大数据的信息基础设施

Clos 网络拓扑

陈一帅

yschen@bjtu.edu.cn

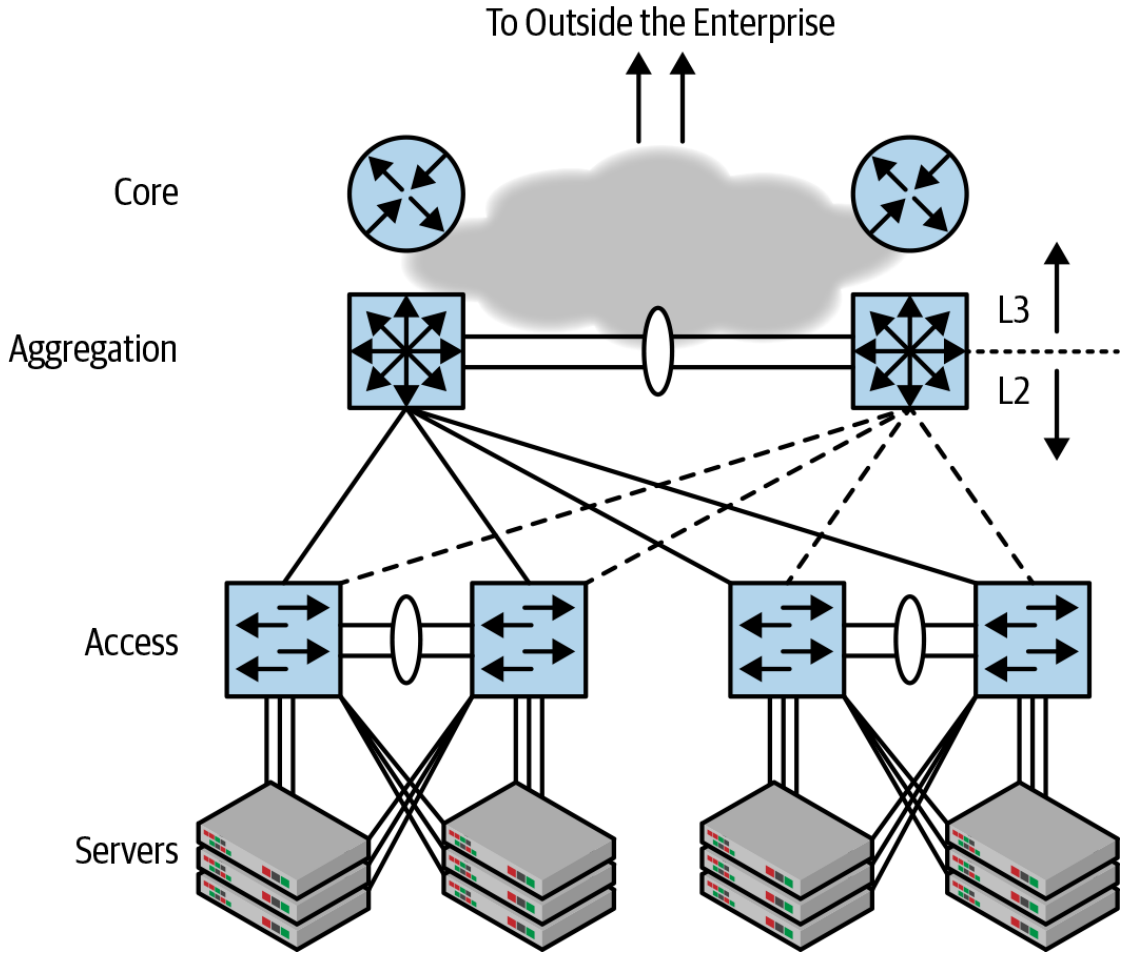
北京交通大学电子信息工程学院

内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

传统三层网络拓扑

- Access-Aggregation-Core



三层网络实际中遇到的难题

VLAN 配置难题

- VLAN 终止于聚合交换机的桥接和路由边界
  - 不能跨越多个聚合交换机，因为它们通过 L3 连接
- 在两对不同的聚合交换机之间不能存在相同的 VLAN
- VLAN 配置很费力
  - 不够灵活，无法让网络工程师根据客户需求将任何可用的空闲端口分配给 VLAN
  - 这意味着网络设计人员必须根据所需的端口数量仔细规划虚拟网络的增长

ARP 负荷问题

- 两个 Aggregate 交换机必须响应大量的 ARP
  - Windows Vista 将默认的 ARP 刷新计时器从一两分钟降低到了 15 秒，以符合 RFC 4861（“ IP 版本 6 的邻居发现”）标准
  - 该 ARP 刷新的频率如此之高，以至于它们带来已广泛部署的聚合交换机很大的问题
  - 过多的 ARP 导致 CPU 阻塞，以致其他控制协议失败，从而导致整个网络崩溃
- 随着虚拟机和容器形式的虚拟端点的出现，此问题变得成倍恶化，因为聚合交换机必须处理的端点数量增加了，即使没有增加这些框下连接的物理主机的数量。

STP 两台聚合交换机限制难题

- 解决东西向带宽日益增长的需求的一种常见方法是使用更多的聚合交换机
- 但 STP 禁止使用两个以上的聚合交换机
  - 如果由于链接和/或节点故障而导致拓扑发生更改，则超出上述限制的 STP 最终拓扑无法预测、无法使用
- STP 仅能使用两个聚合交换机的限制严重限制了网络带宽
  - 使用两个以上的聚合交换机，如果发生某些故障，生成树将变得不可预测
  - 两个聚合交换机 => 它们是瓶颈
  - 单条链路故障可以将可用带宽减少一半

容错设计复杂

- 需要对 Access-Agg-Core 网络进行精心设计，才能防止在链路失败时此类网络发生拥塞
- Agg1 和 Agg2 交换机均宣布对连接到 acc1 的子网的可达性
  - 如果 Agg1 和 acc1 之间的链接失败，Agg1 需通过 Agg1 和 Agg2 之间的链接将数据包发送到 acc1
  - 这意味着 Agg1 和 Agg2 之间的链路带宽需要仔细设计。否则，由于链接故障而发送的流量超出计划的流量，导致应用程序性能下降
  - 这使网络设计，容量规划和故障处理变得复杂

失败发生时性能问题

- 数据中心规模庞大，失败是确定的。对故障的响应至关重要
- 爆炸半径（blast radius）衡量单个故障造成的破坏程度
  - 故障越接近故障点，故障域的粒度越细，爆炸半径越小。
- Access-agg-core 模型易于出现爆破半径大的故障，例如
  - 单个链路的故障将可用带宽减半
  - 单个聚合交换机的故障使整个网络瘫痪，因为整个网络的流量带宽减少了一半

应用变化带来流量变化

- 微服务带来服务器之间的流量增长
- 南-北 vs. 东-西流量
- 以前，大部分流量都是南北
  - 数据中心服务器与外部客户端之间
- 现在趋势是服务器之间进行大数据分析的流量
  - 东西向流量
- 需要更扁平的网络
- 类似于 Fat-Tree 的拓扑

内容

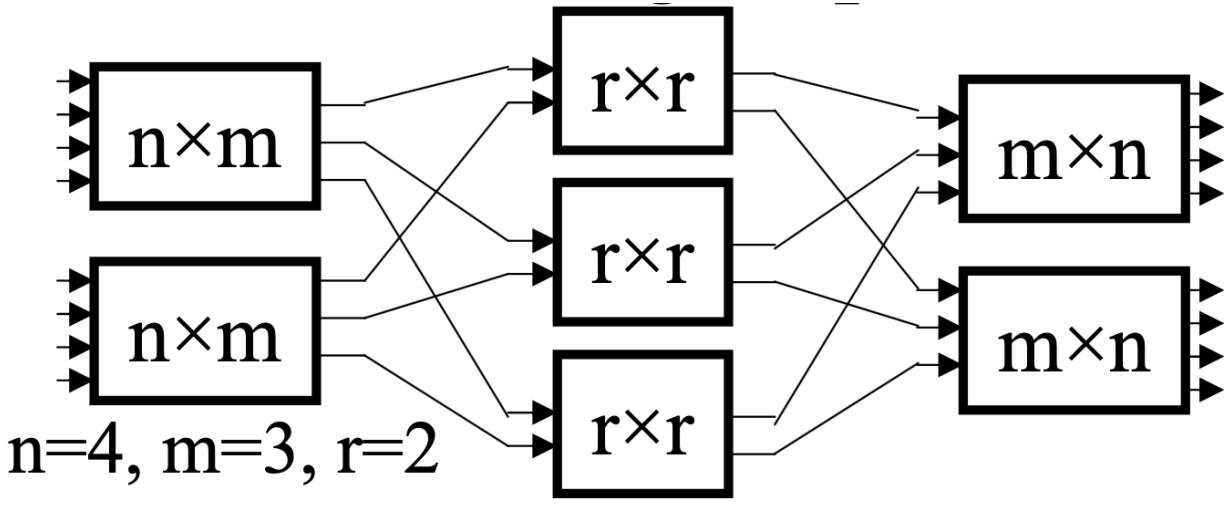
- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

交换结构

- 交换结构（switched fabric）（使用交换机的网络拓扑）的主要目标是通过使用仅具有有限数量的端口的交换机来连接大量端点（处理器或服务器）
- 通过巧妙地连接交换元件并形成拓扑，网络可以互连大量端点

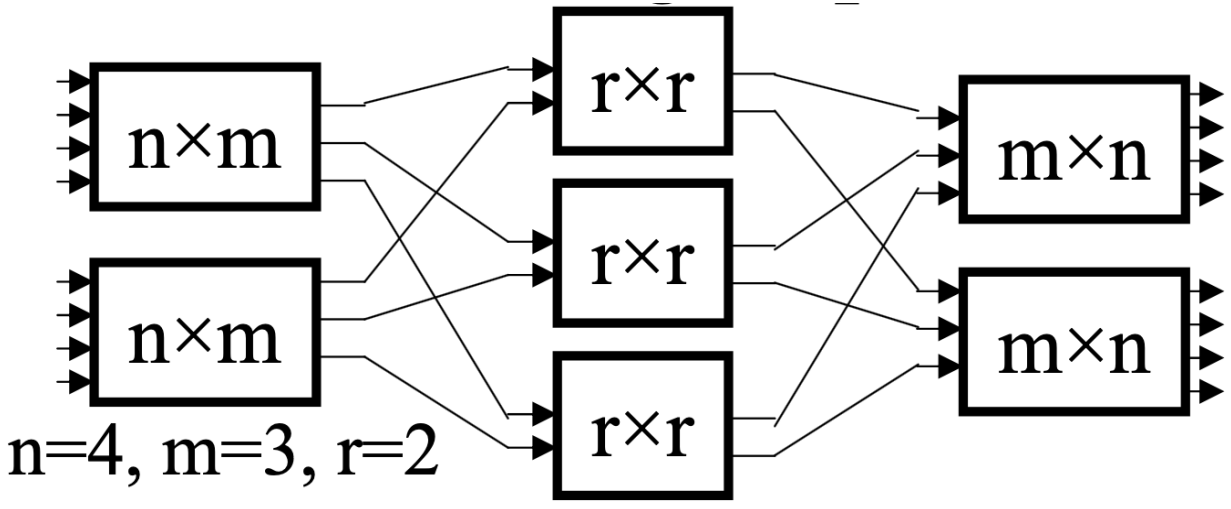
Clos 交换网络

- Charles Clos（贝尔实验室）在 1953 年提出的用于电话交换系统的多级电路交换网络
- 允许使用相对较小的交换机来构建非常大型的交换网络
  - 减少交叉点的数量



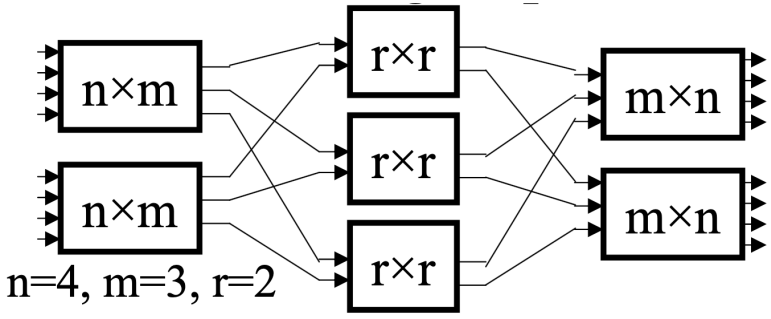
Clos 交换网络

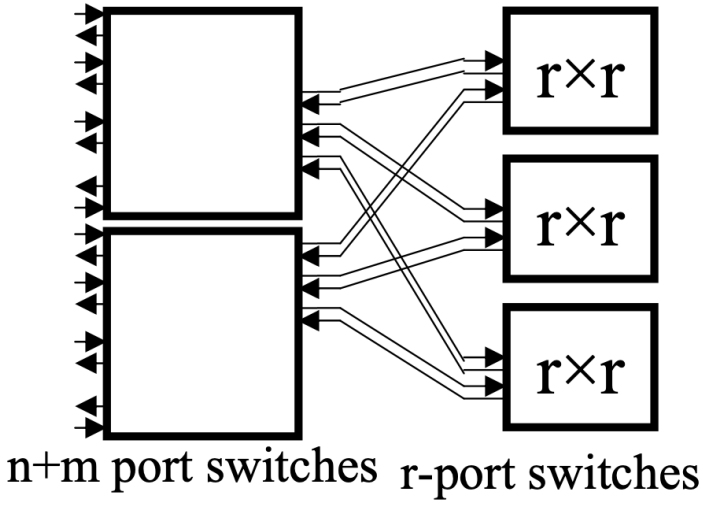
- 可以具有任意奇数级，例如 5
- 3 级 Clos（n，m，r）
  - 入口（r 个 n×m）
  - 中间（m 个 r×r）
  - 出口（r 个 m×n）



Clos 交换网络的折叠表示

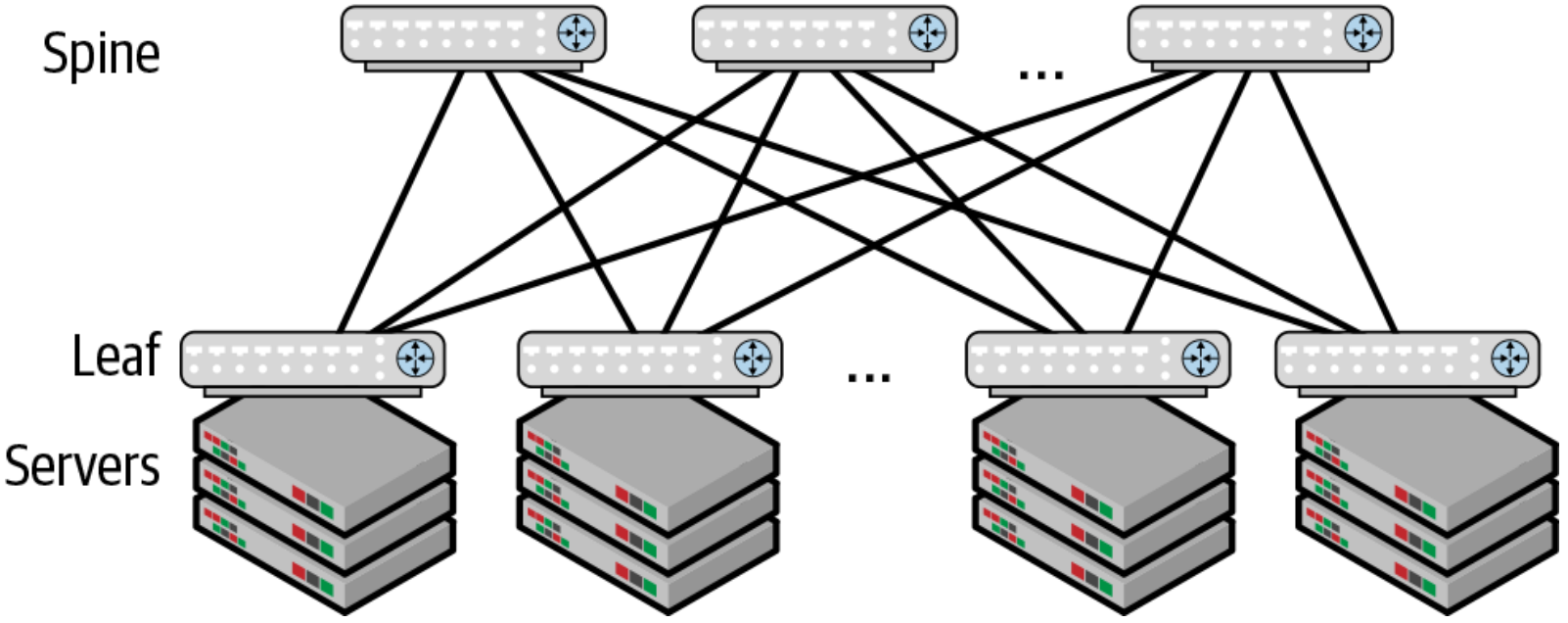
- 折叠：将输入和输出合并到一个交换机中





Clos 折叠实现

- 通常将服务器和分支交换机置于单个机架中，交换机位于机架的顶部
- 因此，通常将叶子称为机架顶部（ToR）交换机



Clos 优点

- 高容量网络
  - 任何两台服务器之间都有两条以上路径
  - 添加更多骨干交换机，增加更多路径
  - 不建议在一个叶子和一个骨干间添加更多链接以增加路径
- 除了互连功能外，所有其它协议功能都放到边缘叶子交换机
  - 比如，不像聚合交换机，骨干不负责响应终端的 ARP 请求
  - 添加更多叶子，骨干上的控制平面负载只会略微增加

Scale-In vs. Scale-out

- Scale-In 扩展型结构
  - 由于拥有横向扩展架构，Clos 的增长非常稳定
  - 可以通过添加更多的叶子和服务器，增加网络支持的工作量
  - 仅将骨干交换机用于增加叶子交换机之间的带宽
  - 这样的体系结构称为 Scale-In 扩展体系结构
- 相反，在 access-agg 拓扑中，网络的扩展是通过增强聚合交换机的 CPU 来提供的
  - Scale-out

Clos 下的网络路由

- 基于 STP 的聚合核心网络的基本限制是其仅支持两个聚合交换机
- Clos 不用生成树协议（STP）用作交换机互连控制协议
- 桥接仅在边缘（即在单个机架内）支持
- 跨机架桥接使用更现代的网络虚拟化解决方案，如虚拟可扩展局域网（VXLAN）

流管理

- ECMP
  - 流哈希（Flow hashing）
  - 但还是会出现 Elephant flow
- 上下行流量不均衡
  - Oversubscription

上下行流量均衡

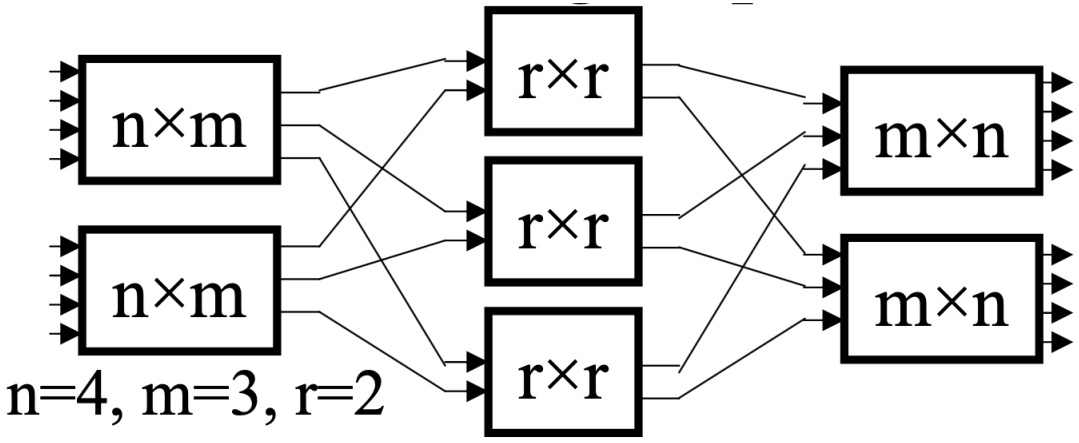
- Oversubscription
- 一个交换机的下行链路与上行链路带宽之比
- 叶子节点
  - 下行链路面向服务器，上行链路面向主干
  - 1：1 意味着总下行链路带宽等于总上行链路带宽
  - 许多网络使用 2：1 或 4：1
- 中间节点
  - 大多数数据中心都确保更高层节点 1：1

非阻塞网络

- 1：1 Oversubscription 网络也称为无阻塞网络
  - 非竞争性的：从一个下行链路到上行链路的流量不会与来自其他下行链路的流量竞争
- 可重排的无阻塞
  - 根据流量模式，流哈希能够区分不同的流
  - 可通过重新排列来自同一下行链路的不同下行链路的流以使用其他上行链路，使网络再次畅通无阻

Clos 交换网络性能

- 如果 m > n，则可重排（Rearrangeably）无阻塞
- 如果 m > 2n-1，则 Strict-sense 无阻塞
  - 现有呼叫可不受影响



Clos 网络练习

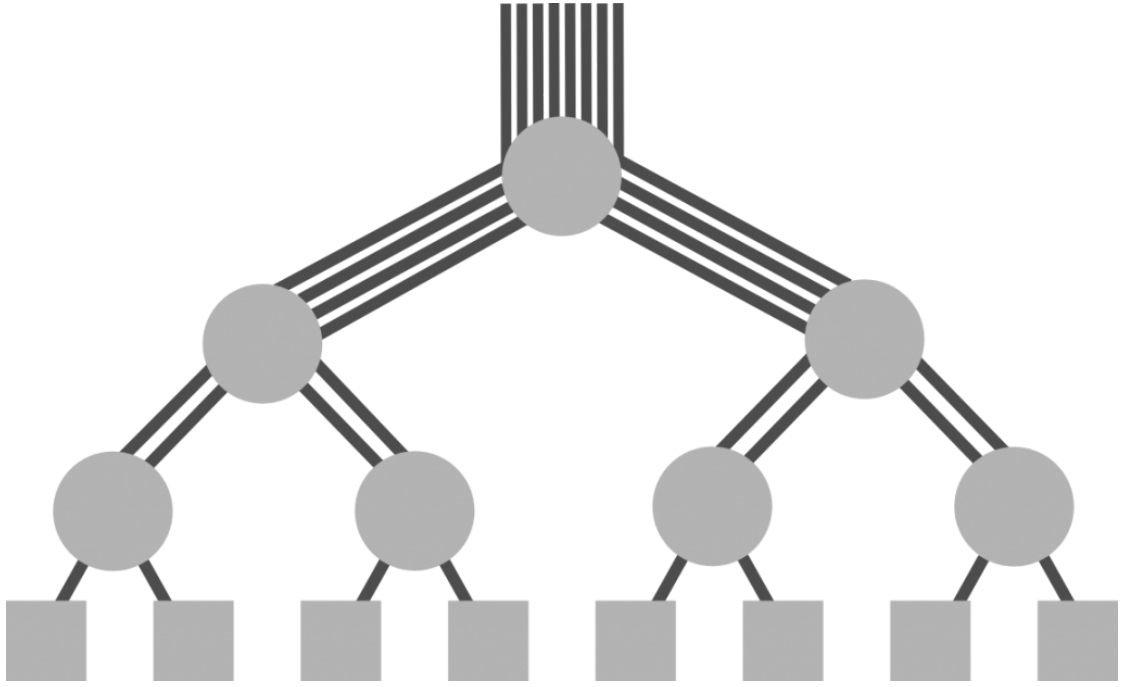
- 绘制一个三级 clos（4，5，3）拓扑及其折叠版本
- n = 4, m = 5, r = 3

内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

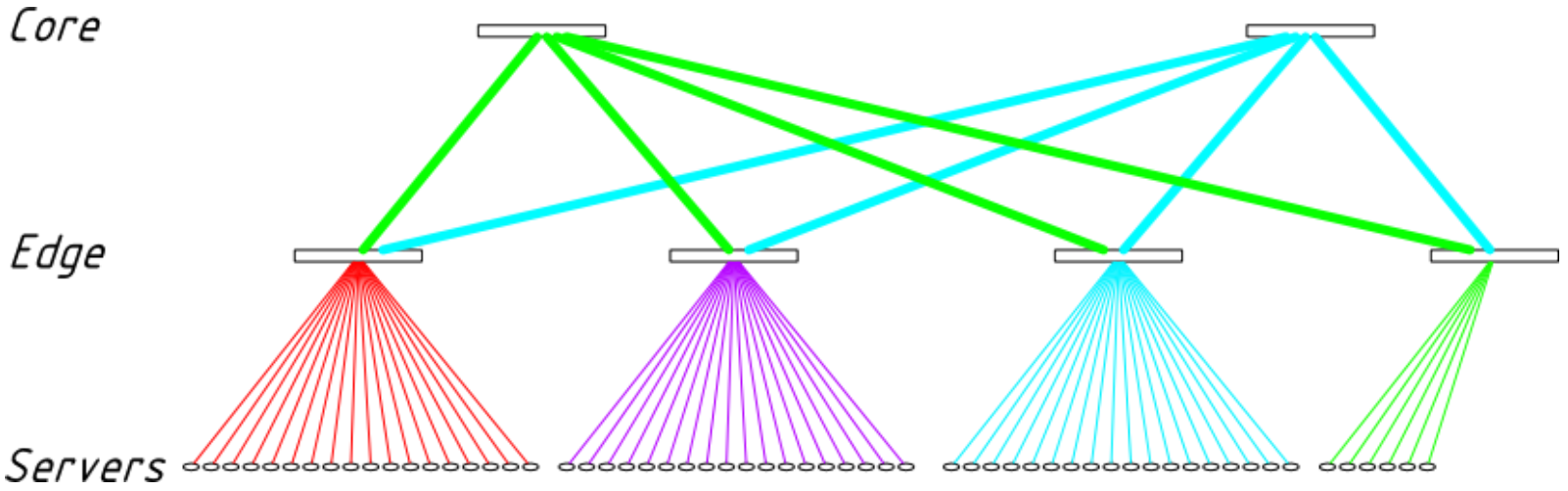
Fat-tree

- Charles E. Leiserson 1985 年提出
- 任何节点，下行链接数量等于上行链接数量
- 因此，树的根节点链路最多



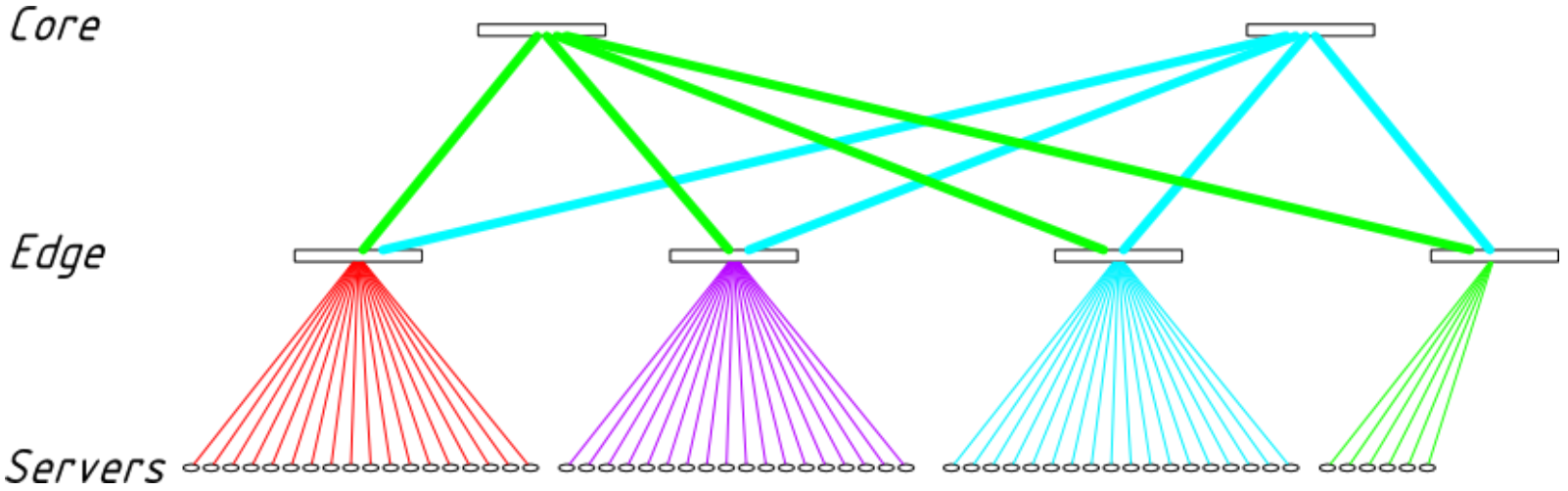
实现

- 采用 Clos 交换结构
- 各级使用相同的交换机
- 边缘交换机，上行端口数目等于下行端口数
  - 达到了 Fat-tree 的效果



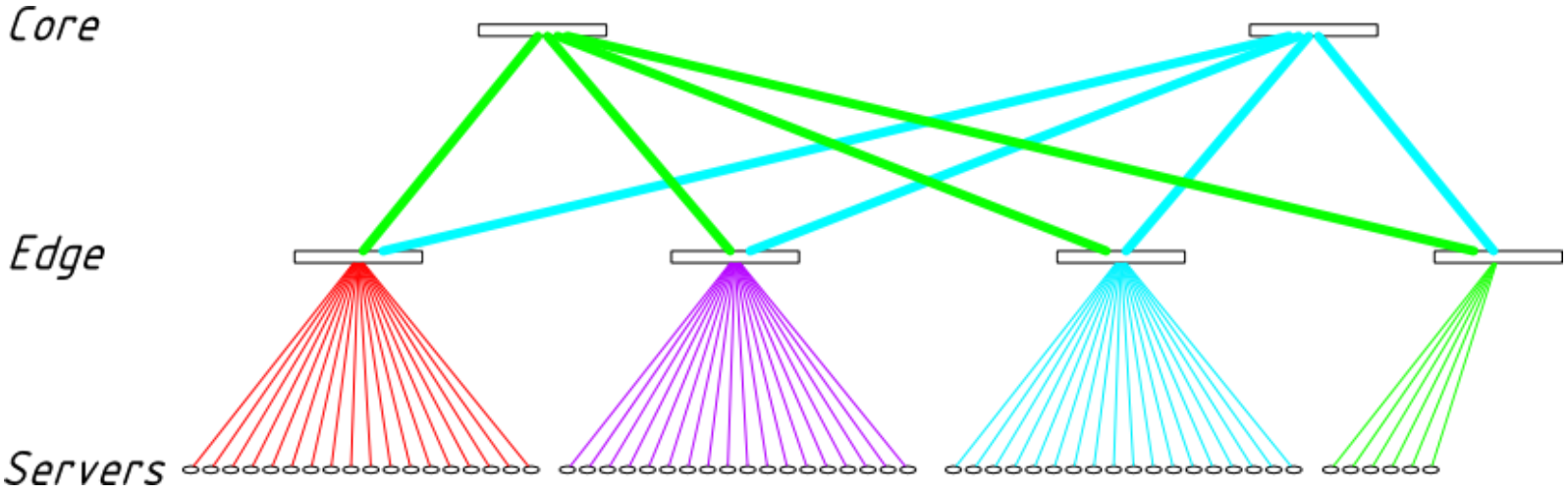
示例

- 6 个相同的 36 端口交换机
  - 4 个作为边缘交换机，2 个作为核心交换机
- 边缘交换机 18 个端口连服务器
  - 剩余 18 个端口分两组，每组 9 个连核心交换机



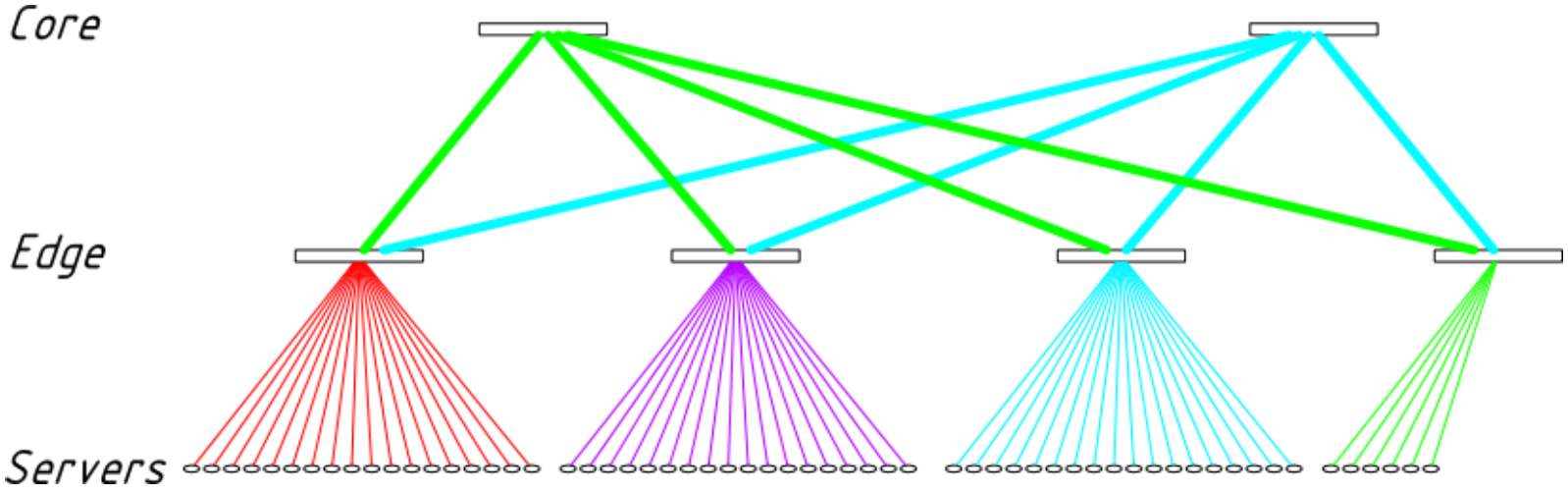
示例

- 达到 Fat-tree 的效果
  - 边缘交换机连服务器数量等于到父节点链接数量：18 个
- 与原始 Fat-tree 的区别
  - 中间交换机有多个父节点（在这种情况下为 2 个）
  - 原始 Fat-tree 中每个中间节点只有一个父节点



性能

- 设每端口速率为 1Gbps
- 路由选择时使用 ECMP
- 结果：任何两个服务器间吞吐量 1 Gbps
- 缺点：布线复杂

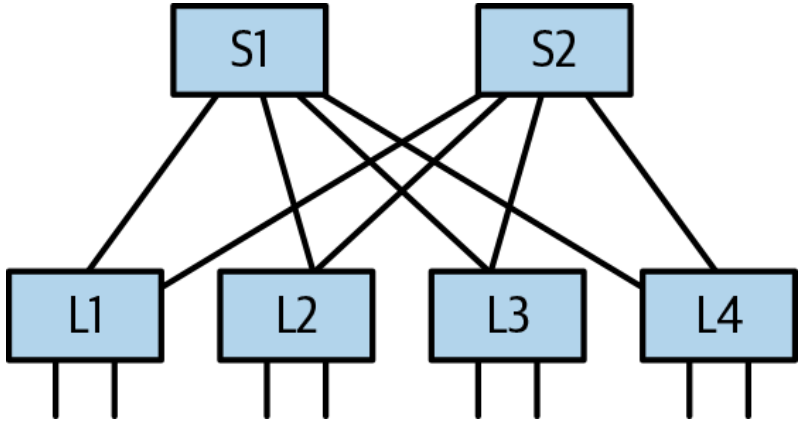


内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

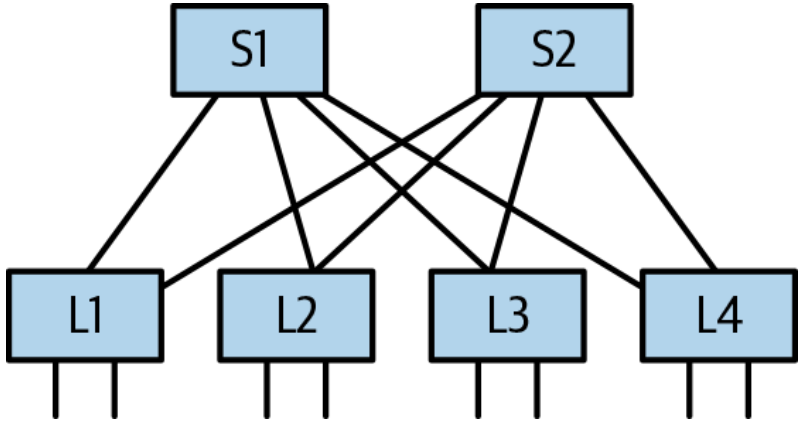
容量

- 求骨干交换机最大数量
- 都使用 n 端口交换机
- 1：1 oversubscription
  - 叶子一半端口连服务器，另一半连骨干交换机
  - n 端口交换机，有 n/2 个连骨干的端口
- 最多连 n/2 个骨干交换机



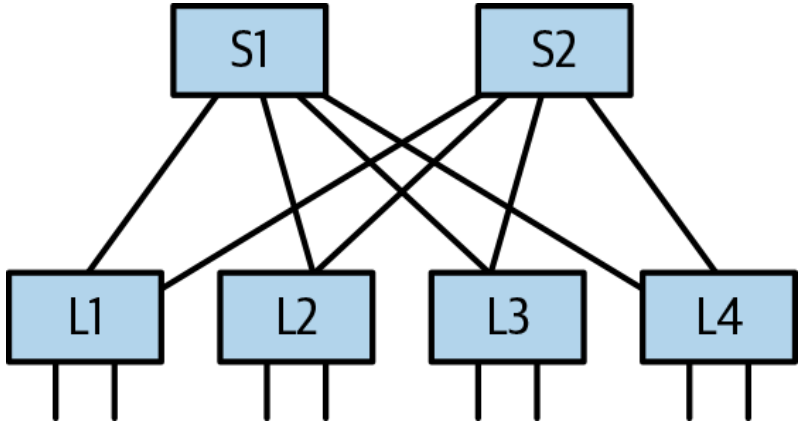
服务器数

- Clos 拓扑中可以连接的服务器的最大数量为 $n^2/2$
  - 最多 n/2 个骨干交换机
  - 把 n 个叶子交换机连在一起
  - 每个叶子节点连 n/2 个服务器
  - 需 $n + n/2$ 个交换机



服务器数

- n = 64
  - 96 个交换机，连接 2,048 个服务器
- n = 128（两倍）
  - 192 台交换机，8,192 台服务器（四倍）



练习

- 使用 4 端口交换机绘制最大的 Fat-tree 拓扑
  - 假设每个服务器都连接到一个叶子交换机，而叶子交换机被多宿主到骨干交换机。没有核心层。
  - 可以连接多少台服务器?
  - 需要多少个交换机?
- 使用 64 端口交换机
  - 最多可以连接多少台服务器?
  - 此时需要多少个交换机?

内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

两级 Fat-Tree 拓扑的优势

- 能够很好地处理东-西向流量（横向流量）
- 叶子和骨干采用同样的设备
  - 维护和更换更容易
- L2 转发仅在每个机架中的叶子交换机使用
  - 机架间路由用新协议 VXLAN
- 叶子可以相同成本通过任何骨干到达另一片叶子
  - ECMP 简化了路由
- 流哈希选择骨干交换机
  - 流的所有包都用相同路径发送，避免乱序到达
  - 流 = {源 IP，目标 IP，L4 协议，源端口，目标端口}

优势

- 一切都改变了
  - 考虑故障
  - 购买交换机
  - 库存管理
  - 网络管理
- 更简单
- 更节省

实际中机架服务器数量限制

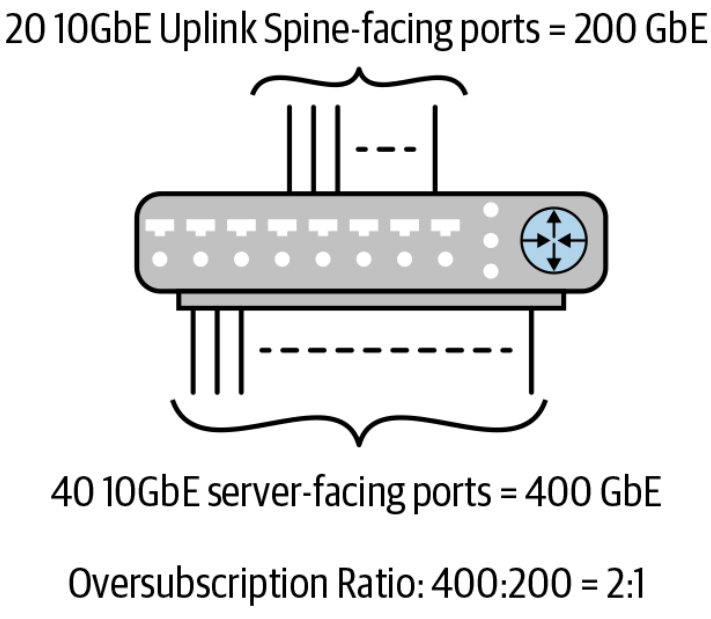
- 微服务器技术能够将 96 台服务器装入一个机架
- 但常见是每个机架 20 或 40 台服务器
- 机架空间限制：每个机架 40 个服务器
- 冷却和电源限制：
  - 电源限制每个机架最多只能支持 20 台服务器
  - 图形处理单元（GPU）更增加功耗，进一步减少数量

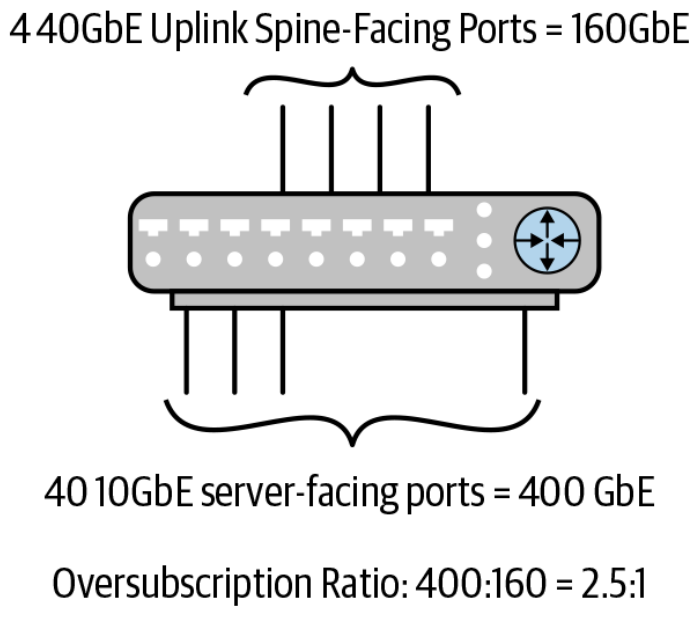
实际中端口带宽配置

- 目前的交换芯片支持 64-128 个 100GE 端口
- 更高速连骨干交换机（ISL：inter-switch link）
- 用更少骨干交换机获得相同 Oversubscription
- 叶子交换机常用配置
  - 6 个 100GbE 端口 + 48 个 10GbE 端口
  - 8 个 100GbE 端口 + 48 个 25GbE 端口
  - 100G 的连骨干，低速率的连服务器
- 骨干交换机常用配置
  - 32-128 个 100GbE 端口

例: 配置 1

- 每个机架 40 个服务器，交换机 64 个 10GbE 端口
- 将 20 个空闲端口用于上行，2:1 oversubscription
- 两层 Clos 中能够连接 64×40 = 2,560 个服务器

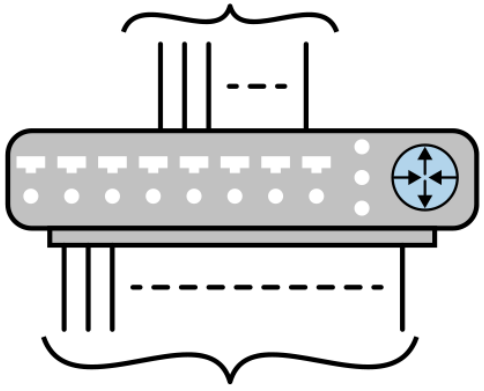




例: 配置 2

- 通常将 24 个 10GbE 端口组合为 6 个 40GbE 端口
  - 使用其中 4 个，连到四个主干交换机
- 400:160 = 2.5:1 oversubscription
  - 10G 的 40 个端口为 400，而 40G 的 4 个端口为 160
  - 完全满足小型数据中心的要求

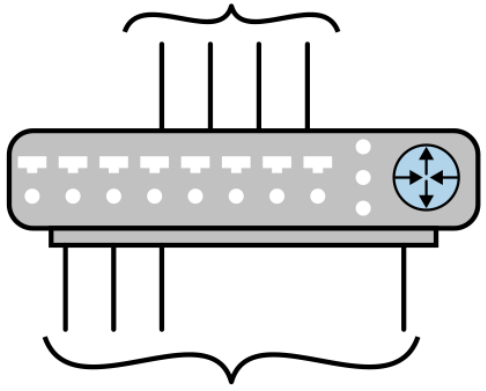
20 10GbE Uplink Spine-facing ports = 200 GbE



40 10GbE server-facing ports = 400 GbE

Oversubscription Ratio: 400:200 = 2:1

4 40GbE Uplink Spine-Facing Ports = 160GbE



40 10GbE server-facing ports = 400 GbE

Oversubscription Ratio: 400:160 = 2.5:1

骨干交换机故障设计

- 大规模提供商使用多达 16 或 32 个骨干交换机
- 最少用 4 个
- 容忍错误
  - 16 个骨干交换机，单个骨干交换机或链接丢失仅导致总带宽减少 1/16
  - 4 个骨干交换机，损失四分之一

叶子交换机故障设计

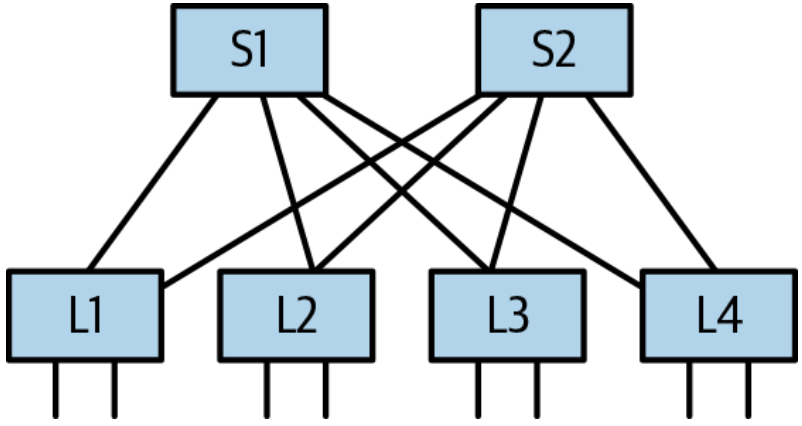
- 叶子的丢失会影响与其连接的所有服务器
- 大型数据中心中，这没有问题，因为它们有成千上万个机架，因此丢失的服务器上的工作可以简单地放弃并重新安排在其他服务器上
- 较小的数据中心在每个机架中放置两个叶子交换机，将每个机架中的服务器连接到两个叶子交换机上

内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

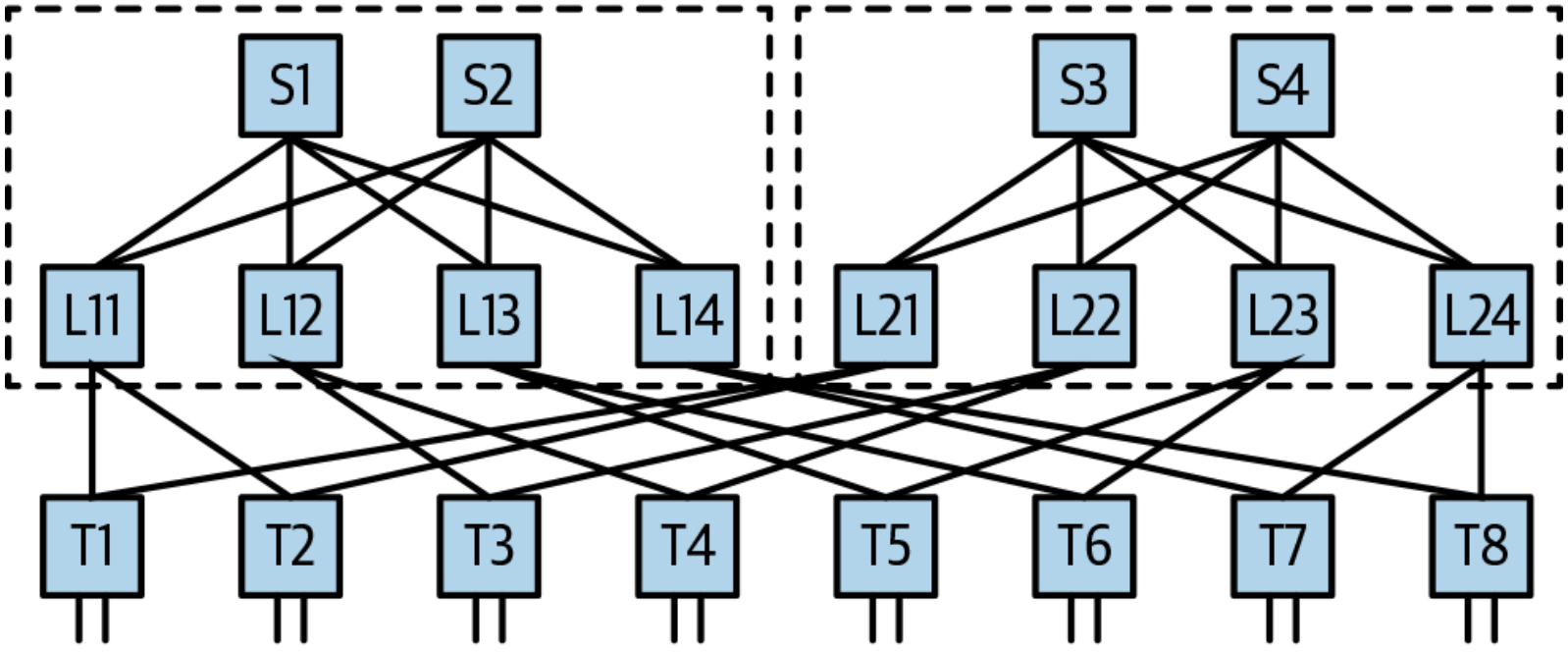
多级（Tier) Clos 设计

- 将两级扩展到三级
- 两种方案
  - Facebook 方案
  - 微软，亚马逊方案
- 假设 1:1 oversubscription ratio



方案 1：Virtual chassis

- Facebook 方案
- 叶子下面接交换机
- 相当于一个虚拟 8 端口骨干交换机



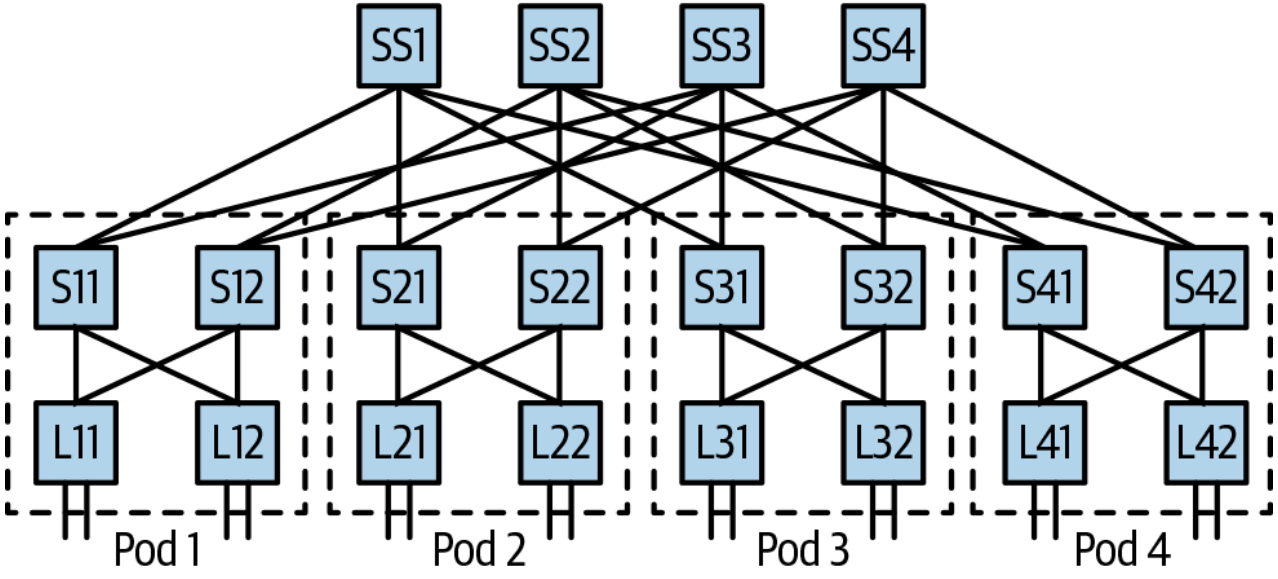
(b) Virtual chassis-based three-tier Clos using four-port switches

方案 2：Pod 模型

- 微软，亚马逊方案
- 上面再加一层
- 相当于一个虚拟 8 端口 Pod 叶子节点

三级 Clos 方案

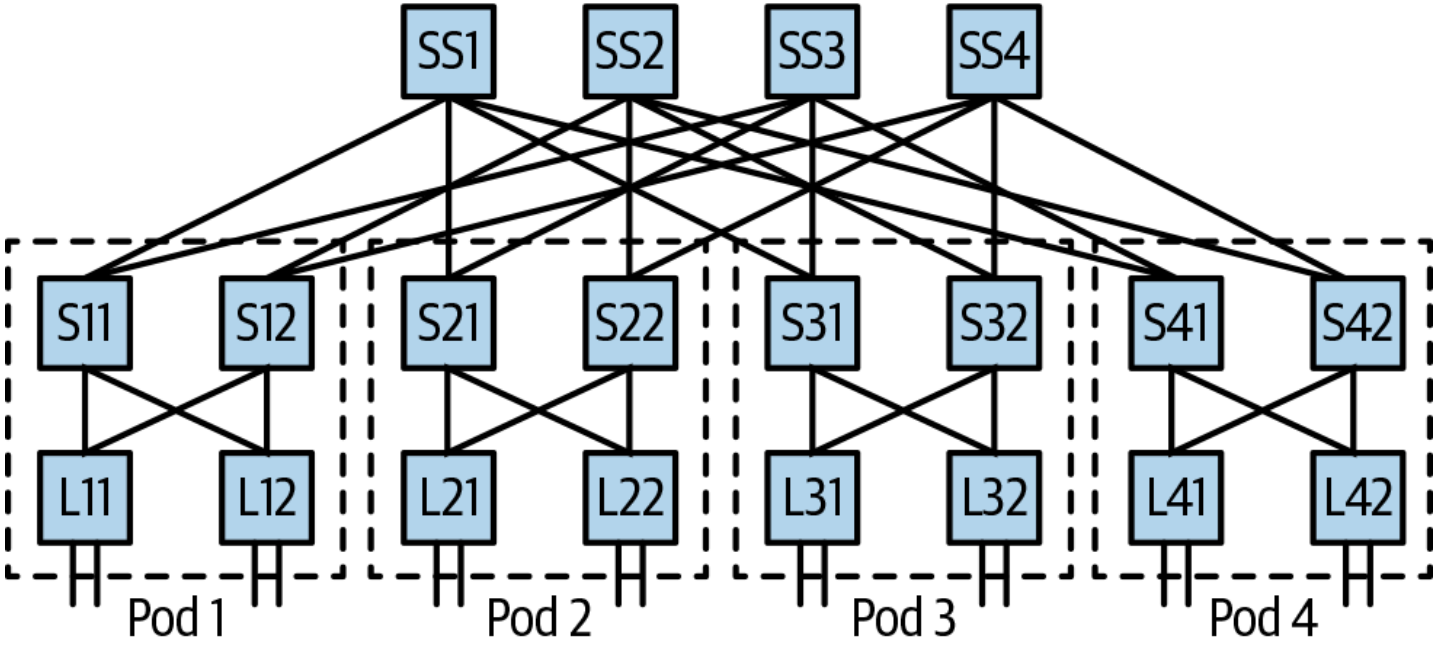
- 维持了 1：1 的 oversubscription
- 在拓扑每个级别都获得了两个 ECMP



(c) Pod-based three-tier Clos using four-port switches

三级 Clos 方案

- 问题：20 个交换机，支持 16 个服务器?



(c) Pod-based three-tier Clos using four-port switches

服务器数

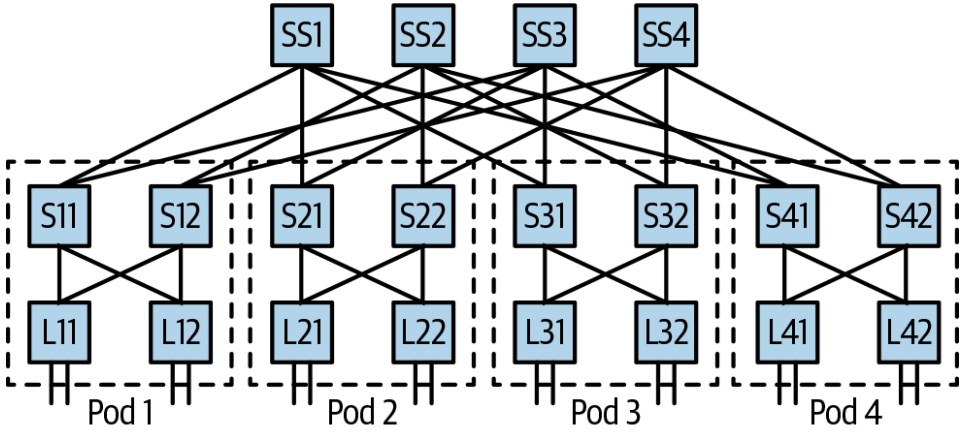
- n 端口交换机组成的三层 Clos 网络可以支持服务器数量为 $n^3/4$
  - 对两种方式都成立
- n = 64, 可支持服务器总数为 $64^3/4 = 65,536$
  - 2 层 Clos 可支持服务器数量为 2,048
  - 这是一个很大的提高
- n = 128，可支持 524,288 服务器

交换机数

- 三层 Clos 拓扑所需交换机总数为 $n + n^2$。
- n = 64，需 64 +（64^2）= 4,160 个交换机

交换机数推导：使用 Pod 模型

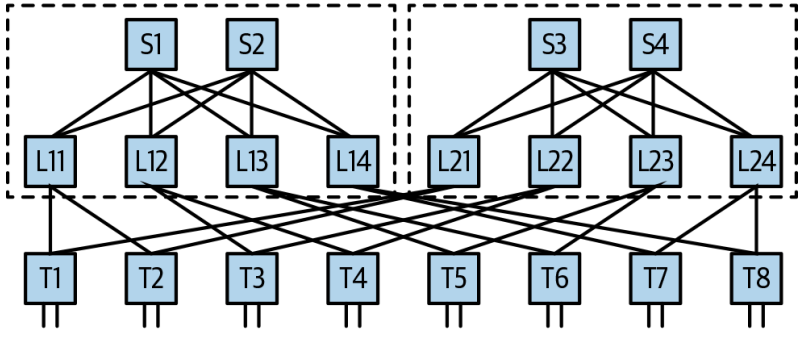
- 每个 Pod 中 n 个交换机
  - 骨干为端口数一半（n / 2）
  - 叶子数量为端口数一半（n / 2）
- 最多 n 个超级主干交换机，连接 n 个 Pod
- 总共 n +（n×n）= $n + n^2$ 个交换机



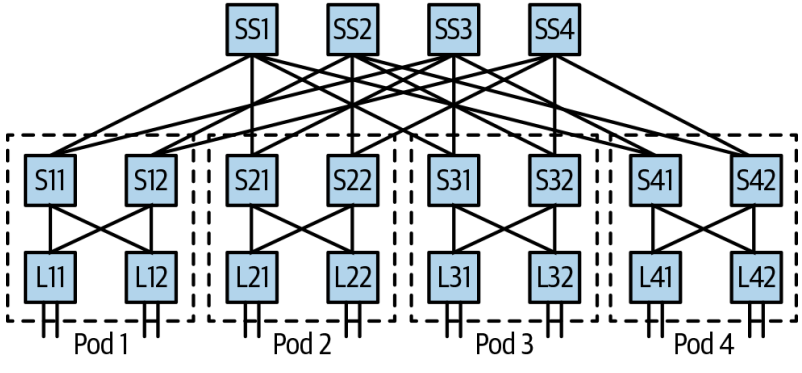
(c) Pod-based three-tier Clos using four-port switches

延时

- 3 跳或 5 跳
- Virtual chassis 更均匀



(b) Virtual chassis-based three-tier Clos using four-port switches



(c) Pod-based three-tier Clos using four-port switches

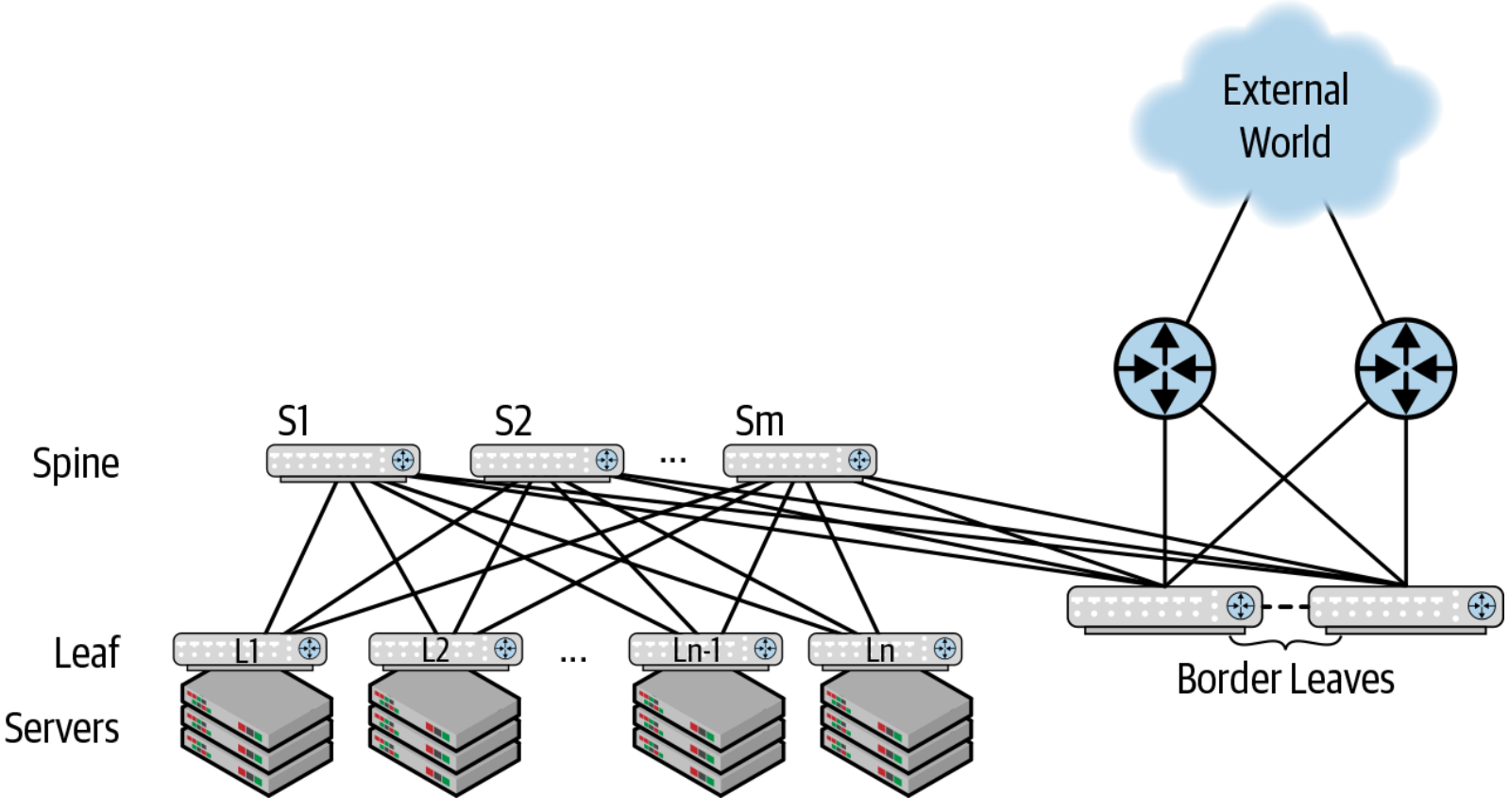
应用场合

- 大多数数据中心运营商都倾向于使用 pod 模型来构建数据中心，以便增量扩展
  - Pod 模型中，如果大多数流量限制在 Pod 中，可以从较少超骨干交换机开始
  - 这在 virtual chassis 模型中不好实现
- 当数据中心运营商制定了非常清晰的扩展计划并知道不会浪费前期投资时，通常会使用虚拟机箱模型，方便实施

内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

外网连接


External
World
Spine
S1
S2
...
Sm
Leaf
L1
L2
...
Ln-1
Ln
Servers
Border Leaves

路由

- VXLAN 基于 UDP，需要 L3 路由
- 对数据中心有用的 DV 协议是 BGP
  - BGP 在数据中心最流行
  - BGP 支持 IPv4，v6，Mac 地址，流信息，组播地址，MPLS，支持最广泛，安全性也最好
  - 每个叶子一个 AS，一个 Pod 一个 AS，所有最高级交换机一个 AS，这样就可以避免包回传
  - 混合云：企业私有云和公有云互连，用 BGP 很合理
  - BGP 的安全机制最好，可以配置路由来源，过滤过来的路由广播

路由

- 对数据中心有用的 LS 协议是 OSPF 和 IS-IS
  - OSPF 也很常用，因为企业网管理员熟悉它
  - OSPF 为了扩展，加了层级，控制 LS Flooding
  - 两级：主干、非主干
  - EVPN 使用 OSPF
  - IS-IS 也支持多级，每级 300 个路由器没有问题

网络故障时的 Debug

- 很难
- Netflix 的 USE 模型
  - Utilization, Saturation, Errors
- 检查以下数据可能是有益的
  - 叶子节点的 Uplink 的利用率
  - 延时

现状

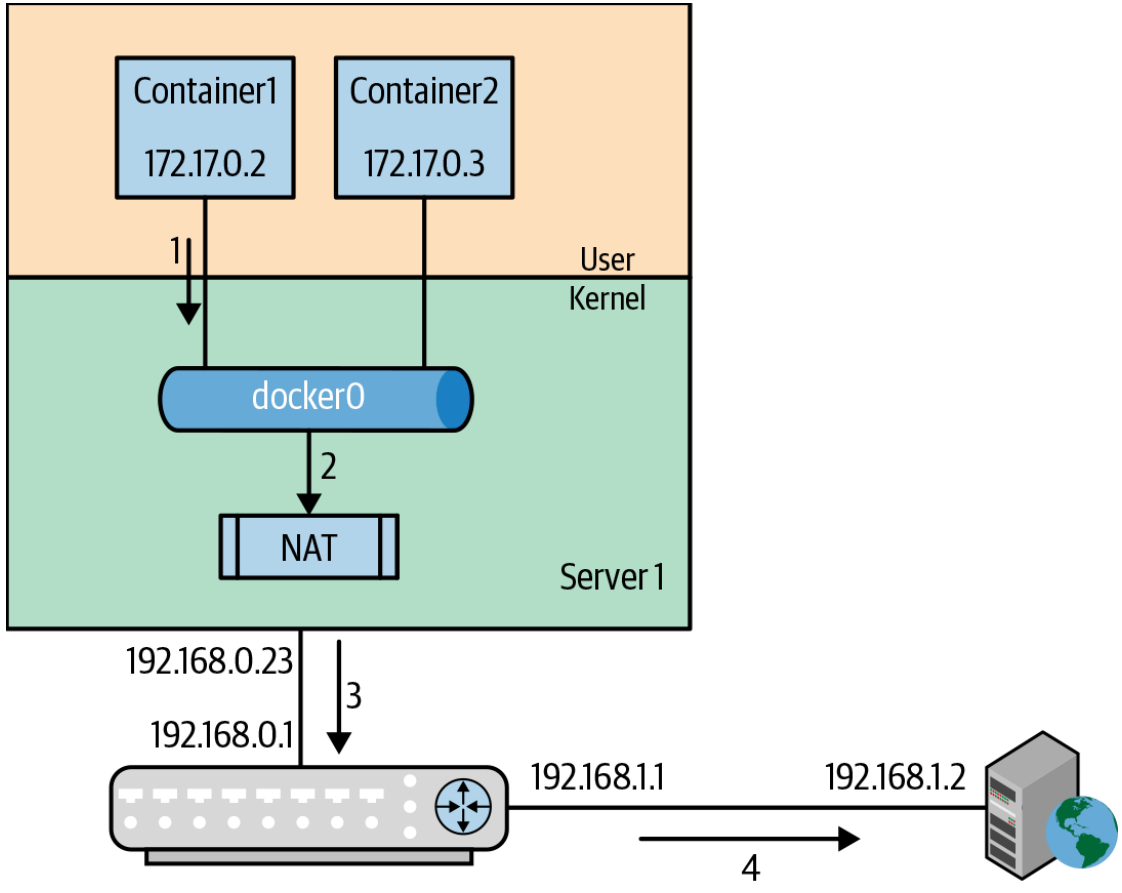
- Network bisection 带宽
  - 衡量在指定的时间段内从超级计算机的一半流向另一半的数据流量
- 谷歌 2015 年声称其 bisection 带宽超过 1 PB /s（10^{15} b/s）
  - 足以让 100,000 台服务器以 10 Gb /s 的速度交换信息

内容

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

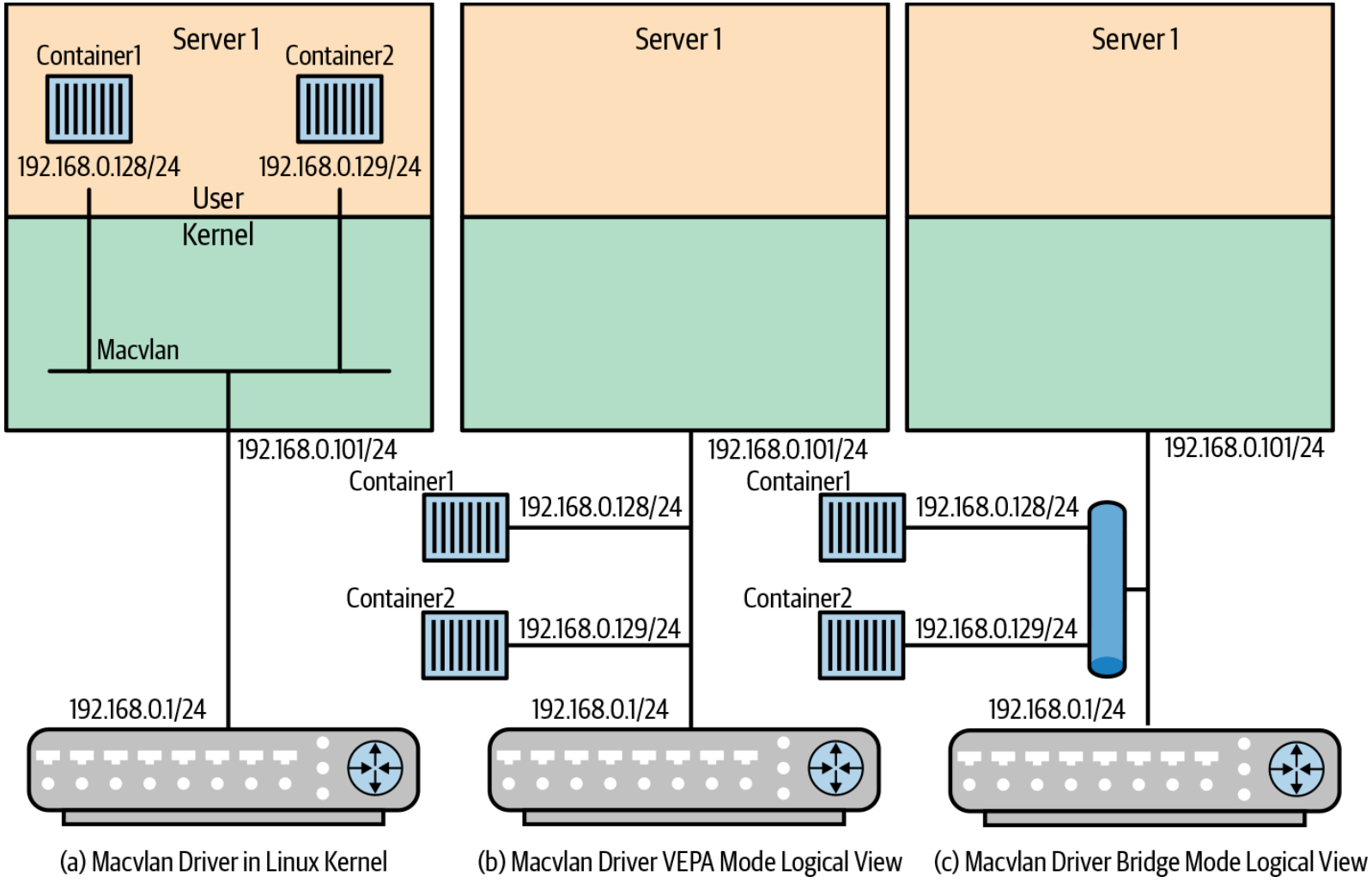
单主机

- 一个主机上的多个容器可以互相通信
  - docker0 桥，每启一个容器，会建一个 veth，从容器的 netns 绑定到 docker0 桥上



单主机

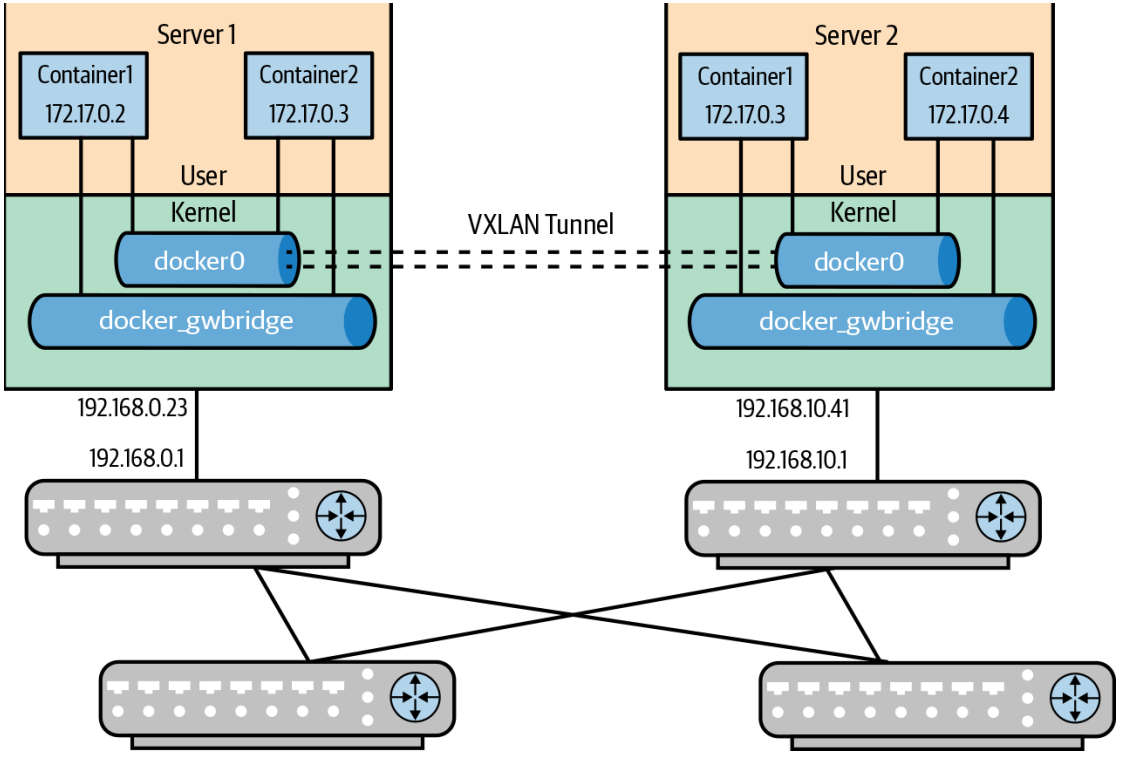
- Macvlan 直接运行在物理接口上，性能更高



(a) Macvlan Driver in Linux Kernel (b) Macvlan Driver VEPA Mode Logical View (c) Macvlan Driver Bridge Mode Logical View

多主机

- 多个主机上的多个容器互相通信
  - VXLAN L2 虚拟网络，统一地址管理

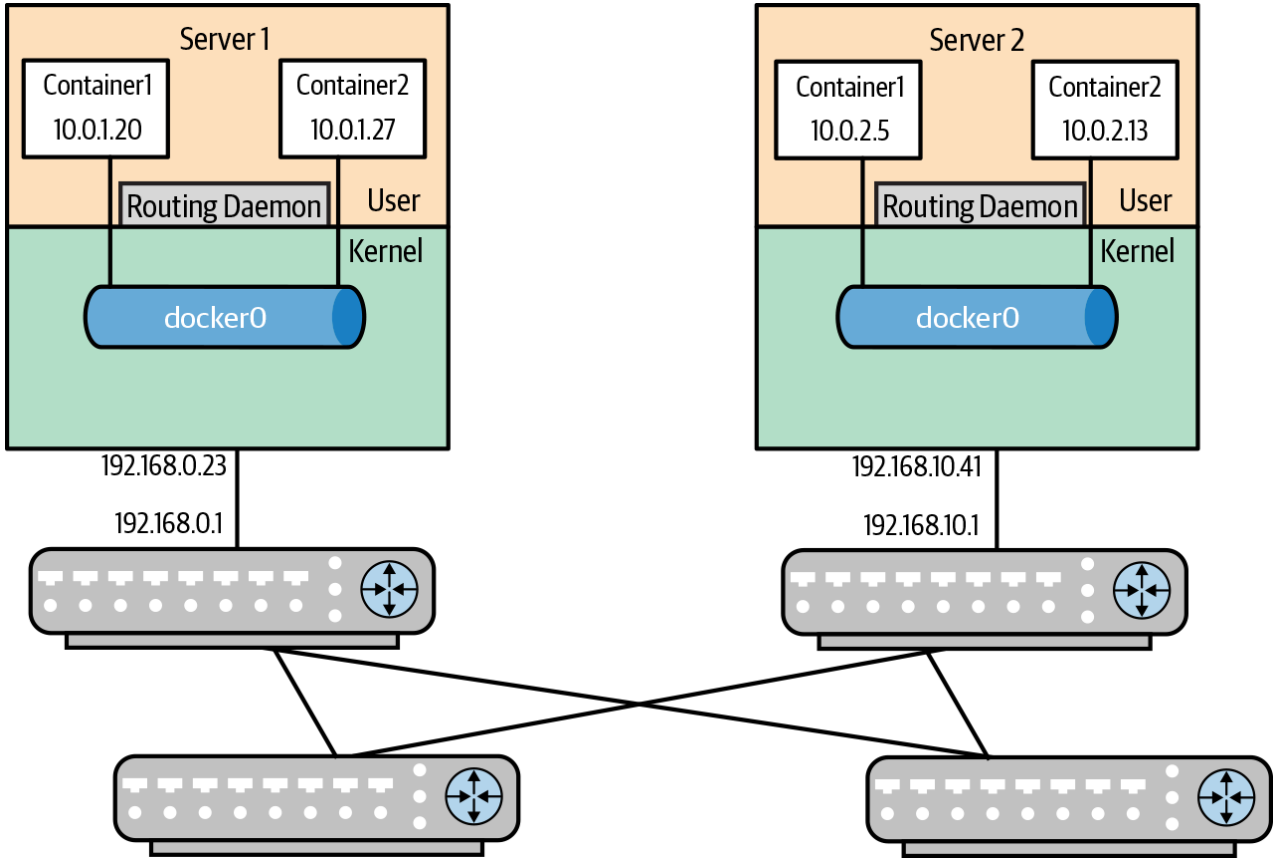


Kubernete 网络

- 用容器部署微服务
- Pod：一套容器，总是一起运行。共享统一的 namesapce 和 cgroup
- Service：一套 pods 提供一个 service，有一个名字，一个 IP 地址
- 可以根据 load 启动或者关闭或者替换 Pod

Kube-router

- 用 BGP 广播容器地址
- 用 iptables 或 IP 虚拟服务器（IPVS）做负载均衡



小结

- 传统三层网络结构的不足
- Clos 交换结构
- Fat-Tree 实现
- Clos 数学
- 实际网络规模限制与设计
- 多级 Clos 设计
- 部署和管理
- 容器网络

练习

- Clos 数学
- 三级 Clos
- 调研云计算平台采用的网络拓扑

参考

- Cloud Native Data Center Networking_ Architecture, Protocols, and Tools-2019

- A. Shieh, “Sharing the Data Center Network,” NSDI 2011, http://www.usenix.org/event/nsdi11/tech/full_papers/Shieh.pdf