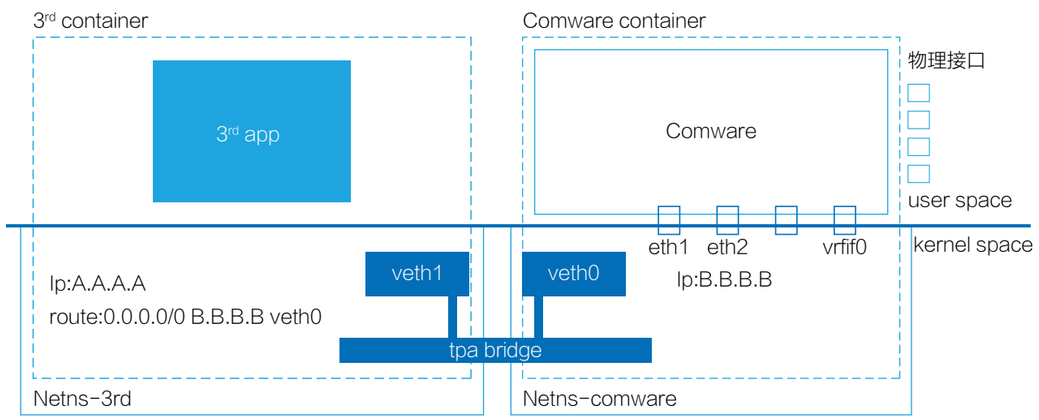
Comware 与 Host 分属不同的网络空间，可通过 bridge 为直接运行在 Host 上的应用提供网络：



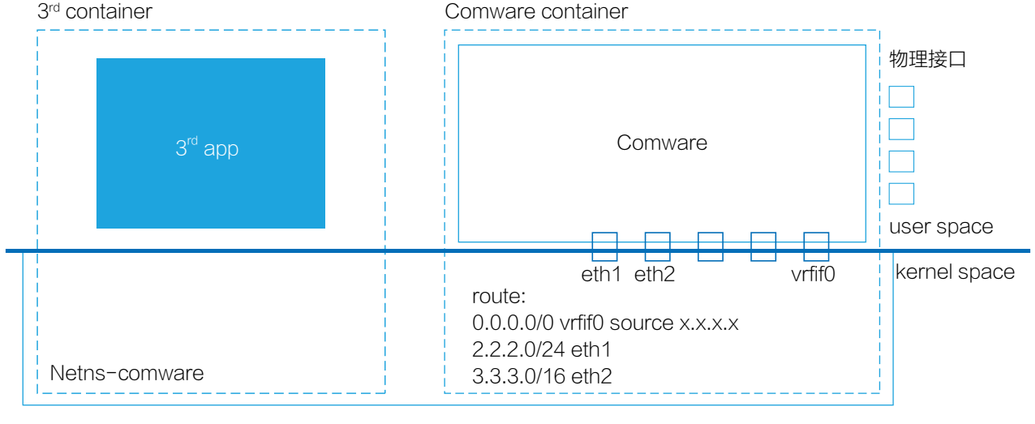
系统会缺省为 Comware 的网络空间创建 eth0 接口，并将其加入到 docker0 bridge 上。修改 host 侧 docker0 接口的 IP 地址，并且下发缺省路由，使其下一跳为 Comware 侧的 eth0 接口地址，这个时候 Comware 作为 Host 的网关为其提供网络。

Comware 为第三方容器提供网络支持需要按第三方容器与 Comware 是否共享网络空间分别描述。

若第三方容器与 Comware 共享网络空间，Comware 会在内核创建 tun 口，对应 Comware 的三层接口，以及每 vrf 的接口（vrfif0），并且会将三层接口的 IP 地址下发在内核对应的 tun 口，触发内核生成直连网段路由。同时会下发本 vrf 内的缺省路由在 vrfif0 接口，缺省路由的源地址为 Loopback0 接口地址。内核转发会使用这些路由，并且通过 tun 口将报文交给 Comware：



若第三方容器与 Comware 不共享网络空间，第三方应用无需看到 Comware 的地址和接口，第三方应用则可以拥有独立的 IP 地址与外部进行通信。Comware 和每个第三方容器都会创建一个 veth 接口并加入 tpa bridge，使得第三方容器和 Comware 在逻辑上处于同一个广播域。为第三方容器内的 veth 口分配地址，使其与 Comware 内的 veth 口处于同一个网段，然后在第三方容器内添加缺省路由，将 Comware 内的 veth 口的 IP 地址作为缺省路由的下一跳，即 Comware 等同于三方容器的缺省网关：

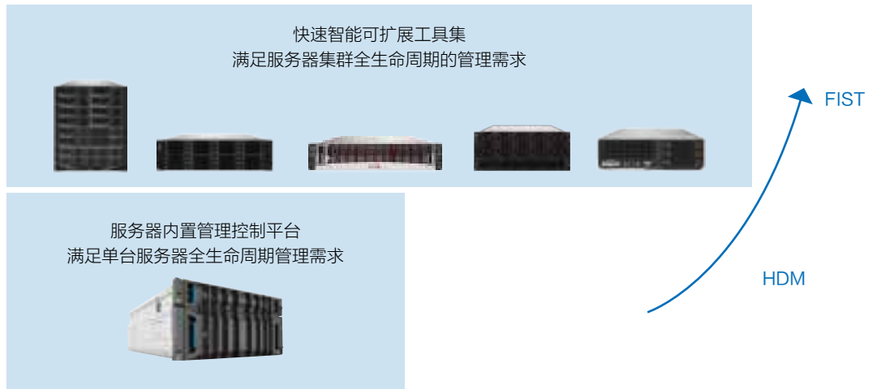


H3C 实现了在 Comware v9 容器操作系统平台中部署 RPM 应用，并且安装第三方应用 BusyBox，并完成了互通。目前 H3C S12500R/S12500CR 云计算核心路由器及 H3C SR6600-I/IE 已经完成对 H3C Comware v9 容器操作系统平台的严格适配并部署上线运行。

H3C 服务器运维管理特性简介

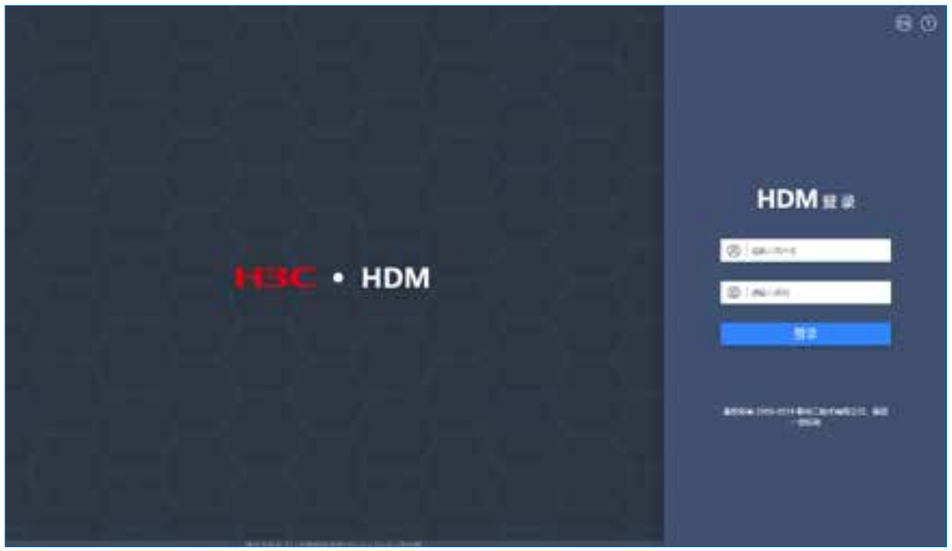
互联网系统部 姜靖

还在为单台服务器管理而担心么？还在为服务器集群运维而苦恼么？新华三数据中心管理软件框架从单台服务器内置管理工具 HDM，到服务器集群轻量管理软件 FIST 能，到新一代数据中心管理协议 Redfish 的支持，能够满足您的 IT 发展阶运维需求，段帮助您解决服务器运维管理问题。下面按照阶段逐步介绍每个阶段使用的管理工具。



HDM 服务器单台内置全生命周期管理

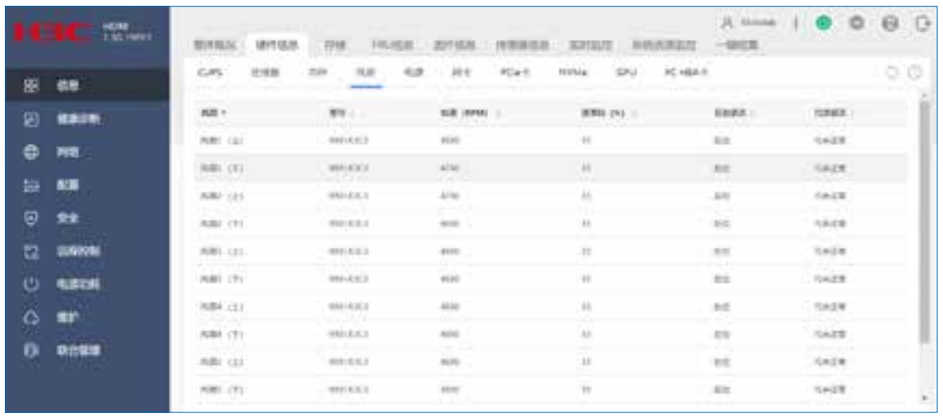
HDM 是新华三服务器内置集成的软硬件管理系统 Hardware Device Management，通过 HDM 可以实现服务器全生命周期的智能运维。可以通过基于 HTTPS 协议的 Web 可视化管理接口访问管理端口 IP 地址，使用设置的用户名和密码，进行访问管理。默认用户名密码就在 H3C 服务器的标签拉片中即可查看。



HDM 通过收集遍布服务器的各种传感器数据，掌握服务器的各部分硬件信息数据，如处理器界面，可以查看处理器的型号，核心数状态，主频，线程，各级缓存容量等细致参数。

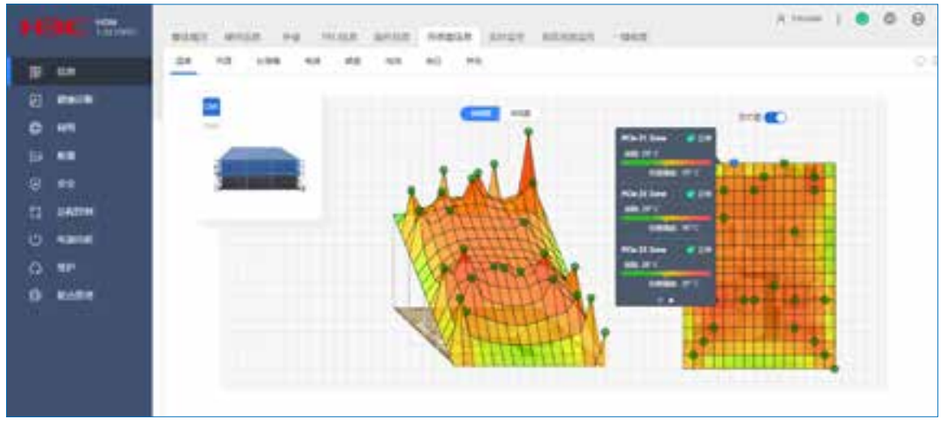
在内存界面，可以清晰直观的展示内存，可以时时了解每颗CPU下可使用的内存容量，验证每个处理器下内存通道使用情况，内存插槽数，内存总容量，工作频率，内存工作电压，确认处理器下的内存插入方式是否可以最优发挥内存性能。

通过硬件信息中的风扇信息，可以获取到服务期内所有风扇的运行状态，包含转速，速度比，运行状态，冗余状态，实时监控服务器风扇运行状态，保证服务器良好散热。



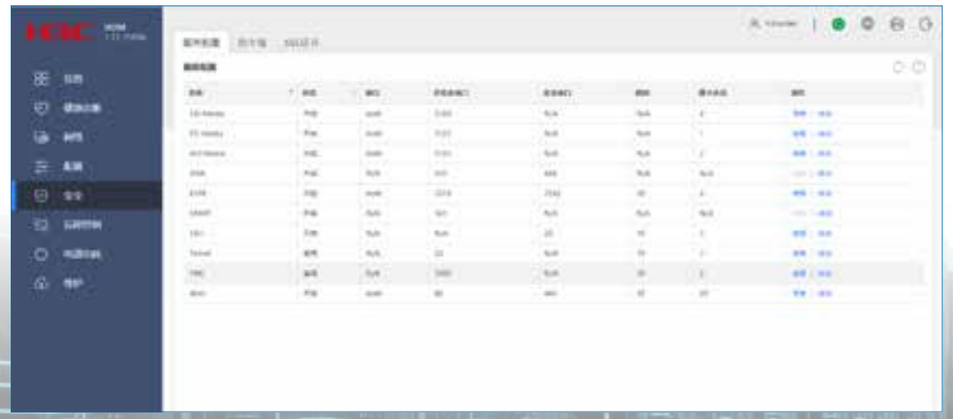
通过信息中的存储 RAID 视图，可以清楚查看到阵列卡信息以及其下所链接的硬盘组成的 RAID 状态，运行状态，链接位置等等，提供简单直观的存储数据并可以进行配置管理。

通过直观精准的 3D 海洋温感显示，可以直观立体的查看服务器内部温度分布，了解服务器散热情况，针对异常温度点，辅助快速排查，让服务器内部温度分布尽在掌握。



同时 HDM 能够实现远程便捷的即时管理工具，虚拟 KVM 可以帮助用户端利用本地的视频、键盘、鼠标对远程的设备进行监视和控制，虚拟媒体可以帮助用户挂在本地系统光盘帮助远程安装系统，提供实时操作异地设备的管理方式，实现整个生命周期的各类操作，使运维工作更加得心应手。

通过账户权限管理，可以设置不同用户访问权限，增加防火墙设置，强大的加密库等安全措施，实现了端到端的安全性，保证服务器安全稳定运行。



在遇到的系统崩溃，如 Windows Server 的崩溃，一般会有蓝屏报错，但是这种瞬间屏幕代码提示很难被直接记录获取到，为了辅助故障定位分析，针对蓝屏报错，提供自动的蓝屏截图机制或者系统崩溃前的录像回放记录，辅助分析，帮助快速定位故障原因。

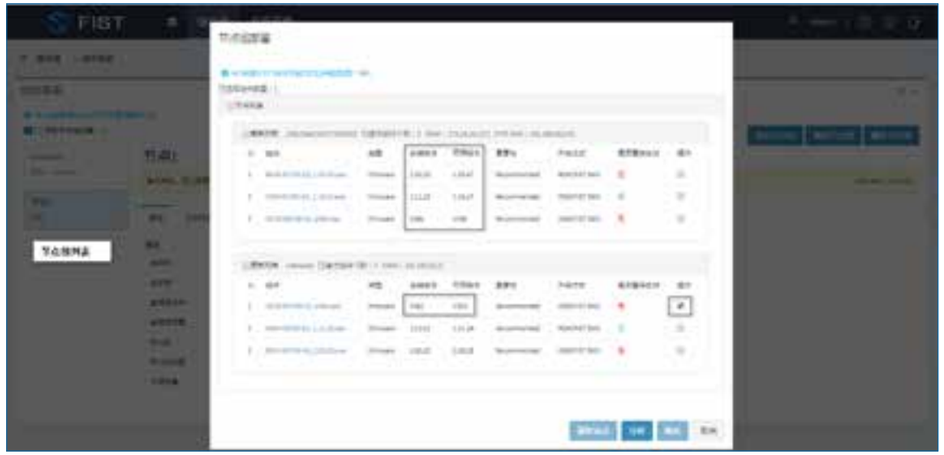
FIST 服务器集群全生命周期管理

FIST 是一个轻量级的集群管理软件，可以让服务器从单一管理步入到集群式管理。主要有四个核心功能模块，机箱管理，交换机管理，服务器管理，系统管理。通过这些功能模块可以实现服务器集群运维管理。

FIST 可以通过 IPMI、Redfish、SNMP 协议批量向 HDM 获取以下信息：CPU 信息 \ 内存信息 \BIOS 信息 \ 主板信息 \ 健康信息 \ 存储信息 \PCIE 信息实现批量监控服务器状态。

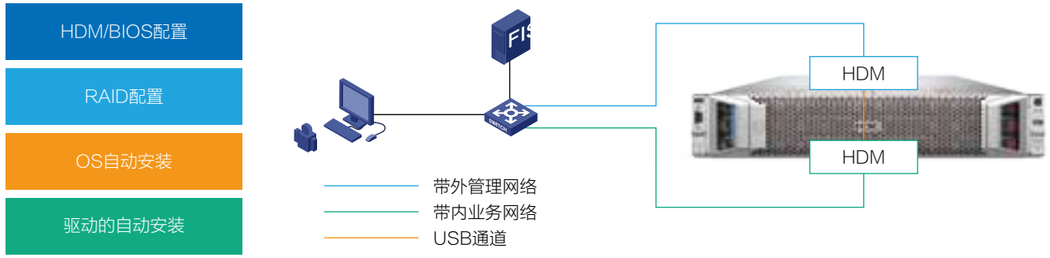
FIST+FIST SMS 可以实现对服务器的带内监控，FIST SMS 运行在客户的业务系统中，可以获取系统相关的信息以及通过 USB 通道可以和 HDM 进行联系。同样可以获取到 CPU 信息 \ 内存信息 \BIOS 信息 \ 主板信息 \ 健康信息 \ 操作系统信息 \ 网络信息等实时监控硬件状态。

FIST 模板管理即 Profile 功能，可以实现批量 HDM&BIOS 配置复制、RAID 配置、OS 安装、驱动安装等功能。如果客户想使用这些功能，需要 FIST 配合 iFIST、HDM 一起使用。FIST 组件更新功能可以实现批量升级服务器的固件和驱动，此功能需要 FIST、HDM、FIST SMS 以及 REPO 一起使用。



FIST 在管控服务器的基础之上可以纳管刀箱，实现对整框的一体化管理；可以监控刀箱内的服务器网络设备等的运行状态，对刀片服务器进行配置管理。

FIST 配合 HDM 实现了带外管理功能，FIST 配合内置 iFIST 实现了带内管理，HDM 和 iFIST 通过 USB 通道实现通信，最终几个功能模块相互配合实现的 Profile 功能，帮助服务器批量部署配置 HDM,BIOS,RAID，批量自动安装系统，更新驱动以及固件，实现服务器的快速上线和配置迁移。

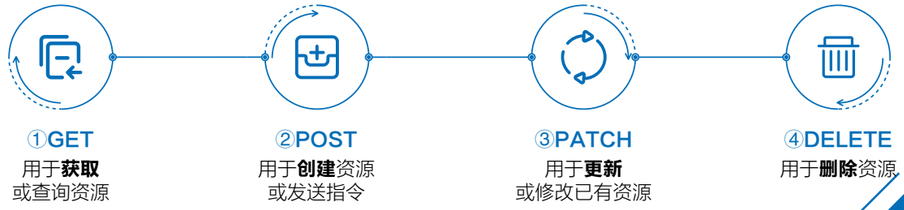


Redfish- 新时代数据中心管理标准

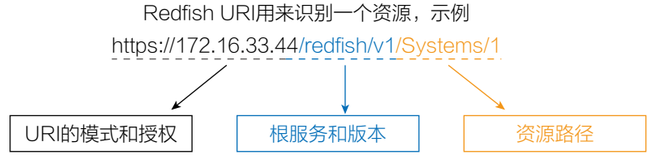
互联网系统部 姜靖

新一代数据中心的规模越来越庞大，运维管理越来越复杂，H3C 内置的服务器管理系统 HDM 支持多种管理接口。如 IPMI 接口，基于 HTTPS 的 Web 可视化接口，简单网络管理协议 SNMP 接口和基于 RESTful 架构的 Redfish 接口等。其中 IPMI 是一个受到广泛支持的行业标准，它指定了一组接口，以提供独立于主机系统的 CPU、固件和操作系统的带外管理和监视功能。但 IPMI 作为较早期带外管理的标准，标准的功能有限，所以各服务器厂商在 IPMI 基础上扩展了很多实用功能，但是因为各厂商标准不够统一，命令不完全通用，导致这种传统标准不足以满足管理众多不同厂商多节点多服务器的融合架构，进而导致现代数据中心统一运维管理难度增加。在这种背景下，Redfish 应运而生， Redfish 是 DMTF（工业化组织）维护的标准之一，由多个软硬件厂商参与，推动全球采用互操作管理标准。在设立之初就设定了安全，高可扩展管理，人类可读取界面，基于现有硬件可实现四项基本目标。其中基于现有硬件指的就是支持 IPMI 的 BMC 平台，这样 Redfish 就可以很好的满足了原有的硬件兼容性，可以保证从 IPMI 平滑升级到 Redfish。通过使用 HTTPS 请求方式，保证了 Redfish 的数据传输过程安全可靠。通过 RESTful 形式的 API 来实现高可扩展管理，而且 Redfis 这些访问的数据格式都是 JSON 形态提交或者返回，它比 IMPI 的 RAW 命令和 XML 更简单易读，是实现人类可读数据界面的重要手段。所以可以看到 Redfish 从设计之初就是为融合基础架构的通用管理协议而设定的，真正降低了开发运维复杂性，易于实施、易于使用，安全性更高，扩展性更强，是可被称为“下一代数据中心管理标准”的。

Redfish 服务器管理标准是基于 Restful 接口和 JSON 数据模型。支持 HTTP/HTTPs 两种请求方式。每个 Redfish 请求都以 UTF-8 编码的 JSON 格式提交或返回一个资源结果。Redfish 接口包含请求动作和 URI（统一资源标识符）。请求动作包含四种：GET；POST；PATCH；DELETE。GET 用于获取或者查询资源，POST 用于创建资源或者发送指令，PATCH 用于更新或修改已有资源，DELETE 用于删除资源。



在 Redfish 中，每个 URL 都代表一个资源，一个服务或一组资源。根据 REST 原则，使用统一资源标识符（URI）指向资源，客户端与资源进行交互。资源的格式根据 Redfish 架构来定义，客户端再根据 Redfish 架构来确定正确的语义（Redfish 语义被设计的非常直观）。设备使用的 URI 形式通常为 https://device_ip/redfish/v1/path，可分为三个部分。第一部分 device_ip 表示要访问服务器的 HDM IP 地址。第二部分是服务和 redfish/v1/，当前设备的 Redfish 是基于 redfish v1 版本开发的。第三部分表示唯一资源 /path。



DM Redfish 接口基于 Redfish V1.1.1 开发。常用推荐安装环境有两种。

方法一：推荐使用网管工具 Postman 进行 Redfish 接口管理。工具下载链接：https://www.getpostman.com/downloads/

方法二：使用 Firefox 浏览器的 httprequester-2.2-fx.xpi 插件（该插件需要 Firefox 浏览器版本低于 56.0b3(64 位)）

目前我们利用 Redfish 管理接口可以实现包括用户管理、获取服务器信息、管理模块信息等常用 HDM 和 BIOS 的管理配置与管理。通过 Redfish 在 HDM 中可以对机箱进行管理，查询机箱集合资源信息，对服务器主机机型电源控制和查询相关共走，对主机风扇，CPU，内存，PCIE 卡，扩展卡，UID 灯，网卡查询或设置等相关操作，可以支持 HDM 主备切换，HDM 重启，支持 RAID 卡信息获取，逻辑磁盘创建，删除，信息查询等，支持固件信息查询和固件升级，支持 NTP，SNMP，SMTP，SYSLOG 信息查询和配置，支持 HDM 账户管理，用户操作接口等。可以实现 BIOS 选项多项功能的设置，如 BIOS 中的 Advance 选项，Platform Configuration 选项，Socket Configuration 选项，Server Management 选项，Secunity 选项，Boot 等选项中的设置。

菜单	描述
Advanced	ACPI设置、休眠设置、终端配置、串口重定向设置、支持传统USB设备功能、XHCI切换、支持大容量USB存储设备、配置OptionROM的加载策略、PXE设置、启动设备检测计数设置
Platform Configuration	支持SATA控制器设置、USB设置、显示设备选择、BIOS串口日志输出设置、SOL功能设置、软件错误注入支持设置
Socket Configuration	支持CPU Core设置、超线程功能设置、Monitor/Mwait功能设置、Intel TXT功能设置、Intel硬件辅助虚拟化技术设置、安全模式扩展功能设置、硬件预取设置、EIST（P状态）设置、TDP等级设置、Intel® Speed Select设置、EIST PSD Function设置、Turbo模式设置、 CPU Core频率设置、硬件P状态设置、硬件PM中断设置、EPP设置、C状态设置、T状态设置、热监控设置、功率性能调节设置
Server Management	支持FRB—2定时器、超时策略设置和OS看门狗定时器、超时策略设置
Security	支持安全启动模式配置，提供出厂默认秘钥设置
Boot	支持数字锁定键状态设置、启动模式设置（UEFI和LEGACY）、UEFI Shell使能设置、启动选项设置

表2 BIOS配置项支持列表

新华三数据中心管理软件框架单台服务器内置管理系统 HDM 和集群管理软件 FIST，以及多种服务器管理协议的支持，可以轻松帮助您远程管理控制服务器，对服务器系统和运行状态进行实时监控，通过对账户权限管理，防火墙设置等强大的安全加密措施，实现安全的对服务器和服务器集群的全生命周期的管理运维。特别是在 Redfish 的支持下，可以使得数据中心硬件设备的高度抽象，使用人类可读的先进数据格式，使得管理架构清晰易扩展，众多互联网客户在 Redfish 加持下，更加方便快捷的对其超大规模的数据中心管理运维。

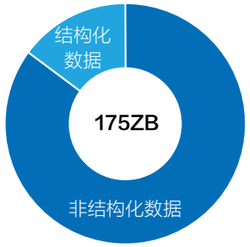
分布式存储技术及实践

互联网系统部 温荣强

背景

在今天的数字化时代，云计算、互联网、社交媒体以及大数据的发展，使得数据量呈现爆炸式增长。据 IDC 预测，2018 年到 2025 年之间，全球产生的数据量将会从 33 ZB 增长到 175 ZB，复合增长率达到 27%，其中超过 85% 的数据都会是处理难度较大的非结构化数据。预计到 2030 年全球数据总量将达到 3,5000EB。

2025年全球数据超过175ZB，85%为非结构化



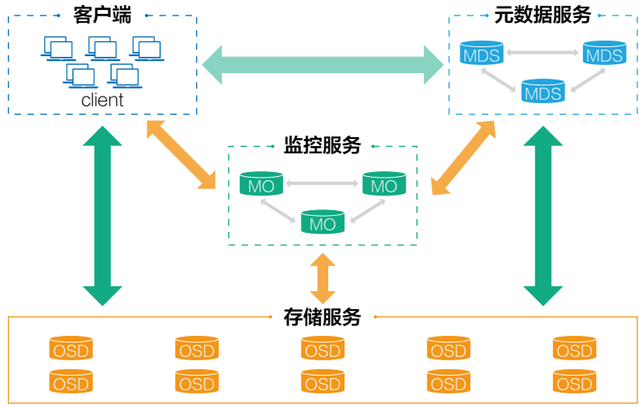
1ZB=1,000,000,000TB 数据来源：IDC

传统存储在应对这些 PB 甚至 EB 级海量非结构化数据存储需求难以满足的同时，其技术架构也面临着诸多挑战，已经很难满足不断增长的数据需求，主要包括超大规模的横向扩展、越来越高的性能要求、数据长期存储的可靠性、统一资源池的管理、更低的 TCO 总体拥有成本等。

在这种情况下，用户迫切需要新的存储架构来解决这些问题，来满足不同应用的存储资源服务需求，所以随着客户需求的变更及技术的迭代升级发展，分布式存储技术应运而生。

分布式存储架构

分布式存储系统是一个高性能、可扩容的分布式存储系统，并且它可提供对象、块和文件三种种类存储服务，以满足客户复杂的业务应用场景，分布式存储架构逻辑上可分为四部分：OSD（Object-based Storage Device，对象存储设备）、Monitor（监控服务）、Client（客户端）、MDS（metadata server，元数据服务）。



对象存储设备（OSD）

OSD 包括操作系统和文件系统的计算存储单元，其硬件包括处理器、内存、硬盘以及网卡等，守护进程提供数据存储服务。在实际应用中，通常将每块硬盘（SSD 或 HDD）对应一个 OSD 守护进程，并将其视为 OSD 的硬盘部分，处理器、内存、网卡等在同一主机的多个 OSD 之间进行复用。

监控服务（Monitor）

监控服务持有集群视图信息，包含关于集群本身的逻辑状态和存储策略的数据表示。集群视图包括监控视图、OSD 视图、元数据服务视图，这些视图构成了集群的元数据。集群视图信息量较少，只有在集群的物理设备（如主机、硬盘）和存储策略发生变化时视图信息才发生改变。

客户端（Client）

用户通过客户端登陆集群并对集群数据进行读写操作。客户端通过与 OSD 或者监控服务的交互获取集群视图，然后直接在本地进行计算，得出数据的存储位置后，便直接与对应的 OSD 通信，完成数据的各种操作。在此过程中，客户端可以不依赖于任何元数据服务器，不进行任何查表操作，便完成数据访问流程。

集群元数据服务（MDS）

元数据是定义数据的数据，主要是描述数据属性的信息，用来支持如指示存储位置、指示历史数据、查找资源、文件记录等功能。

数据存储过程

在存储过程中存储的大数据都会被切分成 Objects，Objects size 大小可以由管理员调整，通常为 2M 或 4M。每个对象都会有一个唯一的 ID。

File：File 就是用户需要存储或者访问的文件。

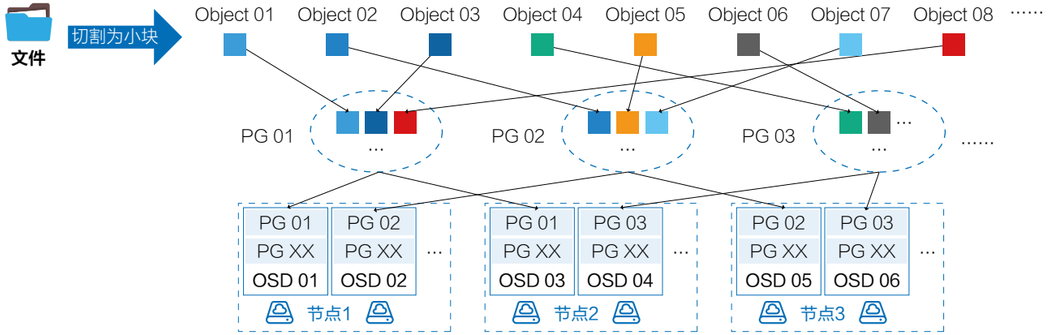
Object：object 是底层系统所看到的对象。Object 与上面提到的 file 的区别是，object 的最大 size 由管理员限定 (通常为 2MB 或 4MB)，以便实现底层存储的组织管理。

PG(Placement Group)：对 object 的存储进行组织和位置映射。具体而言，一个 PG 负责组织若干个 object (可以为数千个甚至更多)，但一个 object 只能被映射到一个 PG 中，即，PG 和 object. 之间是一对多 " 映射关系。同时，一个 PG 会被映射到 n 个 OSD 上，而每个 OSD 上都会承载大量的 PG, 即，PG 和 OSD 之间是“多对多”映射关系。在实践当中，n 至少为 2, 如果用于生产环境，则至少为 3。一个 OSD 上的 PG 则可达到数百个。

OSD: OSD 的数量事实上也关系到系统的数据分布均匀性，因此其数量不应太少。在实践当中，至少也应该是数十上百个的量级才有助于系统的设计发挥其应有的优势。

数据映射过程

基于上述定义，便可以对寻址流程进行解释了。具体而言，寻址至少要经历以下三次映射：



（1）File-> object ID 映射
（2）Object>PG 映射，hash&mask->pgid
（3）PG>OSD 映射，CRUSH 算法

全分布式架构中，客户端是直接读或者写存放在 OSD 上的对象存储中的对象(data object)的，因此，需要走完 (Pool, Object) → (Pool, PG) → OSD set → OSD/Disk 完整的链路，让 client 存储目标数据 object 的具体位置。

数据写入时，文件被切分成 object，object 先映射到 PG，再由 PG 映射到 OSD set。每个 pool 有多个 PG，每个 object 通过计算 hash 值并取模得到它所对应的 PG。PG 再映射到一组 OSD (OSD 个数由 pool 的副本数决定)，第一个 OSD 是 Primary，剩下的都是 Replicas。

分布数据的过程：首先计算数据 x 的 Hash 值并将结果和 PG 数目取余，以得到数据 x 对应的 PG 编号。然后，通过 CRUSH 算法将 PG 映射到一组 OSD 中。最后把数据 x 存放到 PG 对应的 OSD 中。这个过程中包含了两次映射，第一次是数据 x 到 PG 的映射。PG 是抽象的存储节点，它不会随着物理节点的加入或则离开而增加或减少，因此数据到 PG 的映射是稳定的。具体步骤：

（1）创建 Pool 和它的 PG。根据上述的计算过程，PG 在 Pool 被创建后就会被 MON 在根据 CRUSH 算法计算出来的 PG 应该所在若干的 OSD 上被创建出来了。也就是说，在客户端写入对象的时候，PG 已经被创建好了，PG 和 OSD 的映射关系已经是确定了的。
（2）客户端通过哈希算法计算出存放 object 的 PG 的 ID：

- 客户端输入 pool ID 和 object ID （ 比如 pool = “liverpool” and object-id = “john” ）
- 对 object ID 做哈希
- 对该 hash 值取 PG 总数的模，得到 PG 编号 （ 比如 58 ）（ 第 2 和第 3 步基本保证了一个 pool 的所有 PG 将会被均匀地使用 ）
- 对 pool ID 取 hash
- 将 pool ID 和 PG ID 组合在一起 (比如 3.23) 得到 PG 的完整 ID。也就是：PG-id = hash(pool-id). hash(objet-id) % PG-number

（3）客户端通过 CRUSH 算法计算出 object 会被保存到 PG 中 OSD 位置。

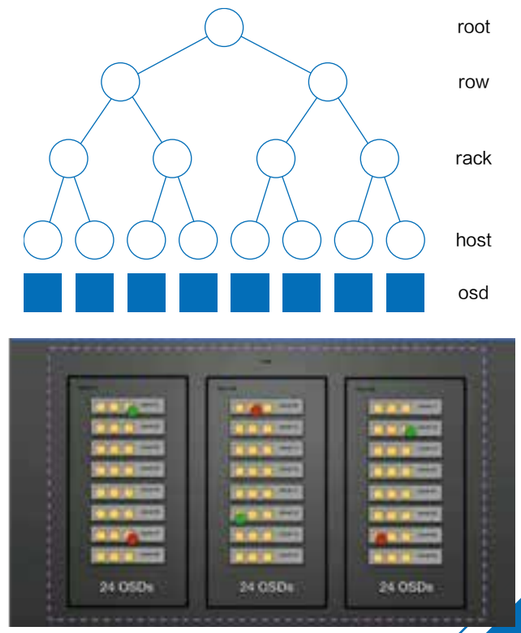
对客户端来说，只要它获得了 Cluster map，就可以使用 CRUSH 算法算出某个 object 将要所在的 OSD 的 ID，然后直接与它通信。

- client 从 MON 获取最新的 cluster map。
- client 根据上面的第 (2) 步计算出该 object 将要在的 PG 的 ID。
- client 再根据 CRUSH 算法计算出 PG 中目标主和次 OSD 的 ID。也就是：OSD-ids = CURSH(PG-id, cluster-map, crush-rules)。

CRUSH 算法

为了保证负载均衡，保证新旧数据混合在一起。但是简单 HASH 分布不能有效处理设备数量的变化，导致大量数据迁移。应用了 CRUSH 算法，它是一种伪随机数据分布算法，它能够在层级结构的存储集群中有效的分布对象的副本。CRUSH 实现了一种伪随机 (确定性) 的函数，它的参数是 object id 或 object group id，并返回一组存储设备 (用于保存 object 副本 OSD)。CRUSH 需要 cluster map(描述存储集群的层级结构)、和副本分布策略 (rule)。

CRUSH 算法根据每个设备的权重尽可能概率平均地分配数据。分布算法是由集群可用存储资源以及其逻辑单元的 map 控制的。这个 map (如下图所示) 的描述类似于一个大型数据中心的描述：数据中心 (root) 由各个机房 (row) 组成，机房由一系列的机柜 (rack) 组成，机柜装满服务器 (host)，服务器装满磁盘 (osd)。数据分配的策略是由定位规则来定义的，定位规则指定了集群中将保存多少个副本，以及数据副本的放置有什么限制。例如，可以指定数据有三个副本，这三个副本放置在不同的机柜中，使得三个数据副本不共用一个物理电路。



给定一个输入 x，CRUSH 算法将输出一个确定的有序的储存目标向量。当输入 x，CRUSH 利用强大的多重整数 hash 函数根据集群 map、定位规则、以及 x 计算出独立的完全确定可靠的映射关系。CRUSH 分配算法是伪随机算法，并且输入的内容和输出的储存位置之间是没有明显相关的关系。我们可以说 CRUSH 算法在集群设备中生成了“伪集群”的数据副本。集群的设备对一个数据项目共享数据副本，对其他数据项目又是独立的。

CRUSH 算法通过每个设备的权重来计算数据对象的分布。目的是利用可用资源优化分配数据，当存储设备添加或删除时高效地重组数据，以及灵活地约束对象副本放置，当数据同步或者相关硬件故障的时候最大化保证数据安全。

分布式存储产品

H3C X10000 海量存储系统，是 H3C 自主研发的新一代全对称分布式存储。X10000 在同时支持块、文件、对象与 HDFS 存储能力，最大支持 4096 个节点的横向扩展，单一命名空间支持 EB 级容量，系统的性能和容量随节点数增加呈线性增长。

该产品拥有高性能、高扩展、高可靠、易管理维护等特性，广泛适用于互联网行业云计算、虚拟化、数据库、文件共享、票据影像、HPC、视频监控、归档备份等场景，为用户提供海量存储资源池。

以 ARM 为代表的多元算力正在成为趋势

互联网技术部 姜月良

ARM 多核心集群服务器需求爆发

随着近年来云游戏、云手机的概念普及，以及移动互联网应用的爆发增长，全球数据量指数级增长，但同时作为能耗大户，全球各国对数据中心能耗指标做出了硬性要求，ARM 架构凭借高性比价、低功耗、开源授权模式以及完备便捷的配套开发工具等优势，开始进入云数据中心的视野。

2008 年，基于 ARM 架构的 Android 操作系统发布，随着技术的发展，在更高性价比、更开放的应用生态需求下，根据网络分析公司 StatCounter 的报告称：2017 年 3 月，Android 首次超过 Windows 的市场份额，Android 用户占比 37.93%，而 Windows 占比 37.91%，虽然这只是 0.02% 的差距，但这表示一个临界点，是“ARM+Android”在移动互联时代战胜 Wintel 的一个趋势。

在移动端处理器市场占据主导地位的 ARM，从未放弃向个人电脑、服务器、数据中心市场的渗透。随着越来越多的厂商，华为、苹果、亚马逊等公司早早的加入了 ARM 集群服务器的行列，开始拥抱 ARM 架构，英特尔及其 X86 架构似乎迎来了危机。即便 ARM 在 2012 年就已经开始支持硬件虚拟化技术，但受限于当时的技术条件和市场需求，鲜有应用。随着 5G 网络的发展，我们真正进入了移动互联网时代，移动应用逐渐云化，5G 催生了云游戏的诞生；Web 应用的加密性也越来越重要，HTTPS 流量越来越大；大数据分布式并行计算成为主流，这些都让 x86 架构的不足逐渐显露出来。随着苹果公司宣布，未来 Mac 电脑逐步改用基于 ARM 架构的处理器。此举无疑成为 ARM 架构打进 PC 市场（X86 架构）的一大标志性事件。基于 ARM 架构的服务器也开始进入了爆发期。

ARM 服务器分类

ARM 服务器包括：ARM 通用服务器和 ARM 集群服务器。

ARM 通用服务器的概念是针对传统的 X86 服务器提出的。X86 服务器是采用 X86 架构（复杂指令集）专用于服务器的 CPU（Intel 和 AMD）设计开发的高性能计算设备，主要针对 PC 端应用。而 ARM 通用服务器则是采用 ARM 架构（精简指令集）的专用于服务器的 CPU（华为鲲鹏 920、飞腾腾云 S2500）设计开发的高性能计算设备，主要针对移动端应用。ARM 通用服务器和 X86 服务器，产品架构基本相同，类似于 PC，也称为 PC 架构。这种架构是以 CPU 为核心，挂载内存、存储设备、显卡及其他外设。

ARM 集群服务器，也被称为 ARM 阵列服务器，是在一个机箱内设计 n 个（n=1 ~ 128 甚至更大）高性能 ARM 处理器计算板卡，每个计算板卡都是一个可独立运行的最小系统，包含处理器、Memory、eMMC 存储、网卡等。各计算板卡通过 1GbE 高速网络互联互通并与外界通信。这些计算板卡通过网络集群的方式，可以灵活实现超高的算力，所有的业务计算放在计算板卡上然后将计算结果通过网络传送给终端，而终端仅承担显示和交互的角色。

ARM 集群服务器项目的主流场景

目前落地的 ARM 集群服务器项目的应用场景主要有 3 类：作为云端服务器，模拟手机环境，为人工智能项目提供 NPU 集群算力。

作为云端服务器

ARM 云端服务器的需求，与 X86 的场景需求基本一致，用户可以像使用当前 X86 架构的服务器一样，在 ARM 云端服务器大规模云化部署应用，从而降低对终端硬件的依赖。ARM 架构处理器相较 X86 架构的处理器在稳定性和功耗方面有很大的优势。

（1）ARM 集群服务器运行基于 Linux 内核的操作系统，对稳定性要求较高的客户在系统稳定性方面深得肯定，Linux 系统处理指令的效率也要远高于 Windows 系统。

（2）ARM 处理的功耗要远低于同等性能表现 X86 处理器，服务器一般是 7*24 小时无休工作的，因此功耗对运营成本有很大的影响。

（3）采用 ARM 处理器的子单元体积小巧，1U 的规格可容纳更多的核心处理器，众核架构设计的 ARM 相较同价位的 X86 单元，ARM 集群服务器提供的算力要高得多。

正是因为 ARM 处理器较 X86 处理器有不可比拟的优势，催生出了市场对于 ARM 集群服务器的庞大需求。

模拟手机环境

模拟手机环境是目前市场需求最旺盛的 ARM 集群服务器的行业领域。移动互联网时代，移动端的应用井喷式增长，无论是游戏、社交、办公都有对应的移动版本。传统的 x86 平台依赖模拟器的指令翻译运行安卓应用，性能损耗大，兼容性也无法保障。此外，服务器端有很多应用需要测试，过去的做法是真机测试，或者手机开发仿真环境，这种做法的资源灵活度低、故障率高、可靠性和易用性都比较差。因此，模拟移动应用所需的手机环境成为了一个刚需。游戏的分发测试、服务器压力测试可以使用模拟手机环境的 ARM 集群服务器。ARM 集群服务器可充分满足基于社交网络的营销、商业化运营的手机环境需求。像快手、抖音、火山小视频、微信等这些发迹于移动互联网的应用，现在很多商家的运营手段是准备 100 多台的廉价手机，然后通过一些软件实现在电脑端对这些移动应用软件的集中控制，这已经算是比较专业的运营手段了。更多的商家是人工守着几十台手机，不停的发消息。这两种方式投入的人力成本和硬件成本都是巨大的。而用 ARM 集群服务器模拟的手机环境，1U 的服务器可模拟 100 多个手机环境，通过显示器和键盘鼠标统一管控操作，效率极高。目前，ARM 平台在移动端和物联网终端的大规模使用，使得对该平台算法和软件的性能优化变得越来越充分。近年来，开源社区对 ARM 平台的支持已经足够用户将 X86 应用在合理的投入下迁移到 ARM 平台。原生 ARM 框架使得从云到端都运行着同一套指令集，Android 应用运行无需 X86 模拟器指令集翻译，云端无缝连接免去了多重指令翻译和转换的环节，运行性能可以较 X86 模拟器架构方案提升高达 80%。

提供 NPU 集群算力

随着 AI 的广泛应用，深度学习已成为当前 AI 研究和运用的主流方式。面对海量数据的并行运算，AI 对于算力的要求不断提升，对硬件的运算速度及功耗也提出了更高的要求。相比于 CPU，GPU 拥有更多的 ALU 用于数据并行处理，GPU 处理神经网络数据远远高效于 CPU，但是需要 CPU 的协同处理，对于神经网络模型的构建和数据流的传递还是在 CPU 上进行。同时存在功耗高，体积大的问题，价格也昂贵，因此对于一些小型设备、移动设备来说将无法使用。一种体积小、功耗低、计算性能高、计算效率高的专用芯片 NPU 诞生了，NPU 是专为神经网络模型设计的处理器，其运行 AI 模型的效率要远高于传统的逻辑处理器。为满足人工智能项目的算力需求，目前所有的 ARM 集群服务器子板均有相应的独立 NPU 产品方案。NPU 工作原理是在电路层模拟人类神经元和突触，并且用深度学习指令集直接处理大规模的神经元和突触，一条指令完成一组神经元的处理。相比于 CPU 和 GPU，NPU 通过突触权重实现存储和计算一体化，从而提高运行效率。当然 NPU 也需要 CPU 协同处理，但是同等功耗下 NPU 的性能是 GPU 的 118 倍，目前这类需求主要应用于边缘计算场景，以及小型的机器学习项目。

H3C ARM 多核心集群服务器方案

相比传统的 X86 服务器集群，ARM 集群服务器的部署更加灵活，运营成本更低，且大多数客户对产品硬件和底层软件有着较高的客制化需求。百度作为我司互联网行业的 top 客户，百度自 2015 年深耕于自主研发的基于 ARM 架构的商业云计算软硬件平台，已有了 6 年的商业化积累，云手机项目作为百度的未来重要的发展方向。新华三作为百度的深度战略合作伙伴，拥有芯片、计算、存储、网络、5G、安全、终端等全方位的数字化基础设施整体能力，双方共同合作百度自研云手机服务器项目，联合开发微 ARM 服务器 H3C R2096 G1。

H3C R2096 G1 ARM 多核心集群服务器，以多块核心板的组合方式，提供标准的软硬件接口，在硬件架构上，以高性价比处理器为核心，支持以 4 个 ARM 架构 SoC（System on Chip，系统级芯片）微服务器单元为粒度，按照业务要求进行灵活按需配置，最大可扩展至 96 个 SoC 微服务器，每个微服务器都是一个可以独立运行的最小系统，包含 CPU、内存、存储、网络等，同时利用微板卡以太网链路技术和良好的机箱空间设计，构建适合于企业数据中心机房和边缘环境部署的高密度、高并发、高性能算力集群。

相对传统的集群服务器，致力于为企业级客户，提供具有高能效比、高性价比、高密设计的云边协同算力，能效比超过传统 x86 服务器的 2 倍，算力高出 1.5 倍，在一个机架服务器机箱内集成多达 384 大核 +384 小核，以及 15T*96 NPU 算力，超强的计算性能，灵活的配置扩展以及高速网络接入能力，非常适用于具有多重业务负载的复杂基础设施环境，包括企业级部署、云环境部署、大数据应用环境等。也可满足互联网行业轻量化场景的业务需求及个性化解决方案，可用于 arm 云桌面、云手机、云游戏、广告、直播、ai、vr、安防、人脸识别和视频、数据中心等行业领域。

“元宇宙”背后有什么？H3C XG310 全面加速客户云游戏与视频流处理

互联网系统部 王晨

前言

元宇宙（Metaverse）的概念来源于尼尔·斯蒂芬森（Neal Stephenson）的著作《雪崩》（Snow Crash），书里面描述了一个人们以虚拟形象在三维空间中与各种软件进行交互的世界。随着疫情常态化发展，全社会的上网时长大幅度增加，线上生活由原先短时期的例外状态成为了常态，由现实世界的补充变成了与现实世界的平行世界。

因此，“元宇宙”在 2021 年突然火爆起来。“元宇宙”的兴起，刺激了正在寻求业务转型的互联网公司，各路资本纷纷下场，无论是国外的 Facebook、Roblox、Epic、Microsoft 等，还是国内的腾讯、字节跳动等，都在积极进行跟进。

“元宇宙”的发展，离不开云计算、区块链、人工智能、5G 等新技术的推动。作为互联网硬件基础设施的服务器，自然也需要迎合客户业务转型的需求。因此，H3C 基于 Intel Xe 架构的 SG1 芯片推出了一款面向流媒体和云游戏应用的专业加速卡——XG310。

H3C XG310 产品规格

H3C XG310 是基于 Intel Xe 架构的 SG1 芯片研发的一款加速卡，SG1 芯片是 Intel 专为视频工作负载、安卓云游戏场景优化的一款处理器，功耗为 23W，频率 0.9~1.1 GHz，拥有 16M L3 缓存。

图 1. Intel Xe 架构 SG1 芯片

Package Dimensions	28mm x 34mm	Internal Thermal Sensor	Yes
Package Type	FCBGA	L3 Cache Size	16 MB
Thermal Package Design Power(TDP)	23 Watts	Use Condition	Server/Enterprise
Graphics Engine Operating Frequency	Boost : 1.1GHz	Supply Life	Extended Supply Life
	Base : 0.9GHz	Memory	Operating Freq(max) : 2133 MHz / 4267 MT/s
CPU Interface	PCIe Gen3 X16		Configuration Type : 128-bit wide, 8 GB LPDDR4 68.25 GB/s
EUs	96		

图 2. Intel SG1 芯片规格参数

每张 XG310 上集成了 4 个 SG1 芯片，每个 SG1 处理器配置了 8GB DDR4 的内存，用于数据的处理。

图 3. H3C XG310

H3C XG310 作为一张接口为 PCIe3.0 X16 的 3/4 长、全高单宽卡，在规格上可以兼容市场上绝大多数型号的 2U 机架式服务器，无需对硬件进行重新设计；150W 的功耗，则需要电源线连接主板对 XG310 进行单独供电。

SG1 x4 Card Form Factor	
card Design	4 SoCs per Card LPDDR4 LPDDR4 LPDDR4 LPDDR4 SG1 SG1 SG1 SG1 PCIe Switch
Card TDP	150W
GPUs Per card	4
PCIe interface	Host to switch : Gen3x16 Switch to each GPU : Gen3x8
Thermal solution	Passive
Dimensions	¾ length, full height , single width

SG1 x4 Card Details	
Memory	LPDDR4X
Max memory speed	4267MT/s
Memory capacity	8GB per GPU , 32GB total
Memory bus width	128 bits
Platform Software	Host BIOS-Default Production BIOS will need Upgrading SG1-M IFWI Telemetry-MCU on card
Supported OS	Ubuntu, Centos, Debian OS
Channel	ODM : H3C Goal up to 4 Cards per 2U server

图 4. H3C XG310 规格参数

H3C XG310 产品优势

H3C XG310 具有较高的视频转码密度和均衡的图形转码性能，在高转码密度的短视频处理场景和高应用部署密度的云游戏场景中有无可比拟的优势。

如图 5 所示，这是 XG310 在 HEVC（High Efficiency Video Coding）转码场景下的性能数据，单张 XG310 在 Quality、Balanced、Performance 模式下最多可以支持 60、72、140 路的 1080p30 码流和 32、16、12 路的 4Kp30 码流。

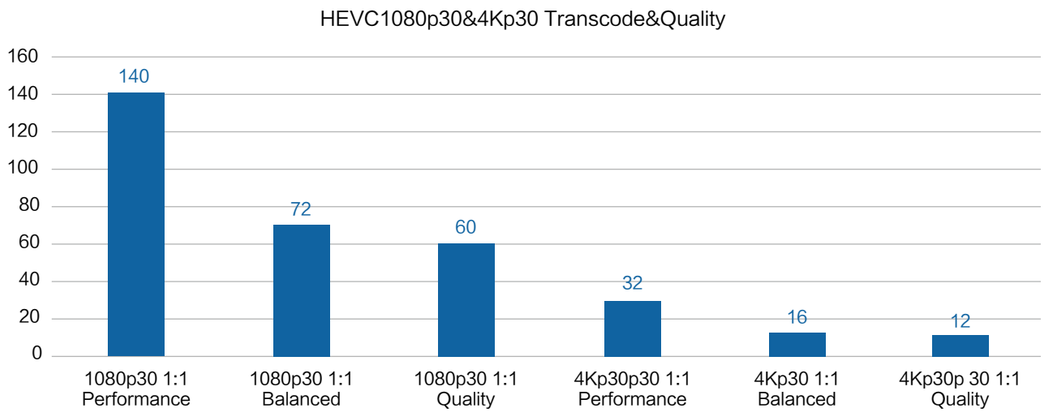


图 5. H3C XG310 HEVC 1080p30&4Kp30 Transcode&Quality

在 AVC（Advanced Video Coding）转码场景下，单张 XG310 在 Quality、Balanced、Performance 模式下最多可以支持 36、60、80 路的 1080p30 码流和 20、20、12 路的 4Kp30 码流。

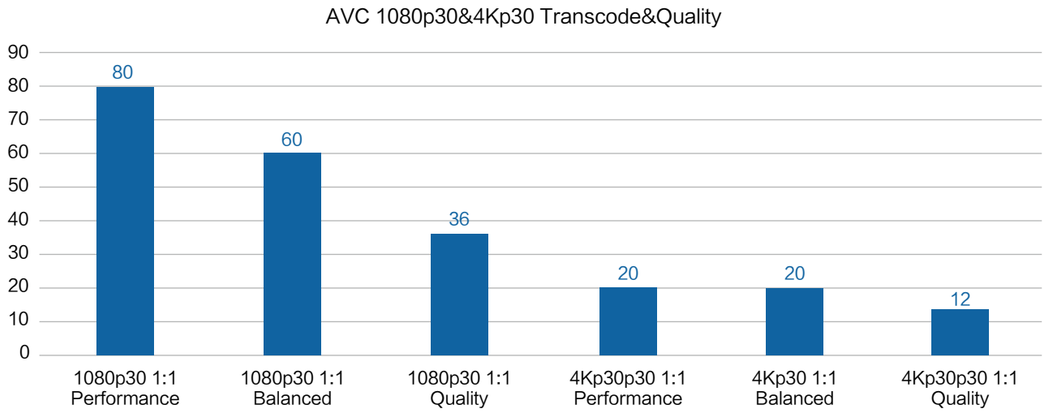
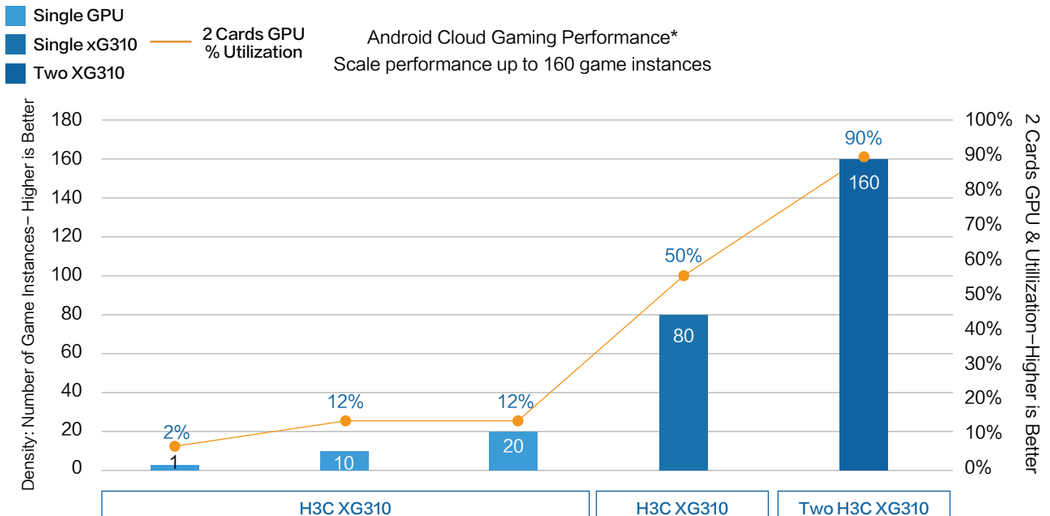


图 6. H3C XG310 AVC 1080p30&4Kp30 Transcode&Qual

如图 7 所示，这是 XG310 在某款小游戏场景下的性能表现，单颗 SG1 处理器最多可以支持 20 个实例，1 张 XG310 最多可以支持 80 个实例，2 张 XG310 最多可以支持 160 路实例。



*Measurement game Riptide

图 7. H3C XG310 Android Cloud Gaming Performance

此外，XG310 在大型游戏场景下的性能表现依旧不俗，根据客户实测，某国民现象级手游 1V1 对战场景下，通过 1 张 XG310 可支持 60 个实例，2 张 XG310 最多可支持 120 个实例。

从上述数据可以看到，随着 XG310 数量的增多，游戏实例也呈现出线性扩展的趋势，没有因为效率低下或系统瓶颈而丢失任何流。客户可以在 1 台服务器中灵活选择插卡数量，以匹配实际业务需求。

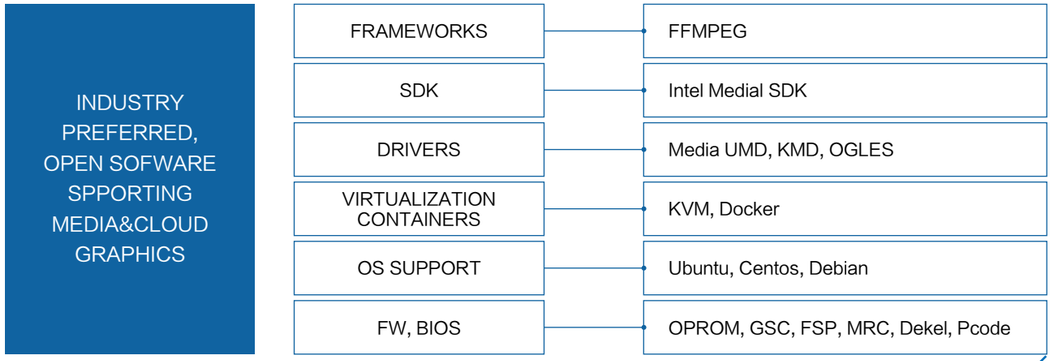


图 8. H3C XG310 软件栈

为了让客户业务更方便地使用 XG310，最大发挥其性能优势，Intel Media SDK 集成了 FFMPEG 框架，原生可以支持 FFMPEG 4.1 及 FFMPEG 4.2。

Intel Media SDK–Overview

Deliver Fast, Efficient, High Quality Video Transcoding

WHAT IT IS	Acomprehensive development software development kit for media/video applications • An API that enables real–time, hardware–accelerated video encoding, decoding, and processing • Decode and encode video using HEVC and AVC • Supports multiple widely used video preprocessing filters like Color Conversion, Scaling, De–interlacing, De–noising, Composition, Alpha Blending and more.....
WHO NEEDS IT?	• Cloud Service providers • Comms Service Providers • Media&Entertainment • Software Vendors • Government,Academia,Science • System Integrators • OEM/ODM

图 9. H3C XG310 Intel Media SDK

在虚拟化支持方面，每颗 SG1 处理器可以支持 1 个 KVM 虚拟机，一张 H3C XG310 最多可以支持 4 个 KVM 虚拟机，从而满足 CSP（Cloud Service Provider）计算虚拟化的需求。

此外，客户可以在 KVM 虚拟机上部署 Docker 环境来支持原生安卓应用，从而实现安卓业务云化，提高服务密度，降低 TCO 成本。

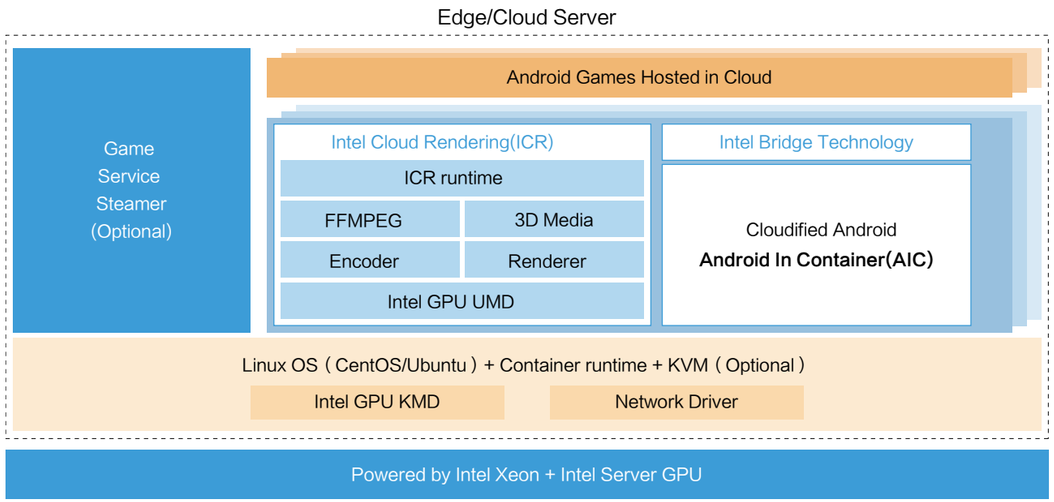


图 10. H3C XG310 安卓云化方案

总结

当前 H3C 2U 机架式服务器 R4900 G3、R4900 G5 上均进行了 XG310 的适配，硬件规格上最多可支持四张 XG310。

H3C XG310 增加了服务器的流密度，结合 H3C Uniserver R4900 系列服务器 2U1GPU/2U2GPU/2U4GPU 的灵活配置，可极大减少基础设施的成本支出。而基于 Intel SG1 芯片设计的 H3C XG310 GPU 和 Intel Xeon® 可扩展处理器的结合使用，可以在单服务器节点流密度提升的同时，确保高可靠和高可用，从而降低客户总体拥有成本（TCO）。

当前，H3C XG310 在多家大型互联网企业已有应用案例，为互联网客户云游戏和流媒体业务保驾护航。