中山大學 软件工程学院

SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

SSE316：云计算技术
Cloud Computing Technology

陈壮彬

软件工程学院

https://zbchern.github.io/sse316.html

数据中心网络

- 数据中心应用及流量
- 数据中心网络架构基本特点
- Fat-tree 网络架构

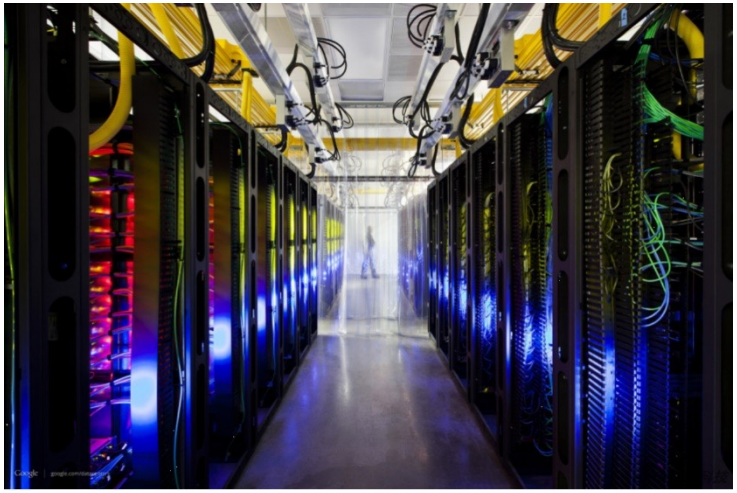
数据中心网络

- 数据中心应用及流量
- 数据中心网络架构基本特点
- Fat-tree 网络架构

早期数据中心

1961, Information Processing Center at the National Bank of Arizona

数据中心

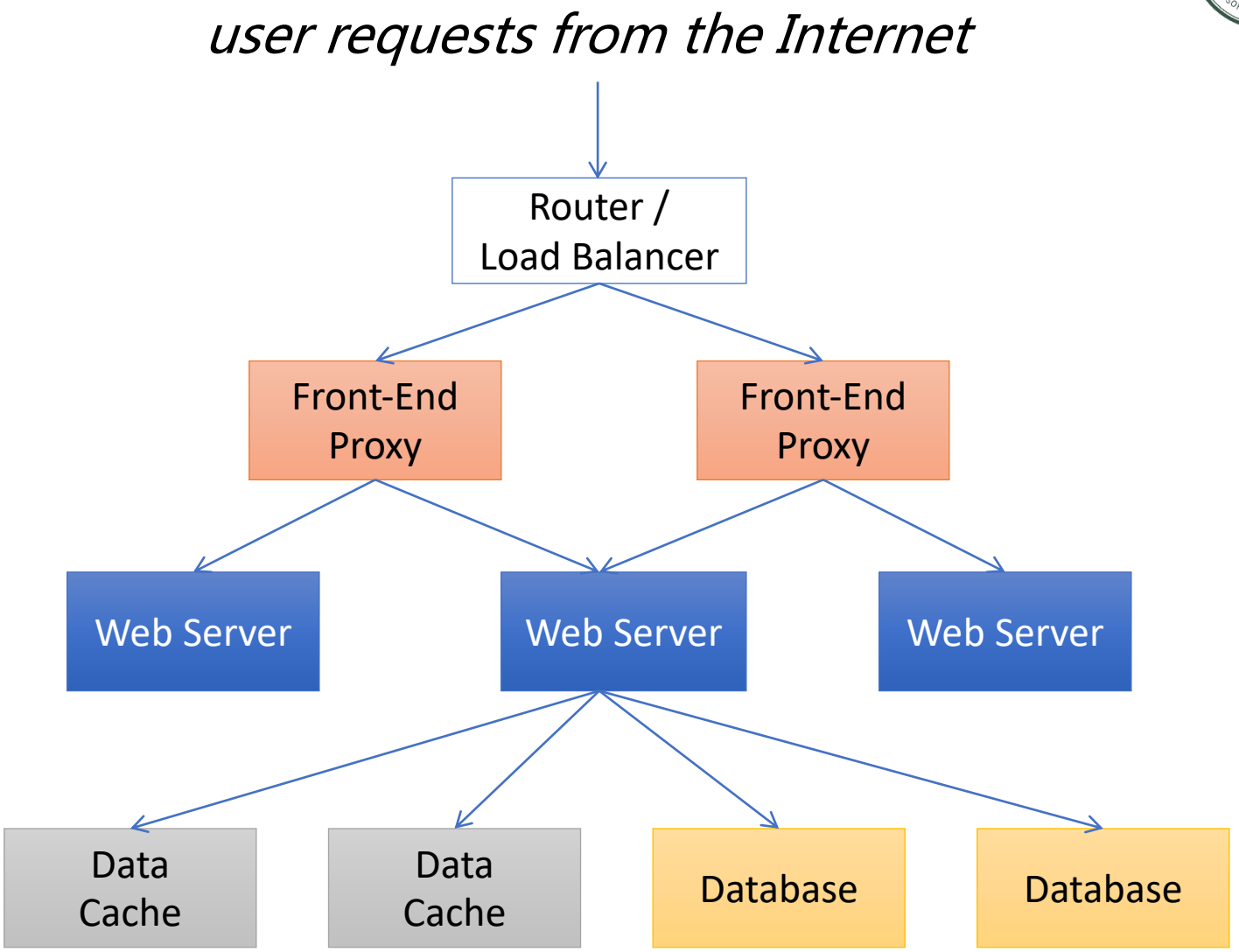


Facebook、谷歌、亚马逊等在多地建立了自己的大规模数据中心

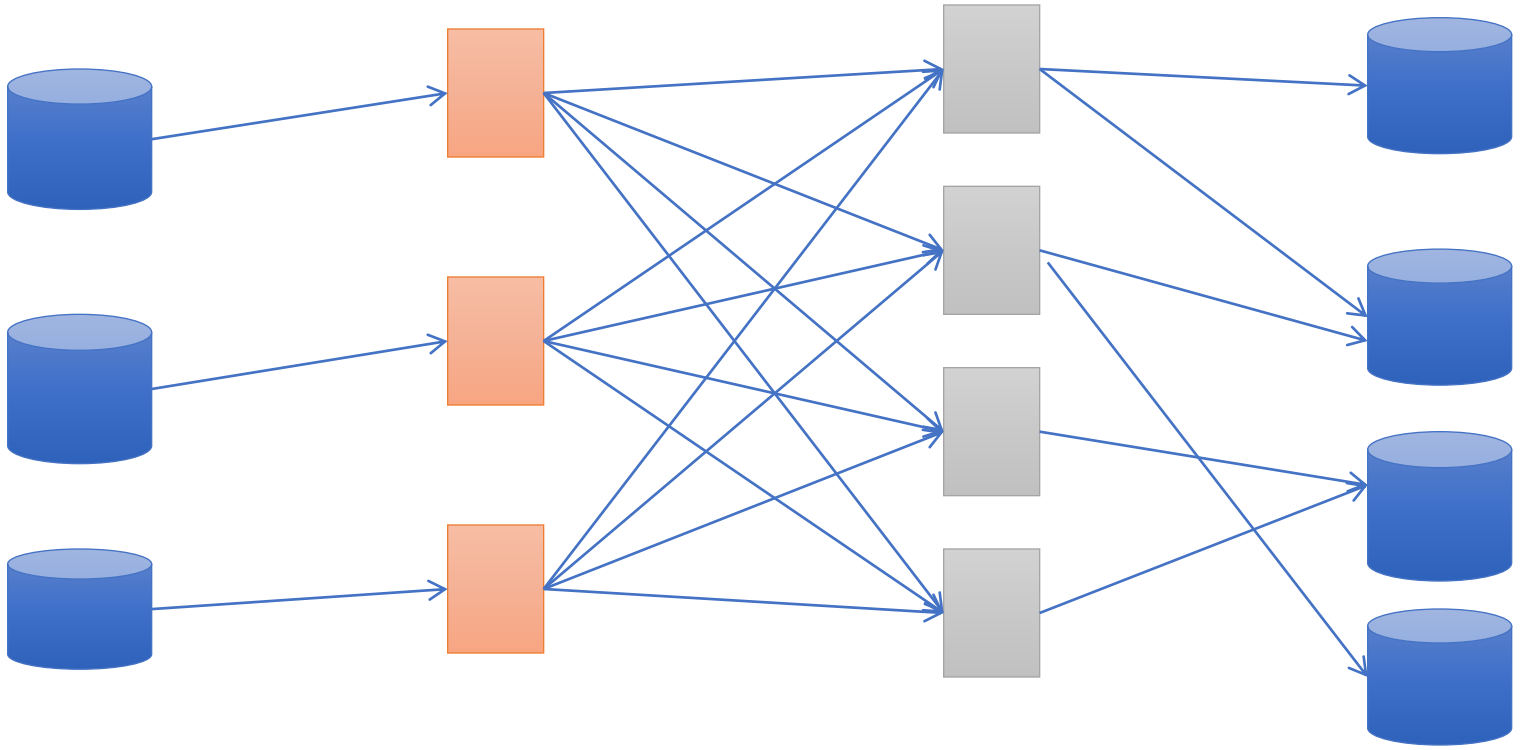
数据中心应用

- 数据中心通常运行两种类型的应用
  - ✓面向外部的服务（e.g., 向用户提供网页浏览）
  - ✓内部计算（e.g., 基于 MapReduce 的网页索引）

外部用户访问网页


user requests from the Internet
Router / Load Balancer
Front-End Proxy
Front-End Proxy
Web Server
Web Server
Web Server
Data Cache
Data Cache
Database
Database

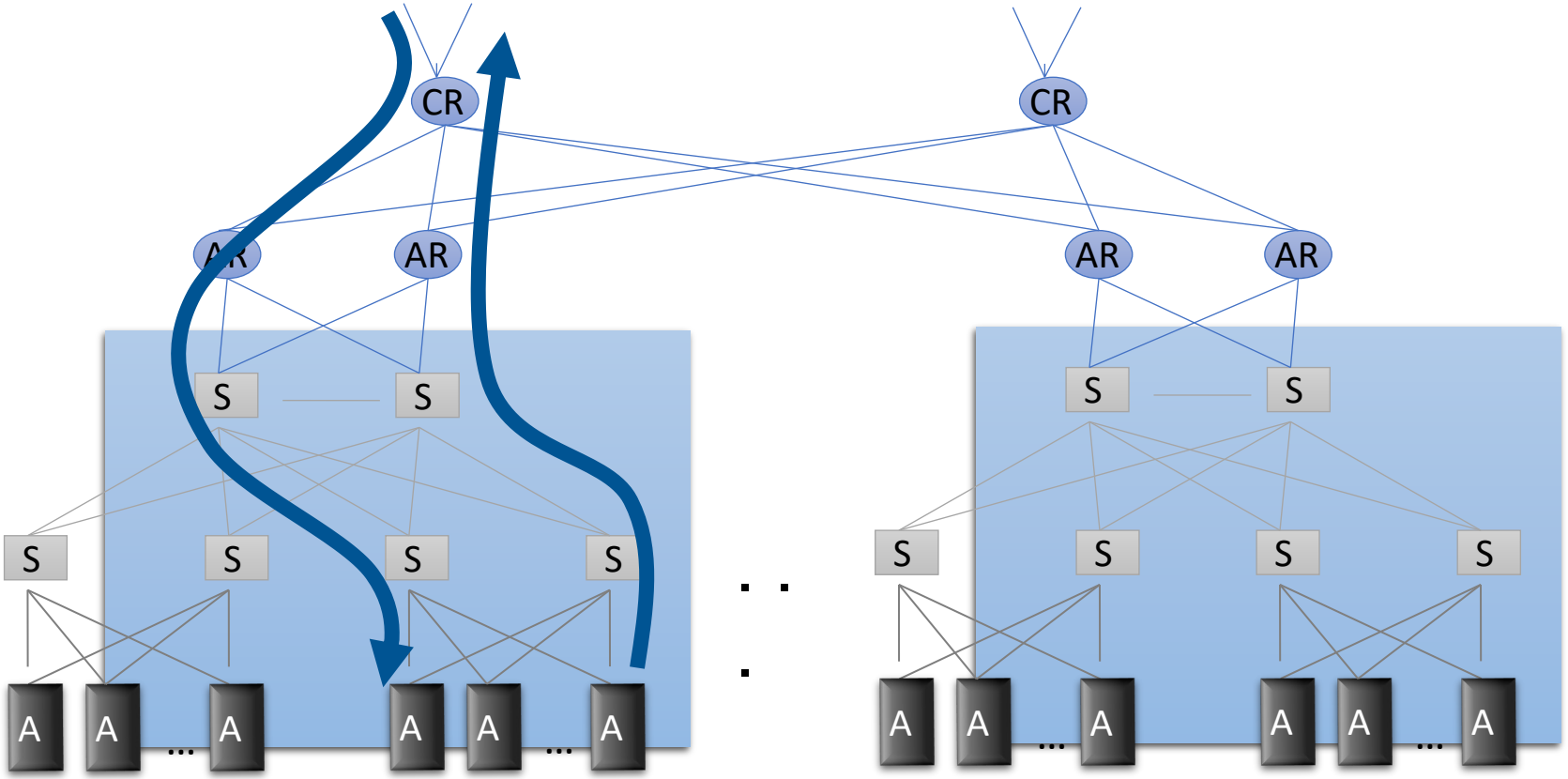
内部计算



Distributed Storage **Map Tasks** **Reduce Tasks** **Distributed Storage**

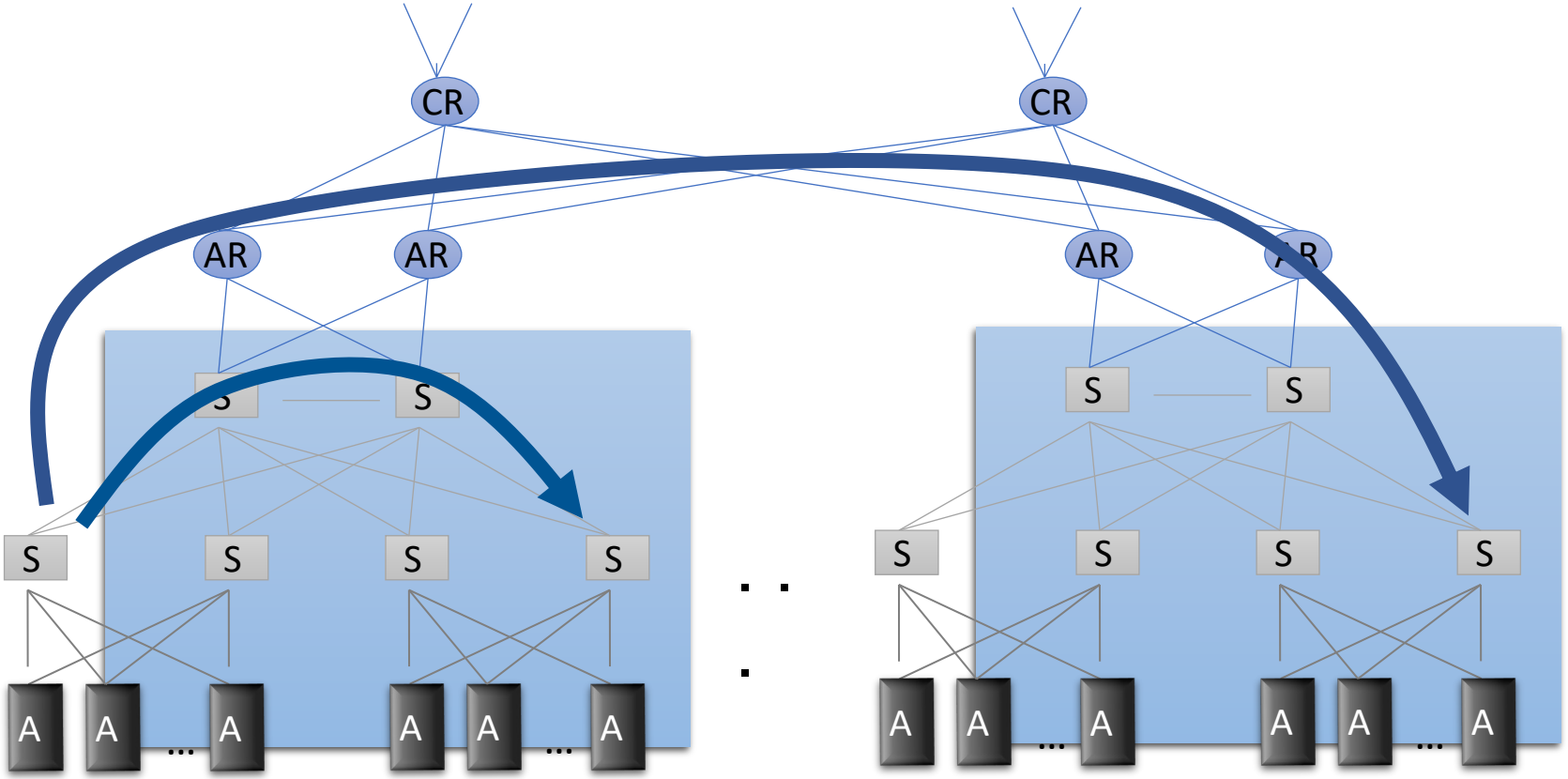
南北向流量（North-south Traffic）

- 外部客户端和数据中心之间的交互式查询与响应
- 由前端 web 服务器、中端应用程序服务器和后端数据库处理



东西向流量（East-west Traffic）

- 数据中心服务器之间的流量
- “大数据”计算中的通信

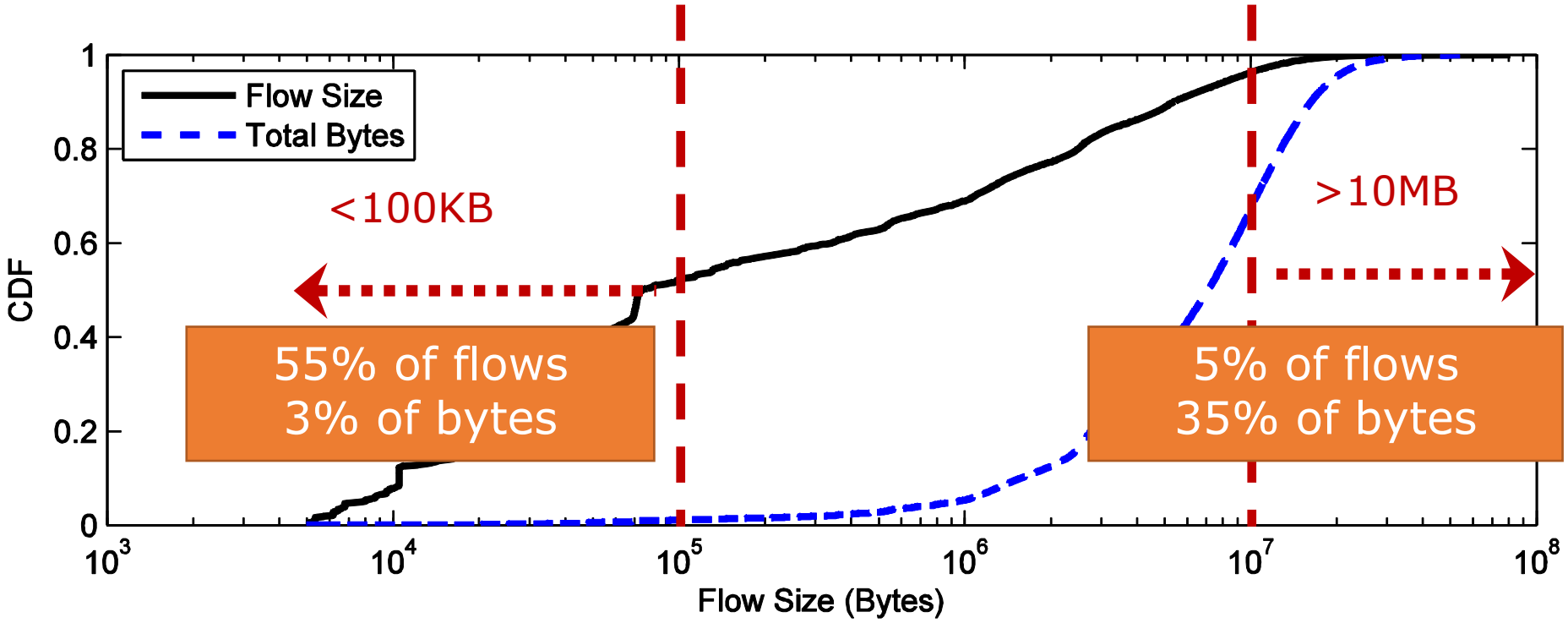


为什么要区分不同方向的流量？

- 对于数据中心网络设计和管理非常重要，因为它们具有不同的特征和需求
  - ✓南北向流量通常需要更高的带宽和更强的安全性，因为它涉及到数据中心与外部网络之间的数据传输
  - ✓而东西向流量则更注重低延迟和高吞吐量，因为它涉及到数据中心内部服务器之间的通信，如数据库查询和应用程序交互等

流量特点

- Web search, data mining (Microsoft) [Alizadeh 2010]



数据中心网络

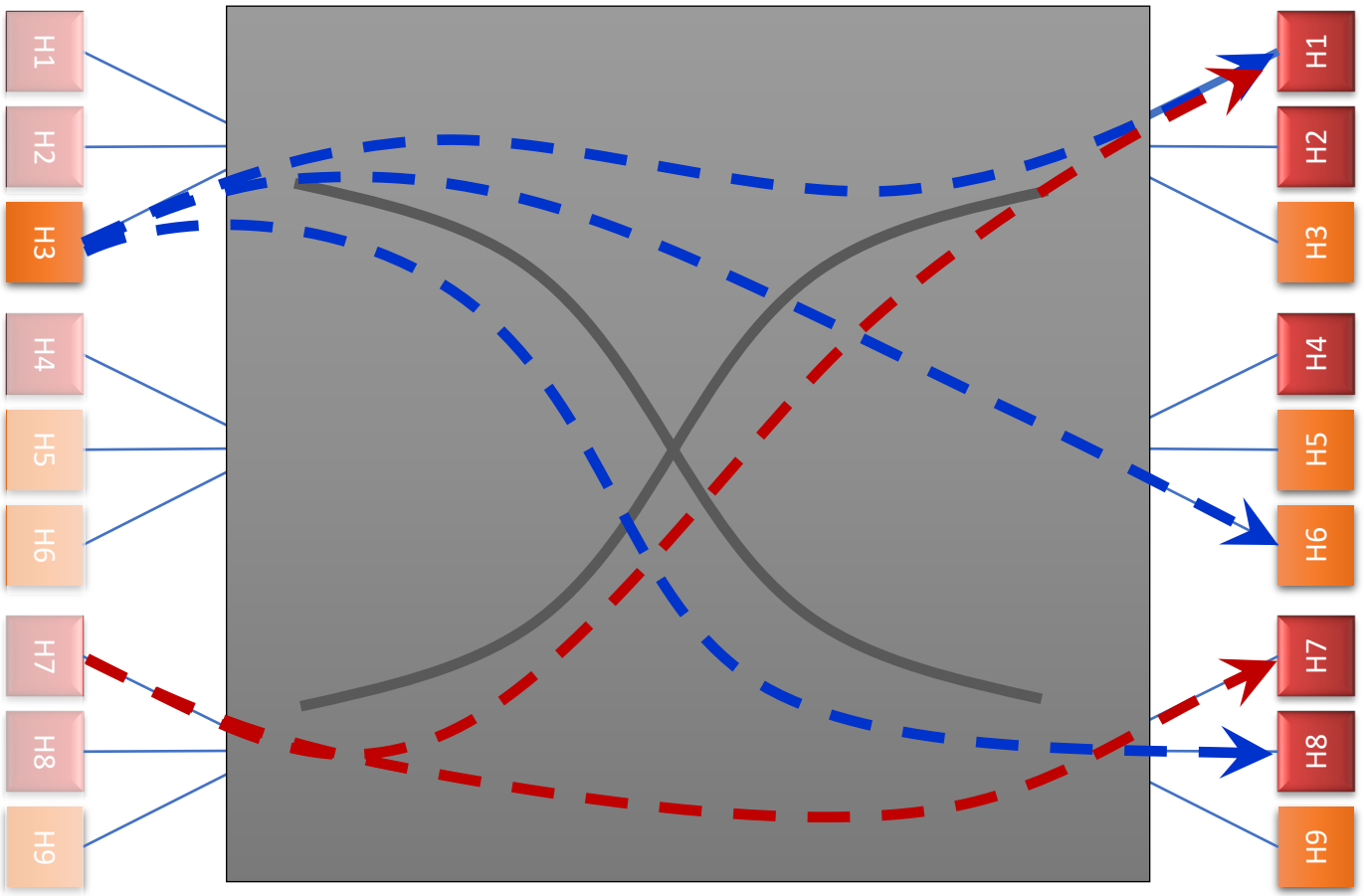
- 数据中心应用及流量
- 数据中心网络架构基本特点
- Fat-tree 网络架构

数据中心网络架构

- 数据中心内部的网络拓扑结构，包括各种网络设备和它们之间的连接方式

- 它是数据中心的基础架构之一，对数据中心的性能、可靠性和可扩展性都有很大的影响

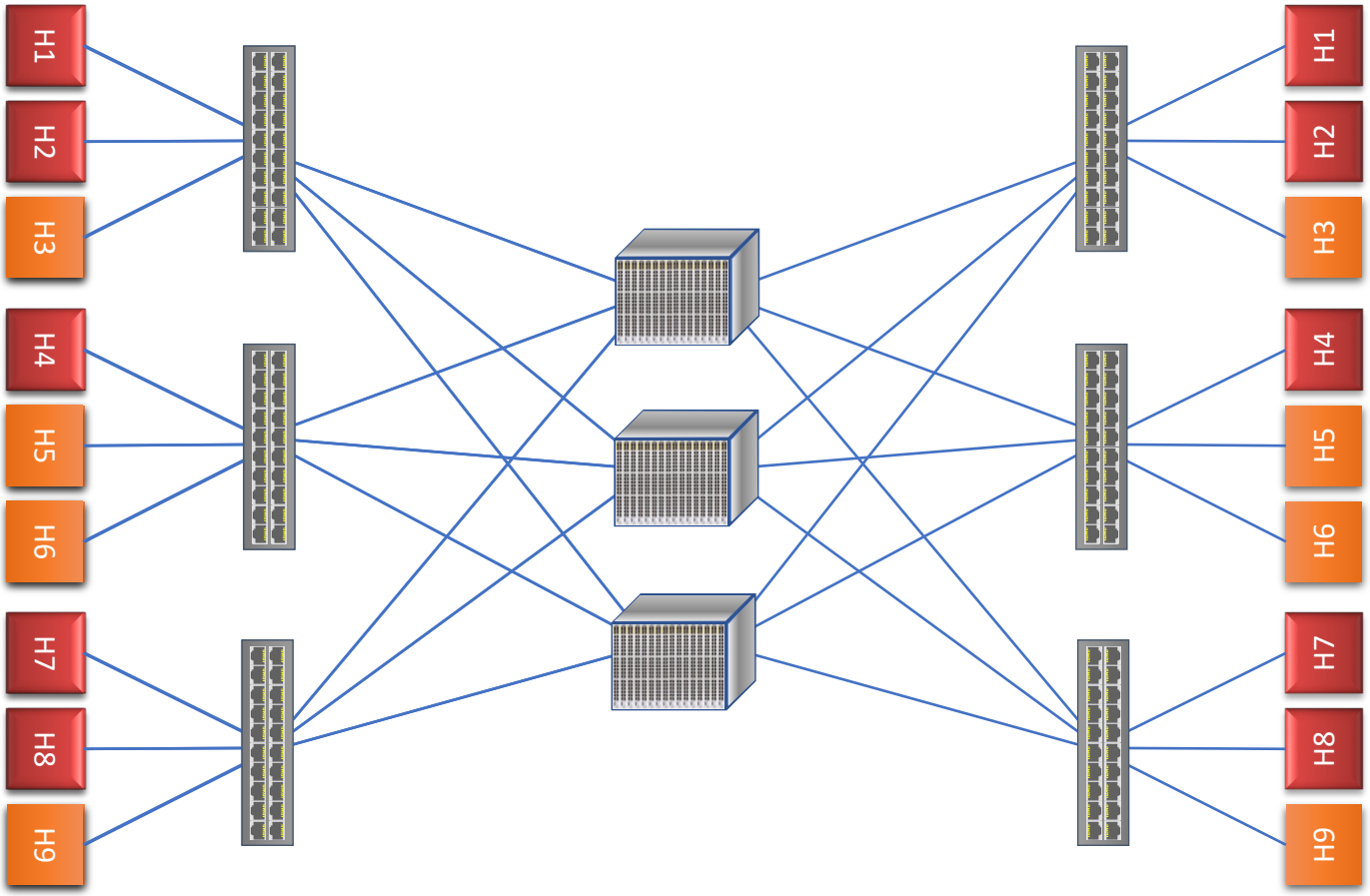
数据中心网络：一个大型交换机



TX

RX

数据中心网络：一个大型交换机



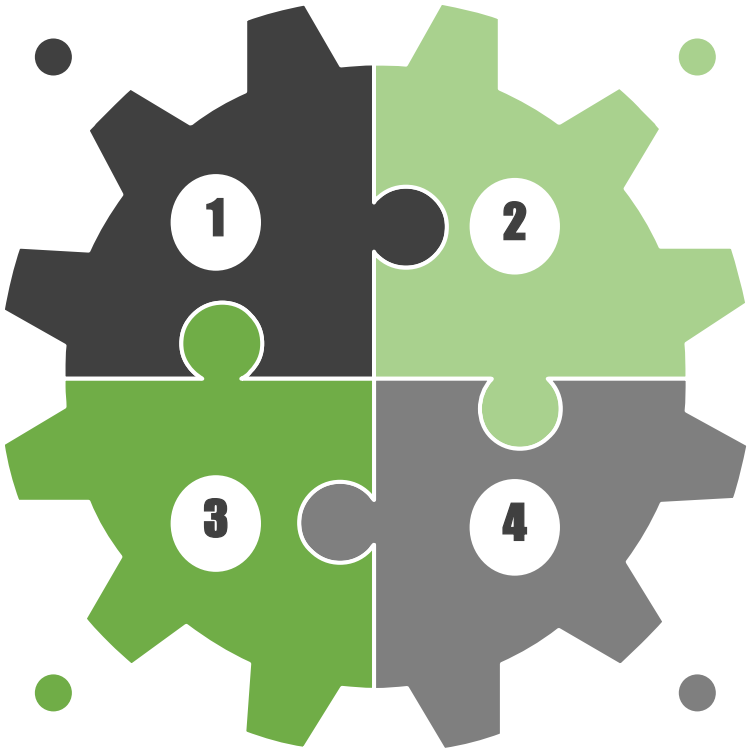
TX

RX

数据中心网络设计要求

高设备利用率

采用良好的架构和虚拟化技术进行系统和数据中心整合,优化资源利用率、简化管理



绿色节能

通过先进的供电和散热技术，降低数据中心的能耗

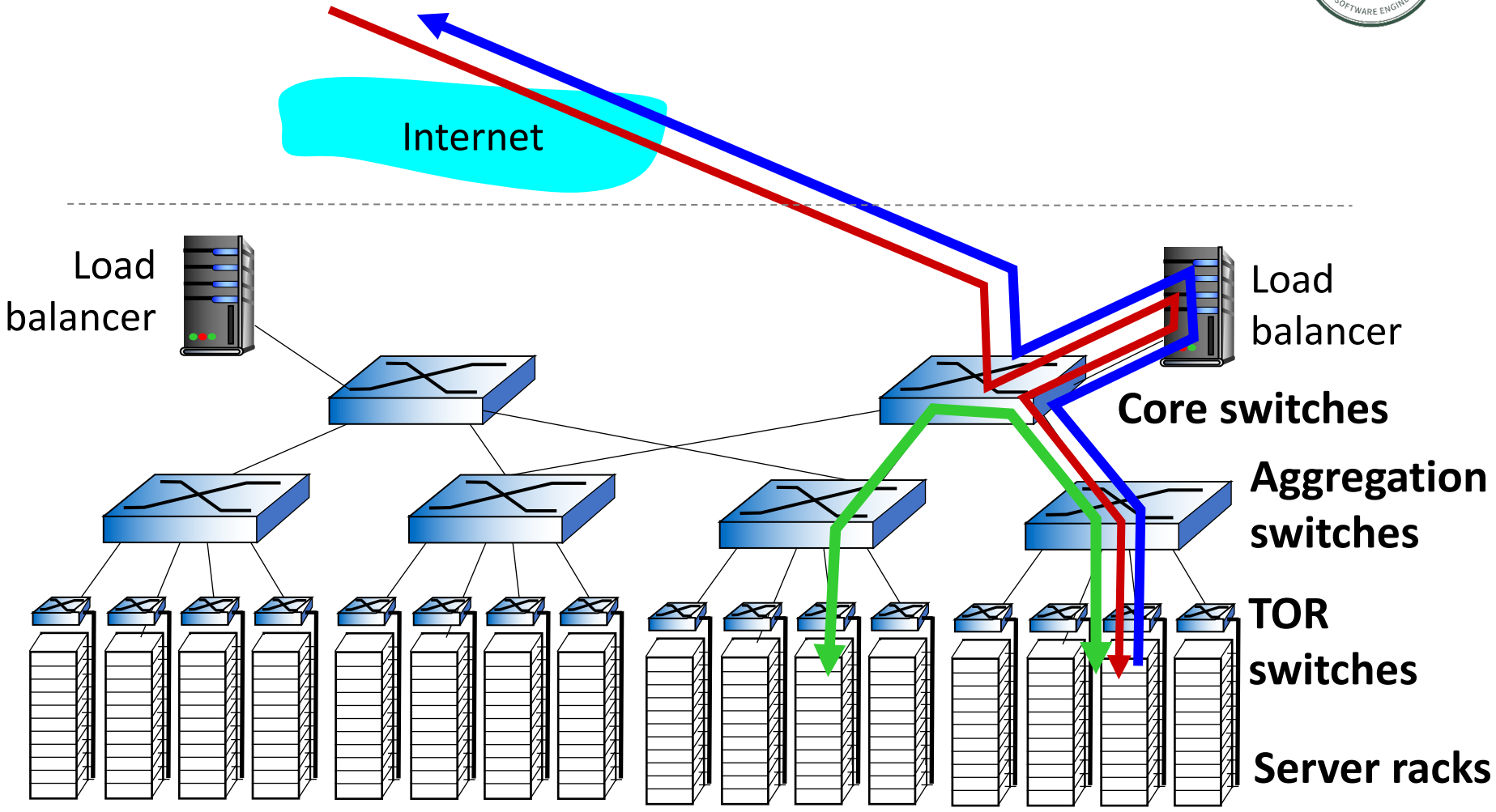
高可用性

当网络故障或升级时，网络能够正常运行，对网络的性能影响不大

自动化管理

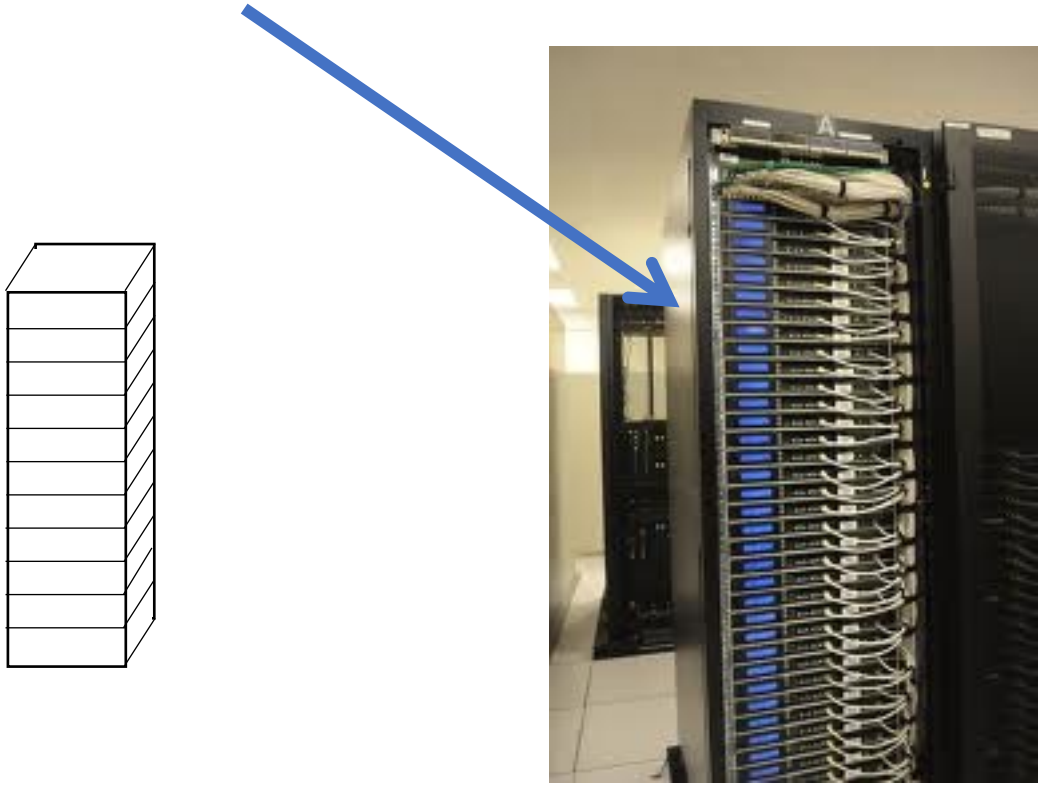
云数据中心应是24×7小时无人值守并可远程管理的

数据中心网络架构


Internet
Load balancer
Load balancer
Core switches
Aggregation switches
TOR switches
Server racks

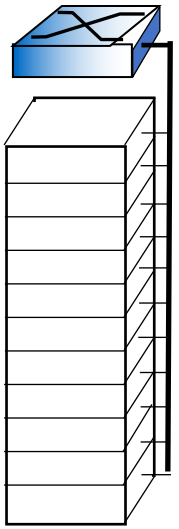
数据中心（网络）内部

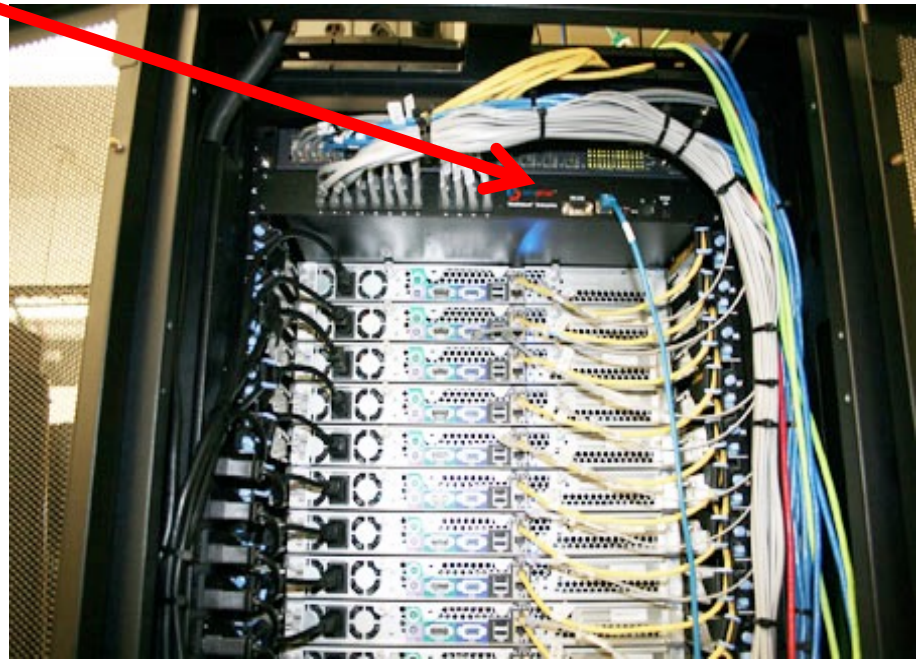
- 以机架（Rack）形式组织服务器（Server）



数据中心（网络）内部

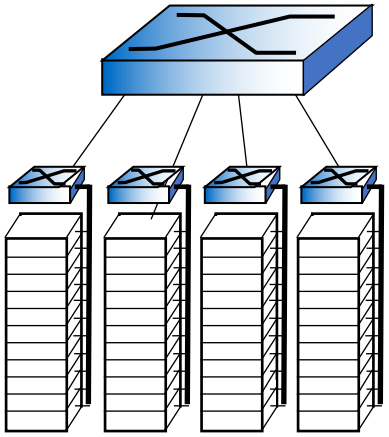
- 以机架（Rack）形式组织的服务器（Server）
- 每个机架都有一个机架（Top of Rack, ToR）交换机

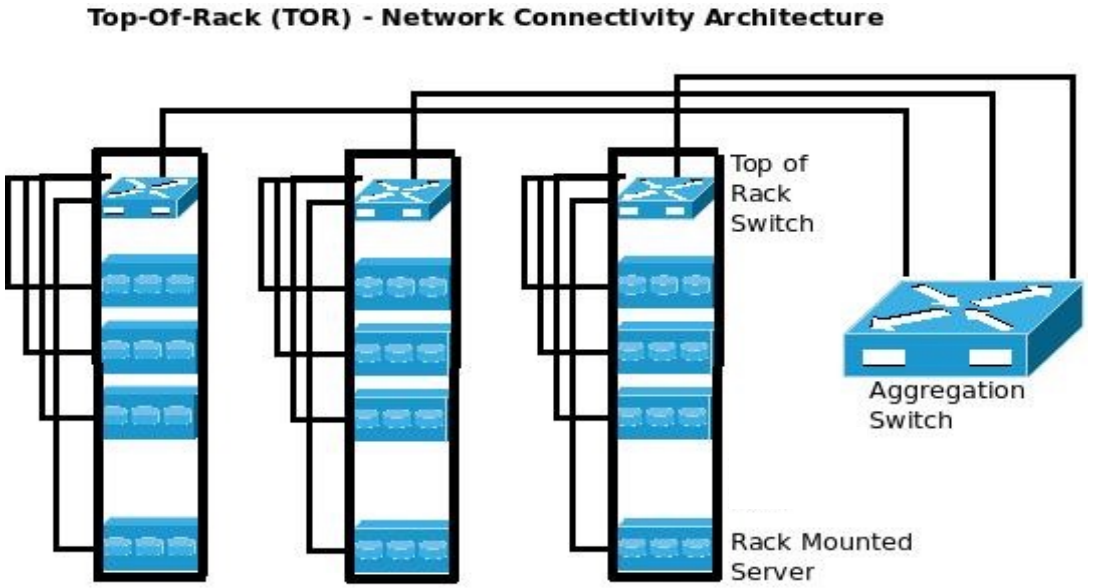




数据中心（网络）内部

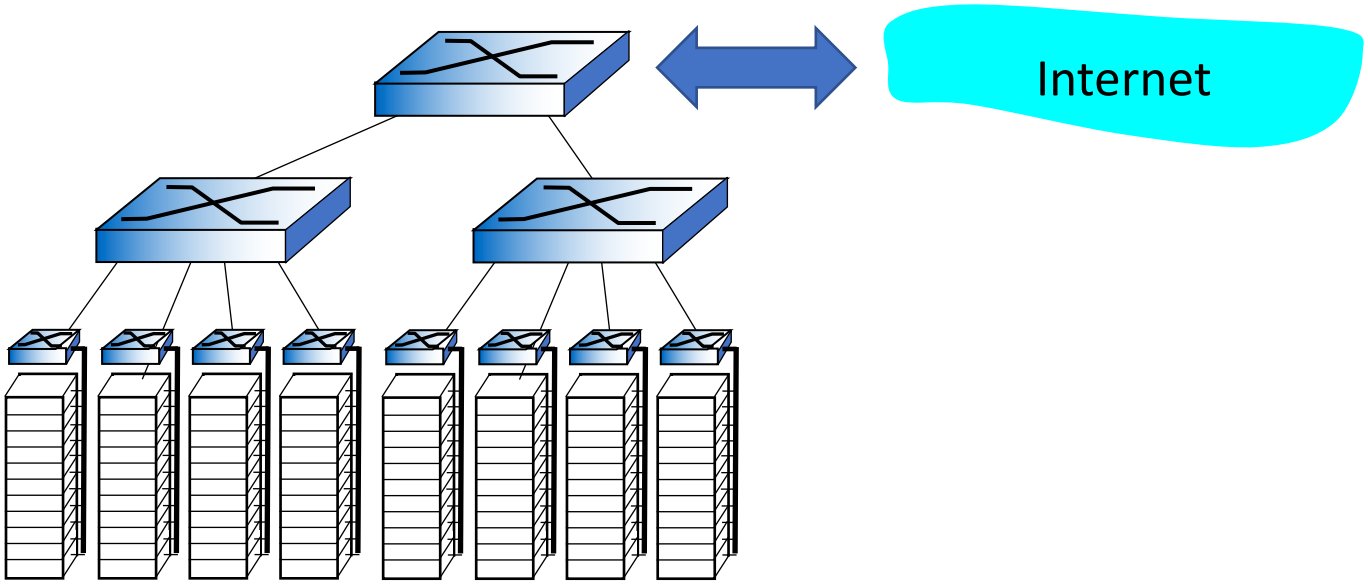
- 以机架（Rack）形式组织的服务器（Server）
- 每个机架都有一个机架（Top of Rack, ToR）交换机
- 汇聚（Aggregation）交换机互连 ToR 交换机





数据中心（网络）内部

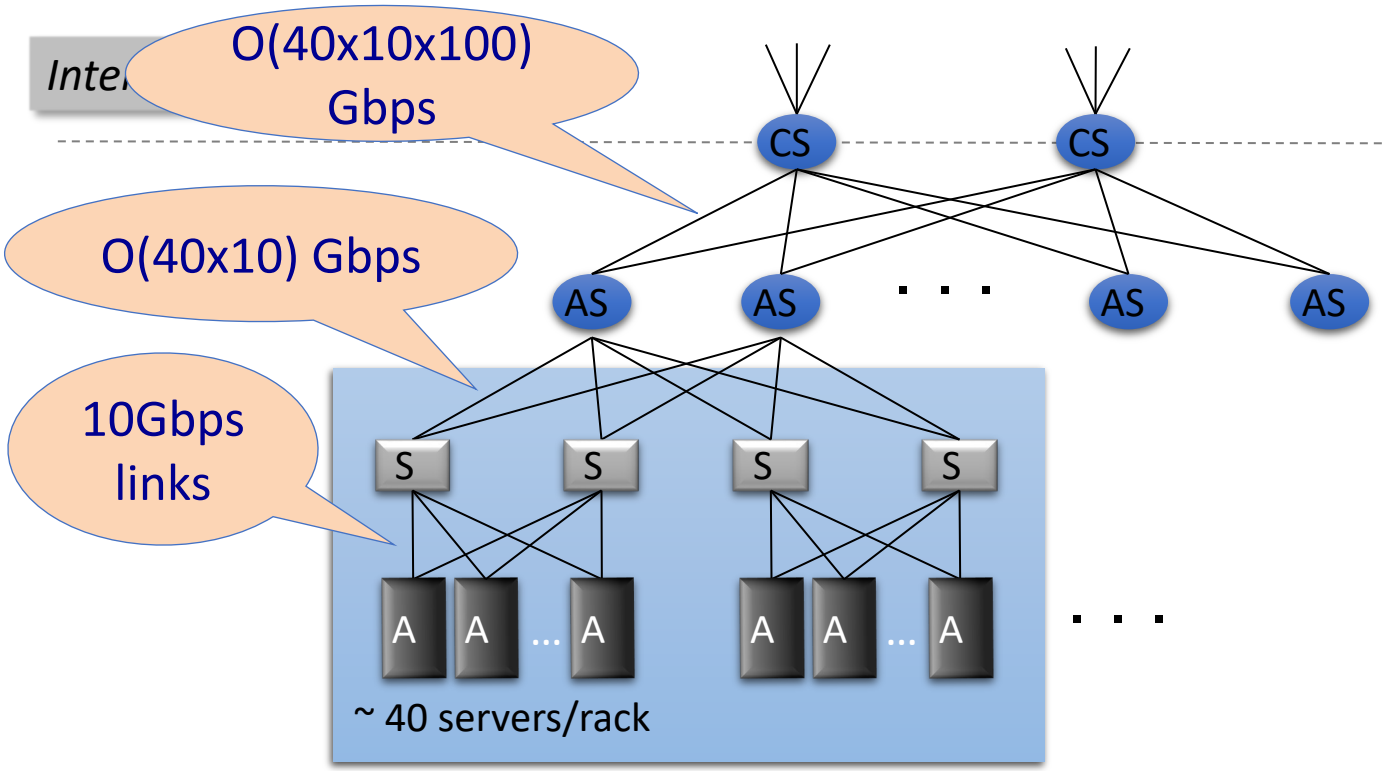
- 以机架（Rack）形式组织的服务器（Server）
- 每个机架都有一个机架（Top of Rack, ToR）交换机
- 汇聚（Aggregation）交换机互连 ToR 交换机
- 通过核心（Core）交换机连接到外部
  - 其他数据中心、互联网和其他网络



二分带宽（Bisection bandwidth）

- 将网络分成两个相等的部分
- 两部分之间的最小带宽就是二分带宽
- 是衡量网络性能的一个重要指标
- 完全二分带宽：具有 N 个节点的网络的完全二分带宽是单个链路带宽的 N/2 倍
  - ✓两部分的任何节点都可以全速通信

完全二分带宽



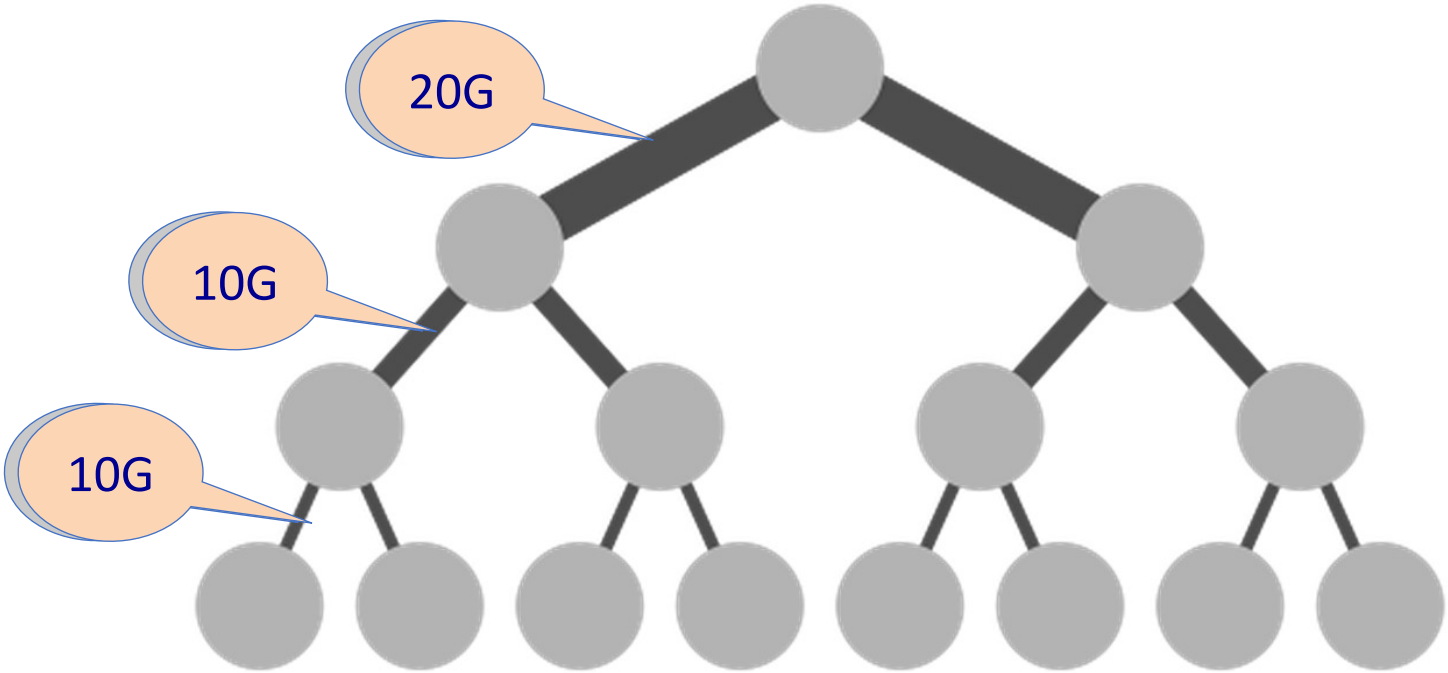
Scale up方法实现完全二分带宽

Scale up方法的问题

- 以 Scale up 方法实现完全二分带宽的问题就是贵
  - ✓需要昂贵/非商用的交换机和链路

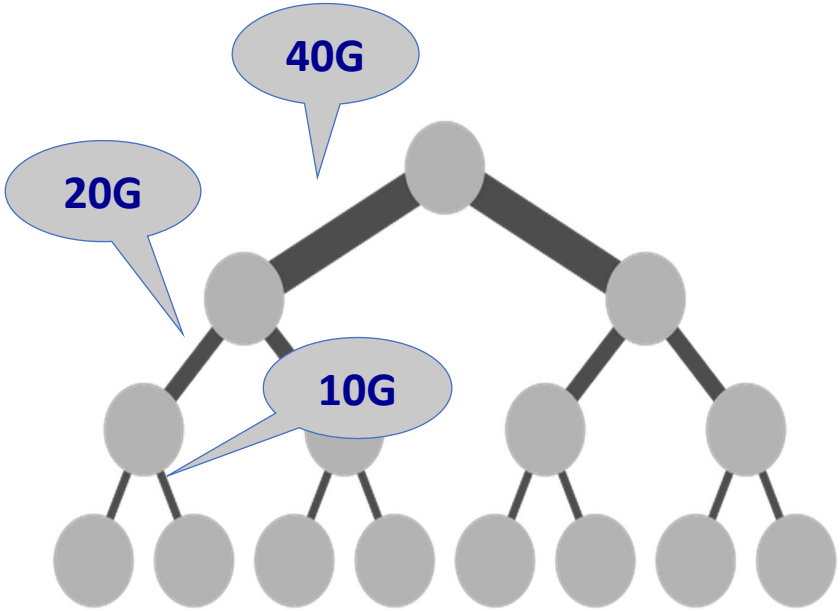
- 解决方法
  - ✓超额订购（Oversubscription），即节点抢用带宽，并不总以全速通信，网络中的超额订购率一般为 2.5:1 (400 Mbps) 到 8:1 (125 Mbps)
  - ✓更优的网络架构

超额订购

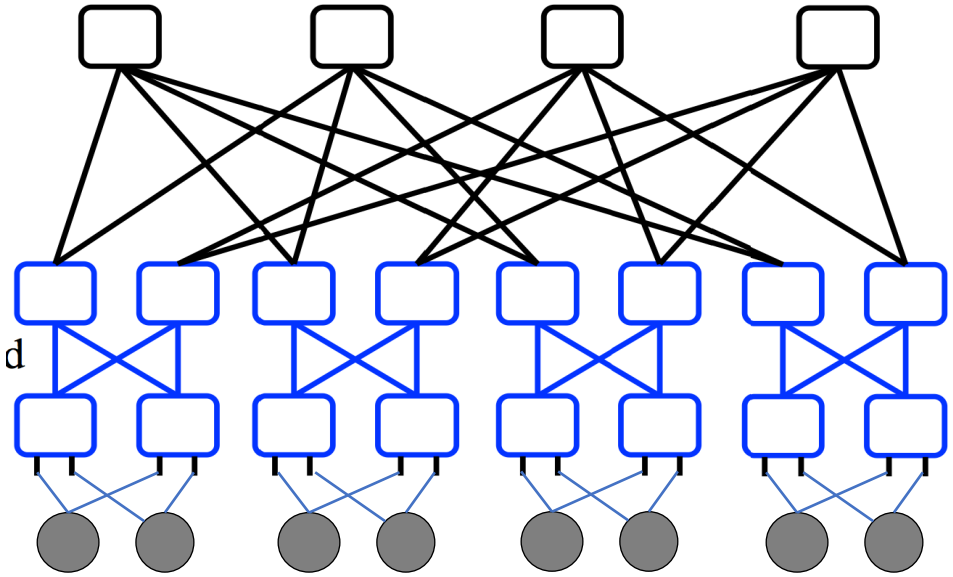


Oversubscription需要特定的算法来减少阻塞

更优的网络架构



scale up 的方法



scale out 的方法

数据中心网络

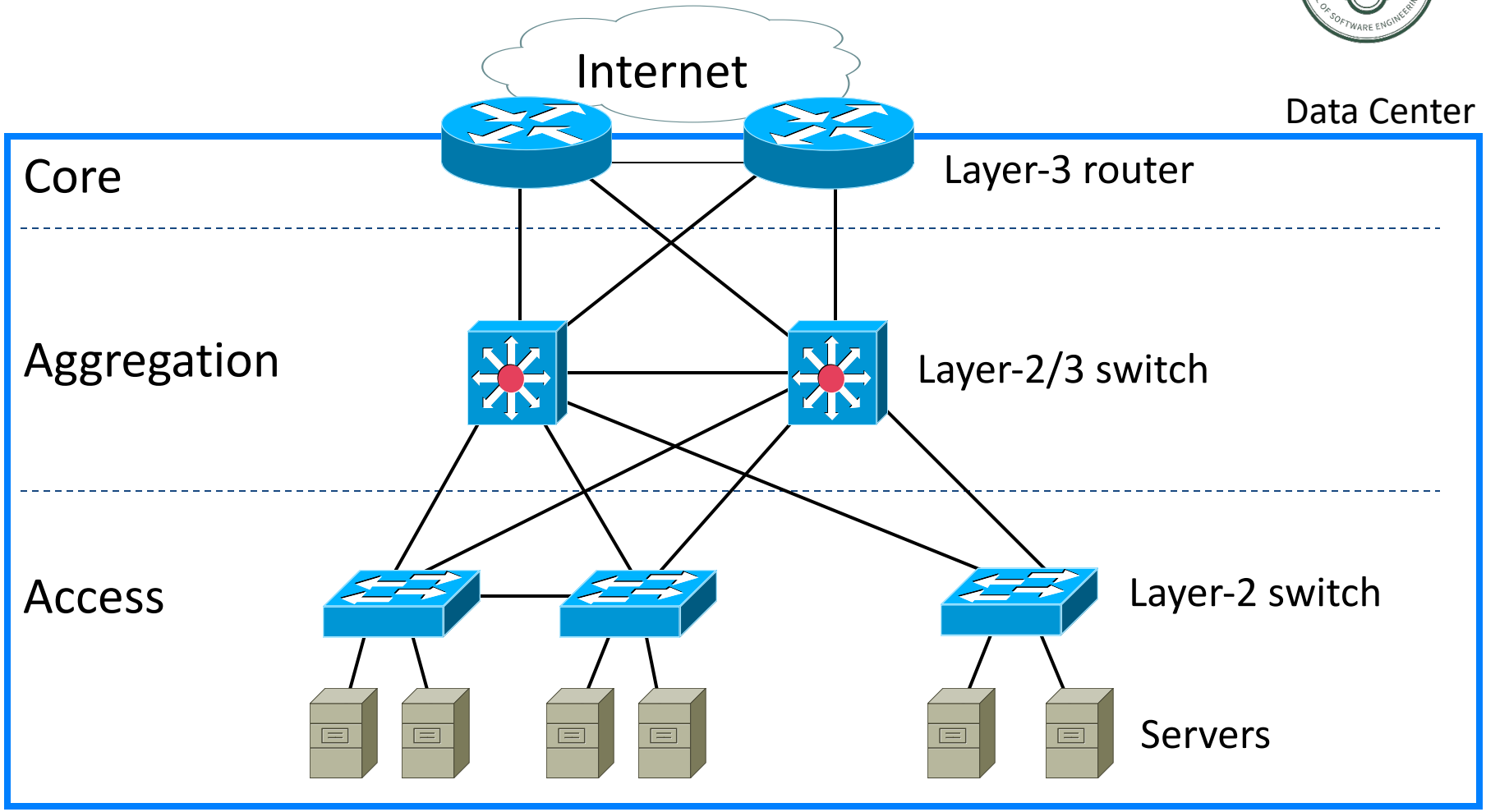
- 数据中心应用及流量
- 数据中心网络架构基本特点
- Fat-tree 网络架构

大量新的网络架构被提出

- 为了适应新型应用的需求，数据中心网络需要在低成本的前提下满足高扩展性、低配置开销、健壮性和节能的要求

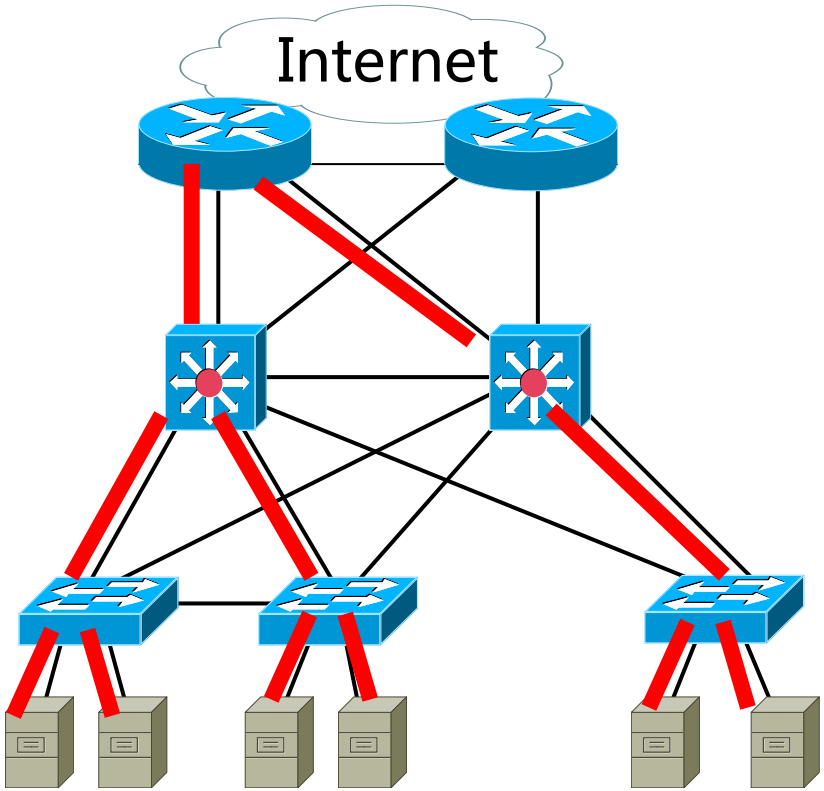
网络拓扑	规模	带宽	容错性	扩展性	布线复杂性	成本	兼容性	配置开销	流量隔离	灵活性
FatTree	中	中	中	中	较高	较高	高	较高	无	低
VL2	大	大	中	中	较高	较高	中	较高	无	中
OSA	小	大	差	中	较低	较高	低	中	无	高
WDCN	小	大	较好	中	较低	中	中	中	无	高
DCell	大	较大	较好	较好	高	较高	中	较高	无	较高
FiConn	大	较大	较好	较好	较高	中	中	较高	无	较高
BCube	小	大	好	较好	高	较高	中	较高	无	较高
MDCube	大	大	较好	较好	高	高	中	较高	无	较高

传统的网络三层架构


Internet
Data Center
Core
Layer-3 router
Aggregation
Layer-2/3 switch
Access
Layer-2 switch
Servers

传统三层架构的问题

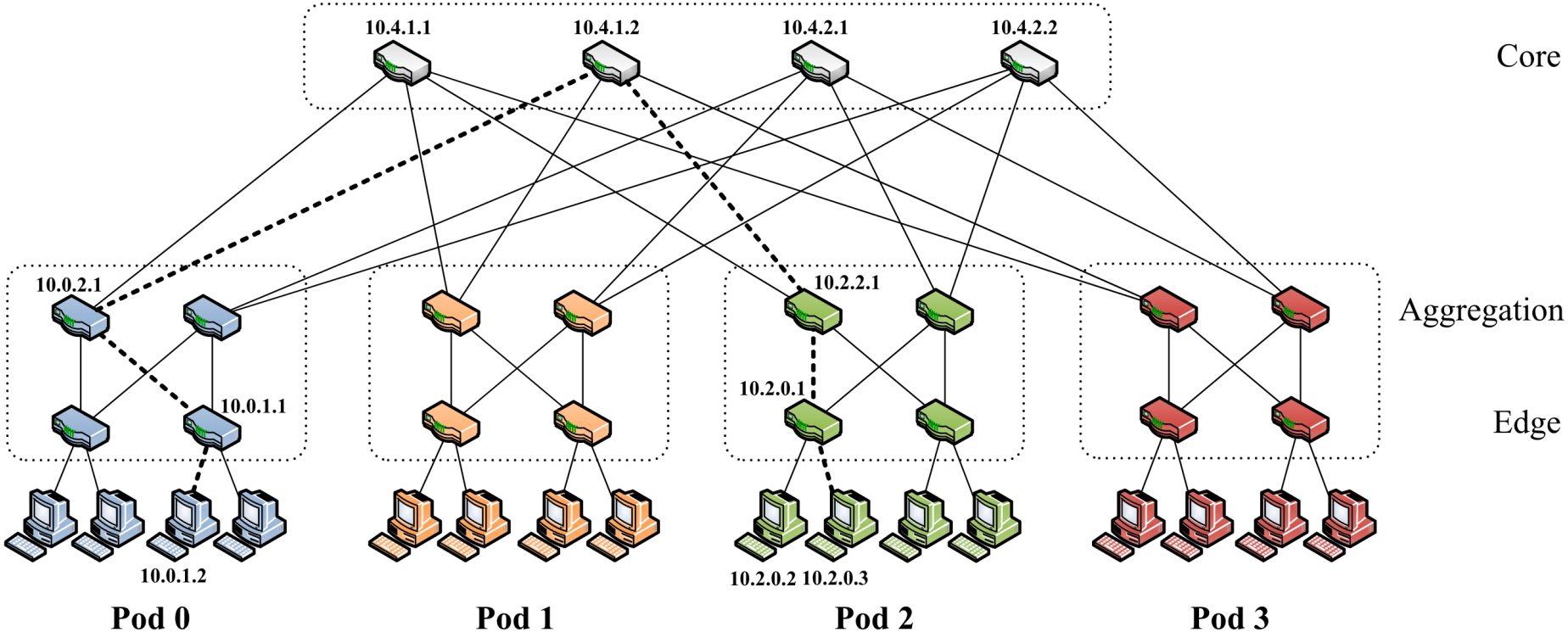
- 使用 spanning tree protocol
  - ✓避免数据包循环
  - ✓忽略备用路径
- 存在的问题
  - ✓单点故障
  - ✓拓扑层次越高，超额订购率越高



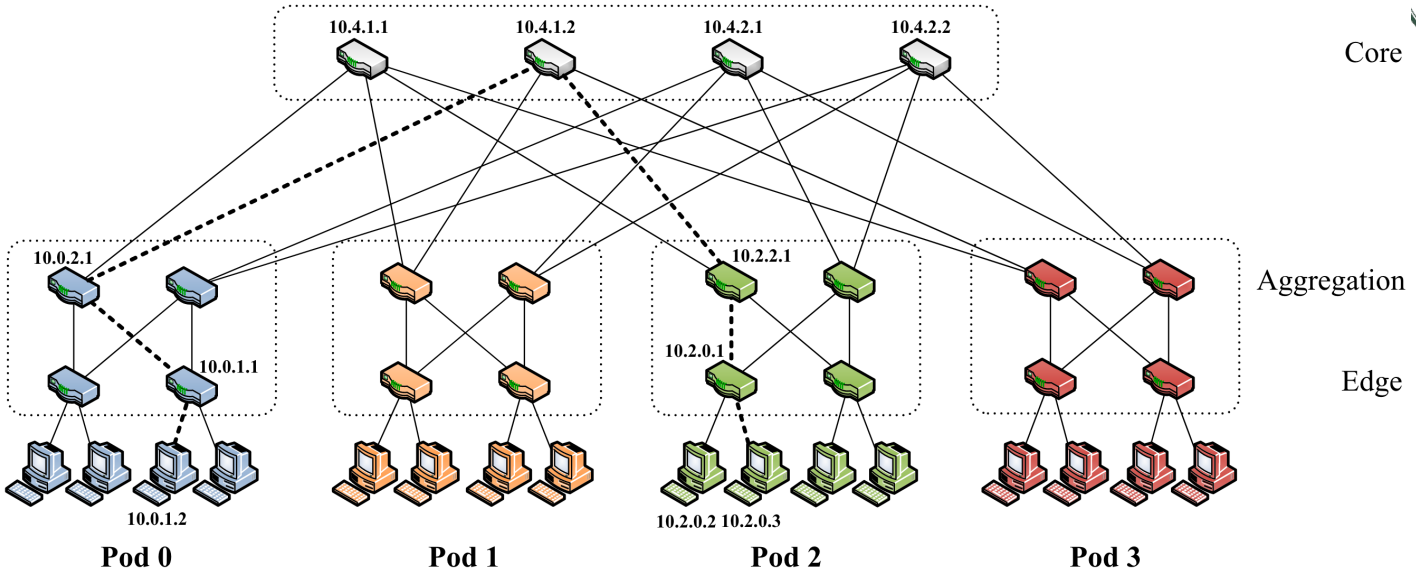
Fat-tree 网络架构

- 基础设施由廉价设备组成
  - ✓每个端口支持与端主机相同的速度
- 如果数据包分布到不同路径，则所有设备都能全速传输
- k 端口的 fat-tree 可以支持 $k^3/4$ 个主机

Fat-tree架构（1）


10.4.1.1
10.4.1.2
10.4.2.1
10.4.2.2
Core
10.0.2.1
10.0.1.1
10.0.1.2
10.2.2.1
10.2.0.1
10.2.0.2 10.2.0.3
Aggregation
Edge
Pod 0
Pod 1
Pod 2
Pod 3

Fat-tree架构（2）



k-ary fat-tree：

- ✓ 每个交换机有 k 个端口，网络包含 k 个 pod
- ✓ 每个 pod 包含两层交换机（edge 和 aggregation），每层有 $k/2$ 个
- ✓ 每个 edge 交换机分别与 $k/2$ 个主机和 $k/2$ 个 aggregation 交换机相连
- ✓ 网络中共有 $(k/2)^2$ 个 core 交换机，所有 core 交换机的第 i 个端口连接第 i 个 pod
- ✓ Aggregation 交换机的端口以 $k/2$ 的划窗连接 core 交换机
- ✓ 能支持 $k^3/4$ 个主机，当 $k = 4$ 时，$k^3/4 = 4^3/4 = 16$

简单 Fat-tree 架构的优缺点

- 简单 Fat-tree 架构的优点
  - ✓所有交换机相同，可使用廉价的商用机降低成本
  - ✓任意两个主机有 $(k/2)^2$ 条最短路径，oversubscription ratio 低

- 简单 Fat-tree 架构的缺点
  - ✓OSPF（Open Shortest Path First）协议可能仅使用二层的一条最短路径
  - ✓如果三层盲目地利用多条最短路径，则可能需要对数据包进行重排序

改进 Fat-tree 架构

A Scalable, Commodity Data Center Network Architecture

Mohammad Al-Fares
malfares@cs.ucsd.edu

Alexander Loukissas
aloukiss@cs.ucsd.edu

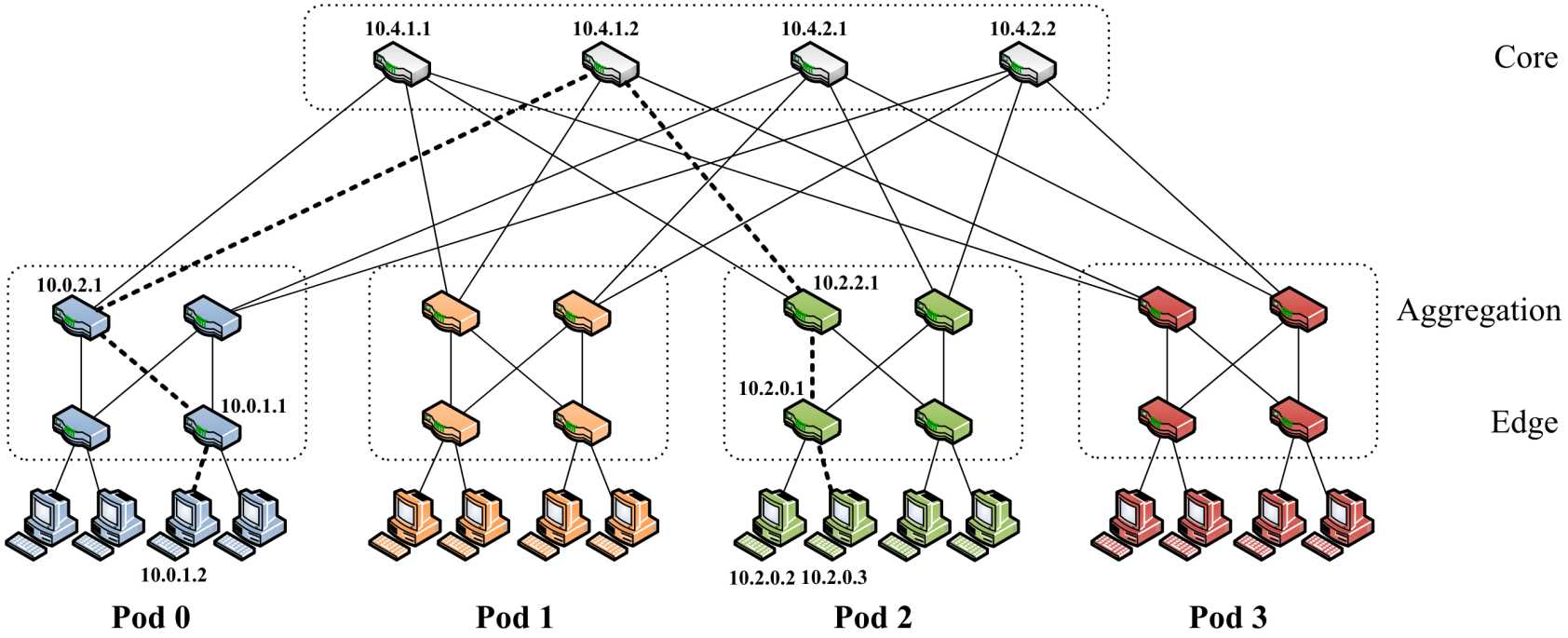
Amin Vahdat
vahdat@cs.ucsd.edu

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404

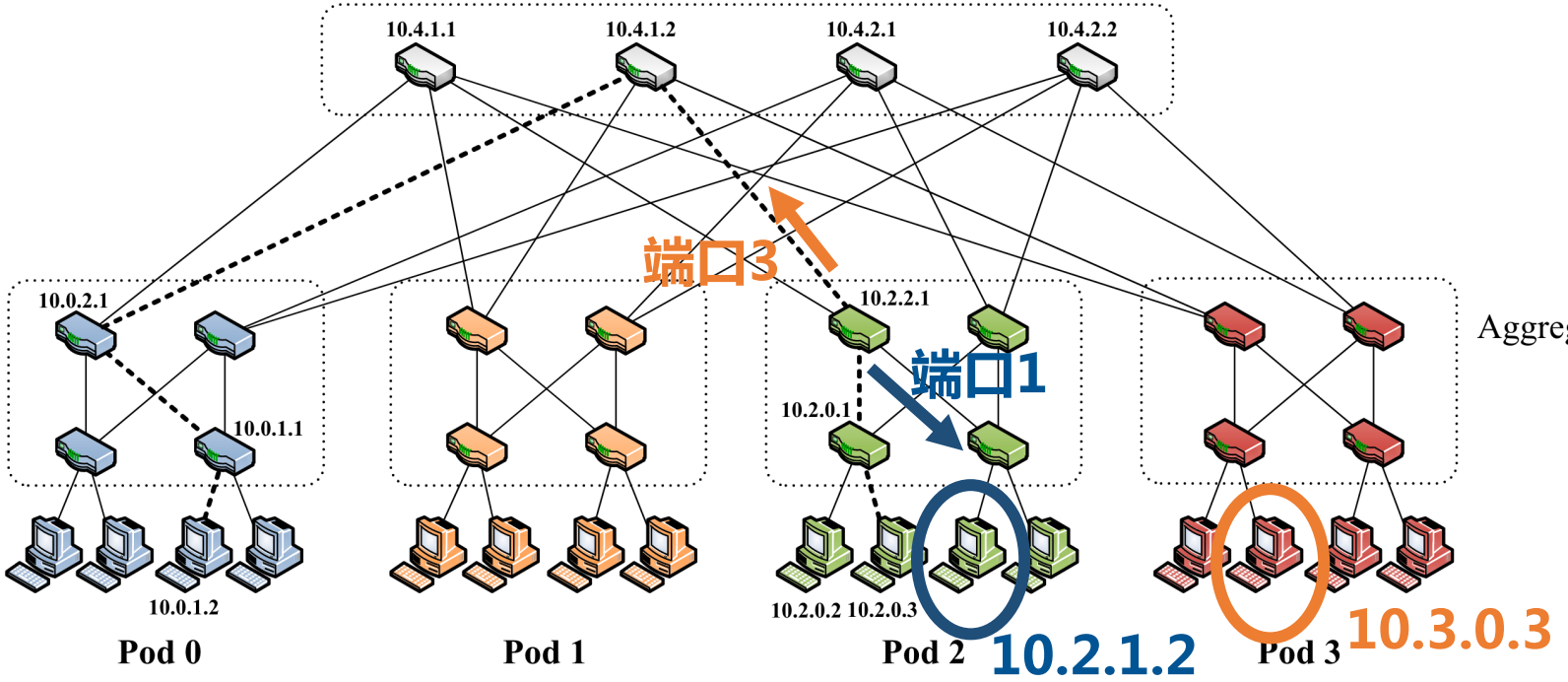
SIGCOMM’ 08

两级路由表

- 第一级是前缀查找
  - ✓用于将拓扑向下路由到端主机
- 第二级是后缀查找
  - ✓用于向核心路由
  - ✓扩散和分散交通
  - ✓通过对相同的端主机使用相同的端口来维护数据包顺序



两级路由表例子



Prefix	Output port
10.2.0.0/24	0
10.2.1.0/24	1
0.0.0.0/0	

Suffix	Output port
0.0.0.2/8	2
0.0.0.3/8	3

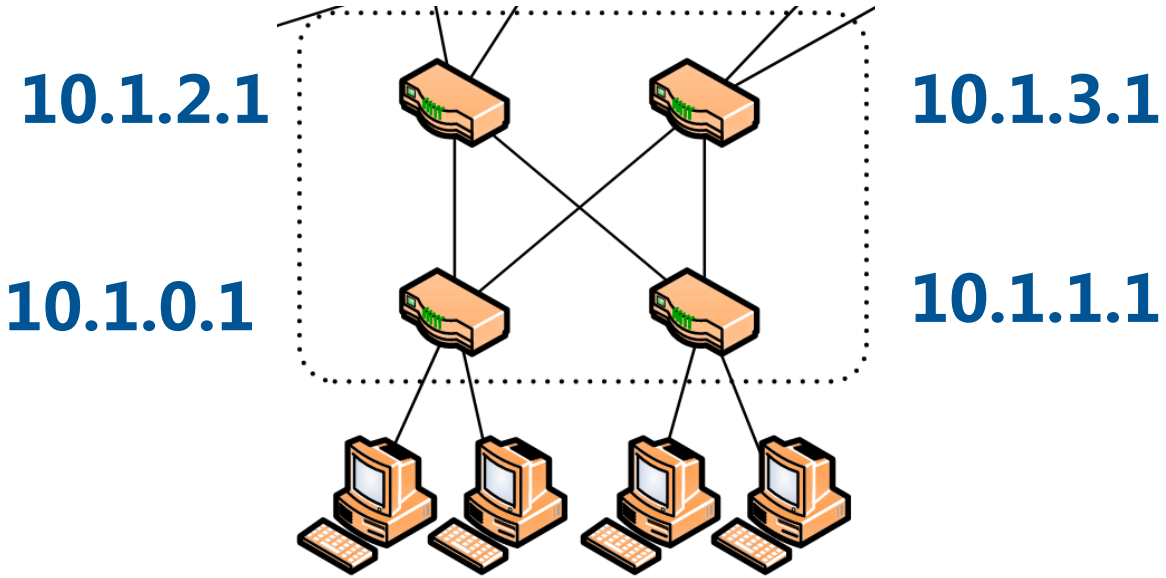
10.2.2.1 的路由表。目的地IP地址为10.2.1.2的传入数据包在端口1上转发，而目的地IP名称为10.3.0.3的数据包则在端口3上转发。

二级路由表算法步骤

- IP地址编排
- 路由表生成

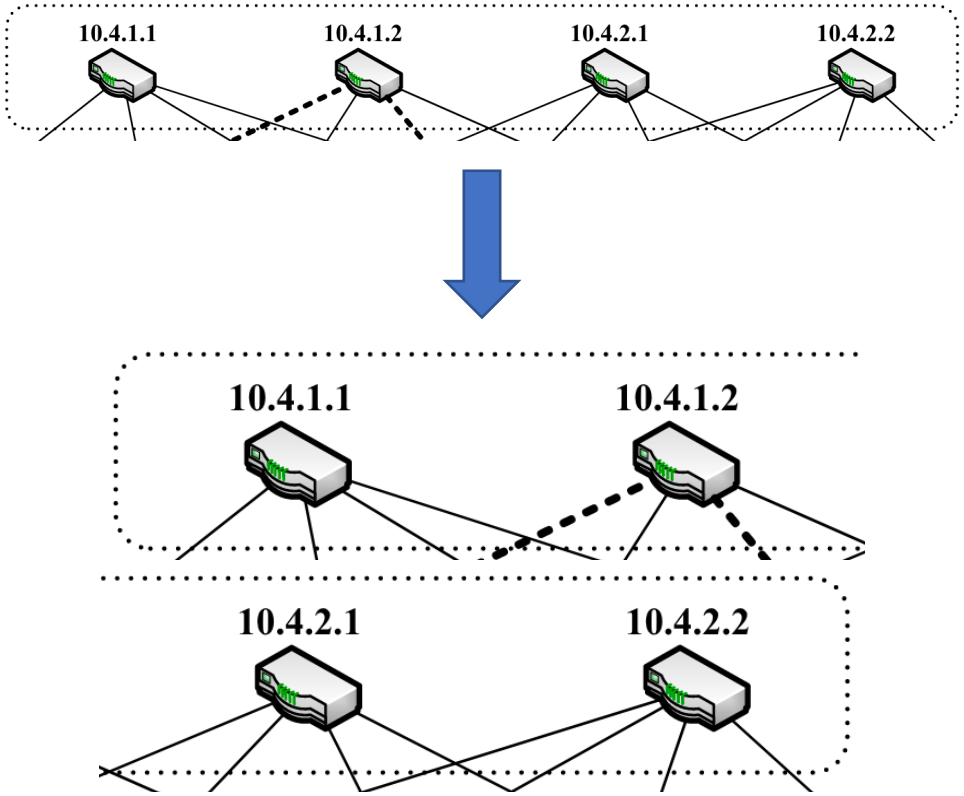
IP地址编排

- Pod 交换机 IP 地址：$10.pod.switch.1$
  - ✓pod 表示 pod ID（$[0, k-1]$）
  - ✓$switch$ 表示 pod 内的 switch ID（$[0, k-1]$），从左到右，从下到上



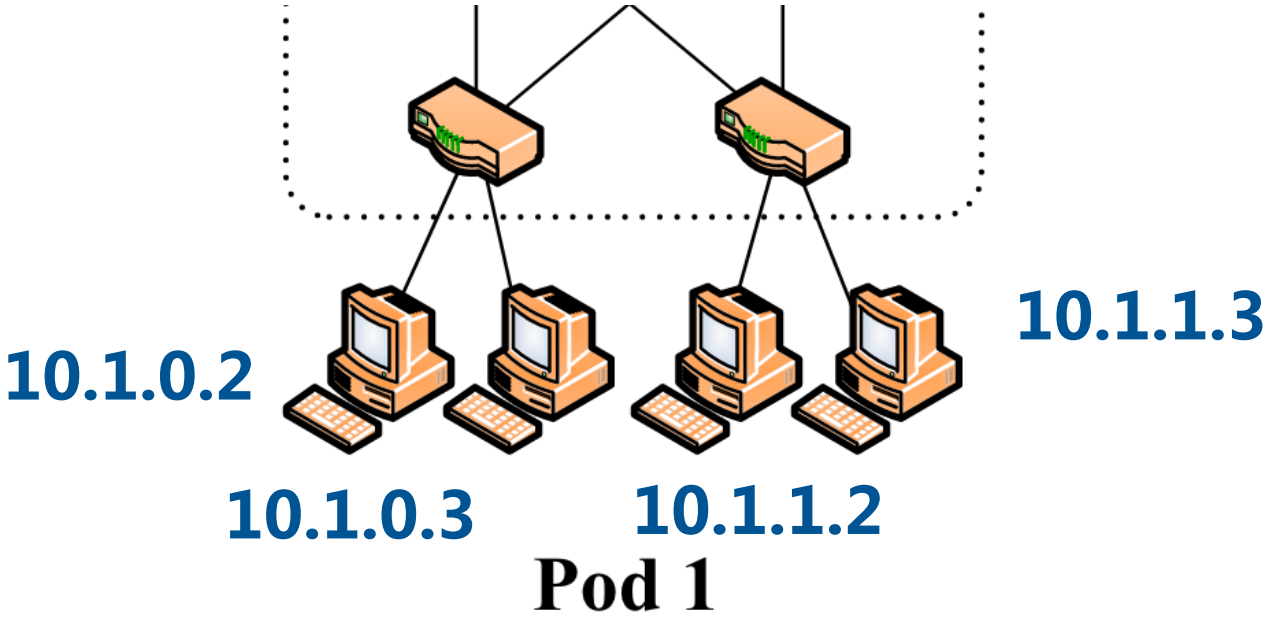
IP地址编排

- Core 交换机的 IP 地址：$10.k.j.i$
  - ✓j 和 i 表示交换机在 $(k/2)^2$ core 交换机网格中的坐标，每个在区间 $[1,(k/2)]$



IP地址编排

- 主机的 IP 地址：$10.pod.switch.ID$
  - ✓ID 表示主机在子网中的 ID（$[2, k/2+1]$）
  - ✓每个 edge 交换机负责 /24 子网中的 $k/2$ 个主机，$k < 256$



路由表生成（1）

- Aggregation 交换机路由表

```
1  foreach pod x in [0, k − 1] do
2      foreach switch z in [(k/2), k − 1] do
3          foreach subnet i in [0, (k/2) − 1] do
4              addPrefix(10.x.z.1, 10.x.i.0/24, i);
5          end
6          addPrefix(10.x.z.1, 0.0.0.0/0, 0);
7          foreach host ID i in [2, (k/2) + 1] do
8              addSuffix(10.x.z.1, 0.0.0.i/8,
               (i − 2 + z)mod(k/2) + (k/2));
9          end
10     end
11 end
```

路由表生成（2）

- Edge 交换机路由表

```
1  foreach pod x in [0, k − 1] do
2      foreach switch z in [0, (k/2) − 1]  do
3
4
5
6          addPrefix(10.x.z.1, 0.0.0.0/0, 0);
7          foreach host ID i in [2, (k/2) + 1] do
8              addSuffix(10.x.z.1, 0.0.0.i/8,
               (i − 2 + z)mod(k/2) + (k/2));
9          end
10     end
11 end
```

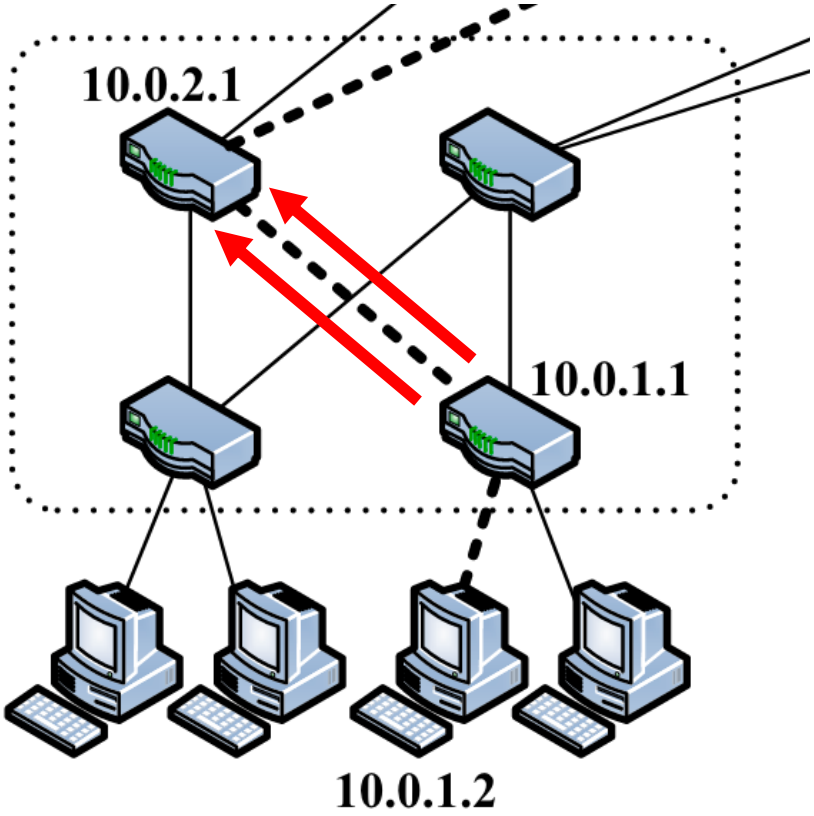
路由表生成（3）

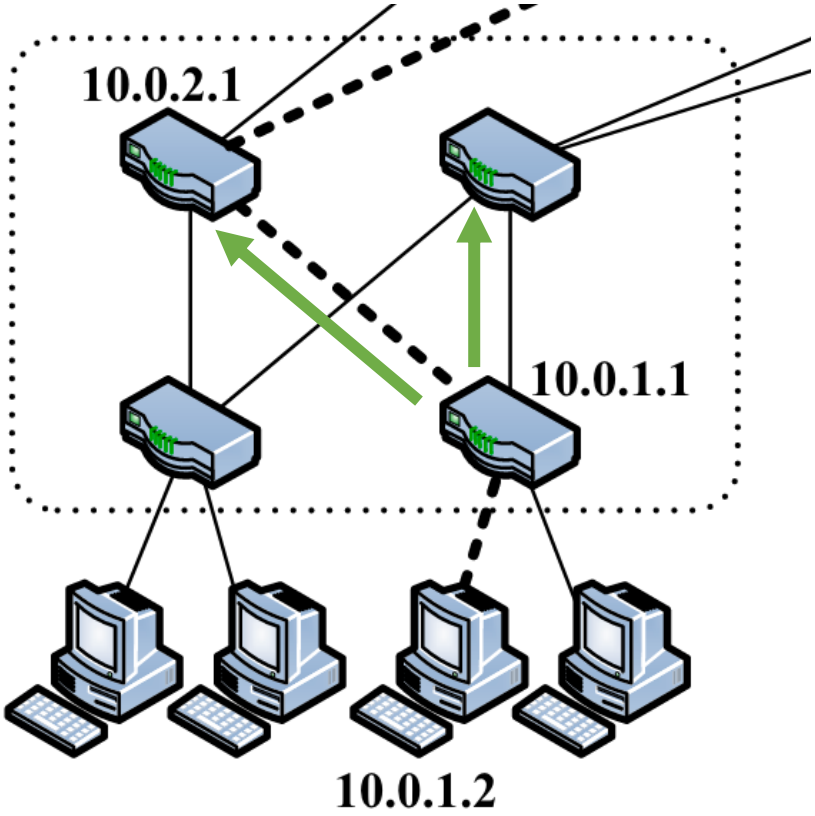
- Core 交换机路由表

```
1 foreach j in [1, (k/2)] do
2     foreach i in [1, (k/2)] do
3         foreach destination pod x in [0, (k/2) − 1] do
4             addPrefix(10.k.j.i,10.x.0.0/16, x);
5         end
6     end
7 end
```

两级路由表分析

- 能根据 IP 地址充分利用多条最短路径，但主机粒度的路由可能导致网络流冲突
- 路由规则固定，不能根据网络负载情况动态调整





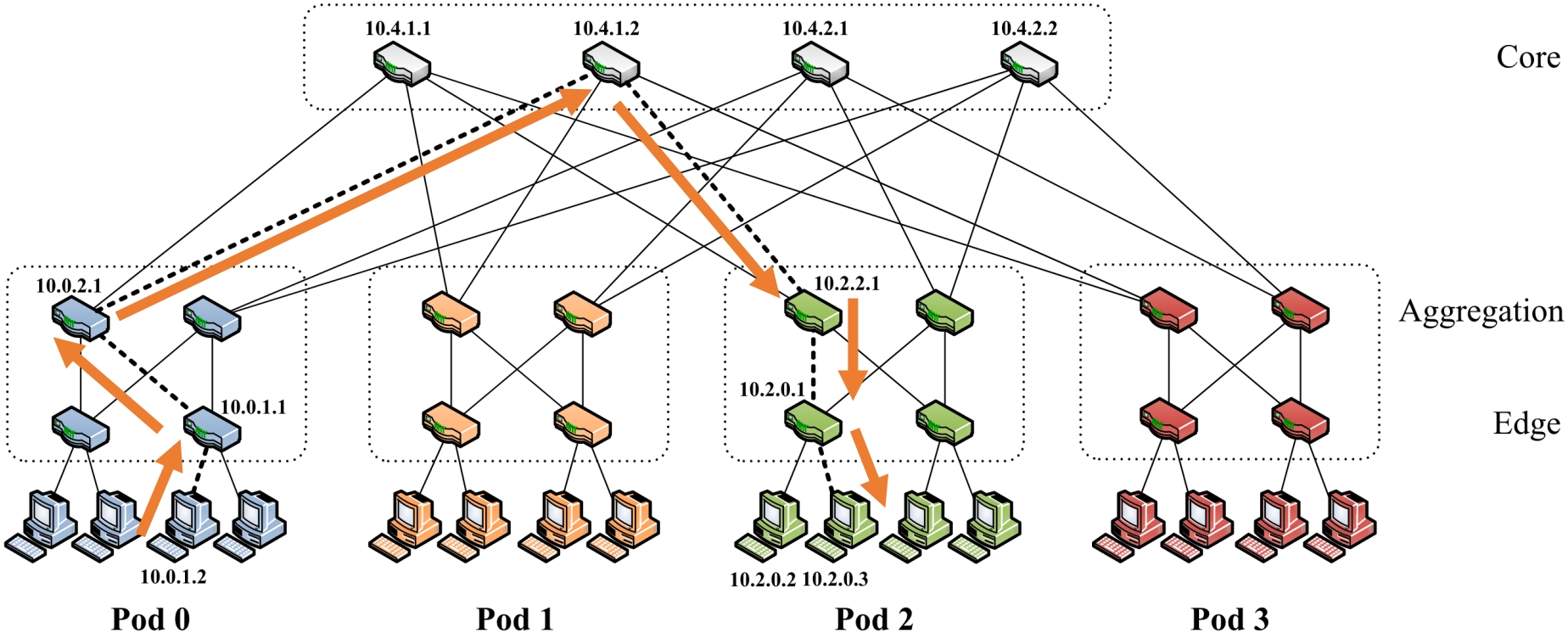
流粒度的路由改进（1）

- 流分类（Flow classification）
  - ✓消除本地阻塞
  - ✓识别同一流的后续数据包，并在同一传出端口上转发它们
  - ✓Edge 交换机定期重新分配流量的输出端口，以最大限度地减少不同端口的总流量之间的差异

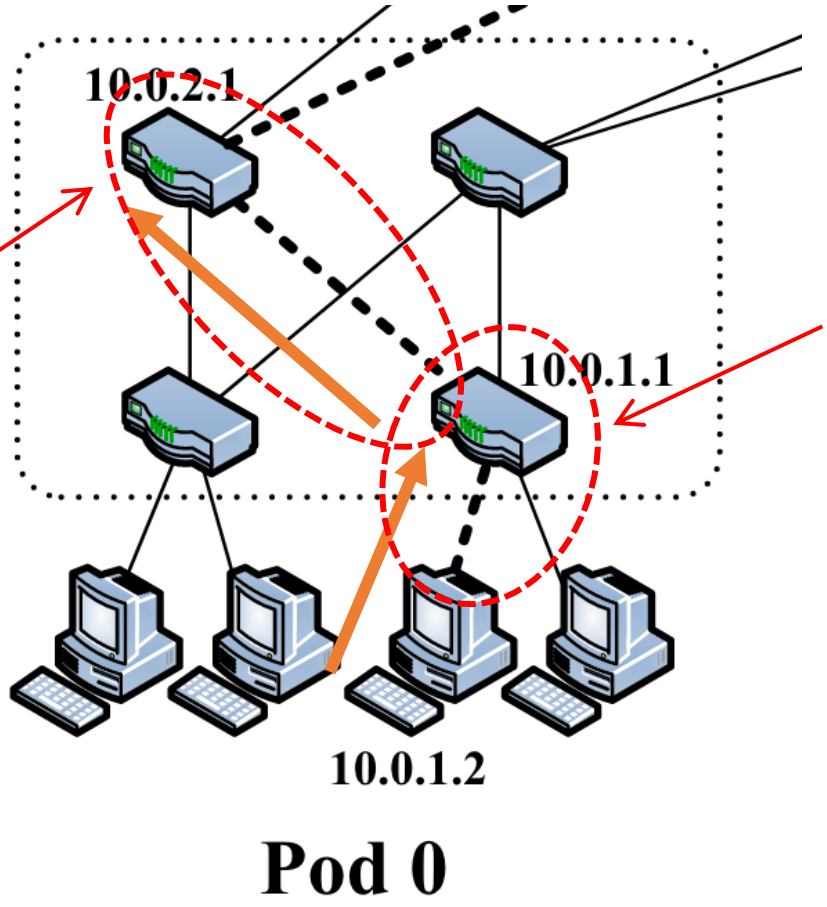
流粒度的路由改进（2）

- 流调度（Flow scheduling）
  - 消除全局阻塞
  - 大流才是决定网络对分带宽的关键
  - Edge 交换机检测大流并发送信号给中央调度器
  - 调度器分配大流路径，防止大流共享相同的链接

网络容错（1）


10.4.1.1
10.4.1.2
10.4.2.1
10.4.2.2
Core
10.0.2.1
10.0.1.1
10.0.1.2
10.2.2.1
10.2.0.1
10.2.0.2 10.2.0.3
Aggregation
Edge
Pod 0
Pod 1
Pod 2
Pod 3

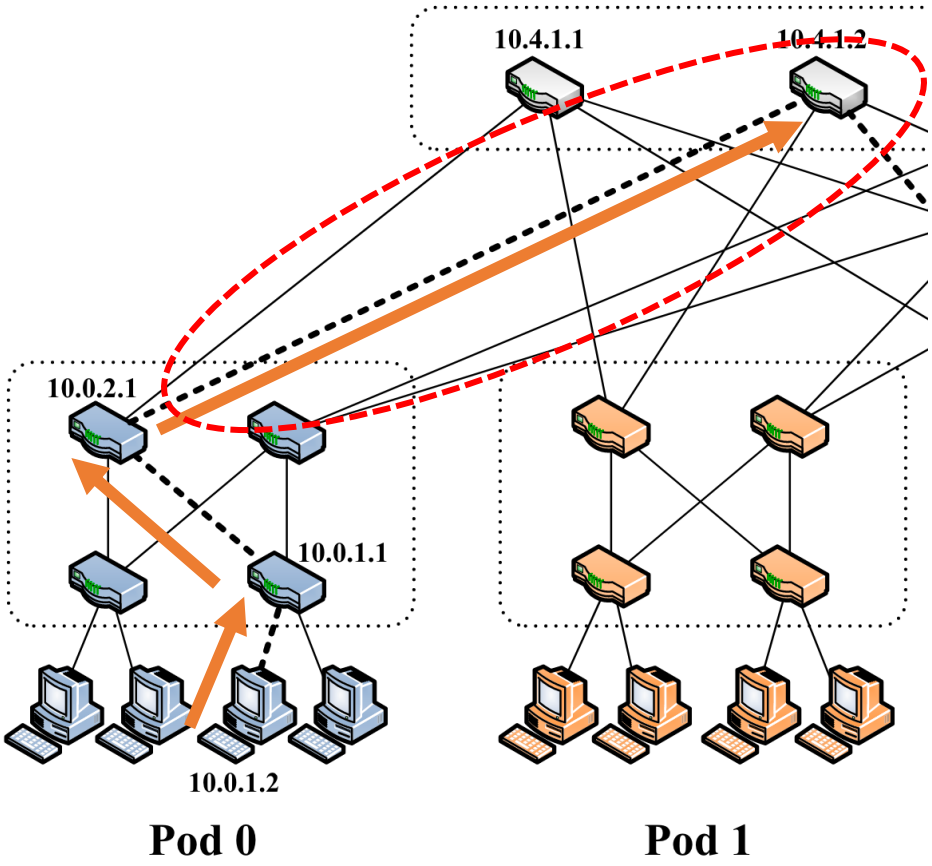
网络容错（2）



此部分故障则 agg 交换机通知 edge 交换机避免通过其发送数据，edge 交换机标记此路径不可用

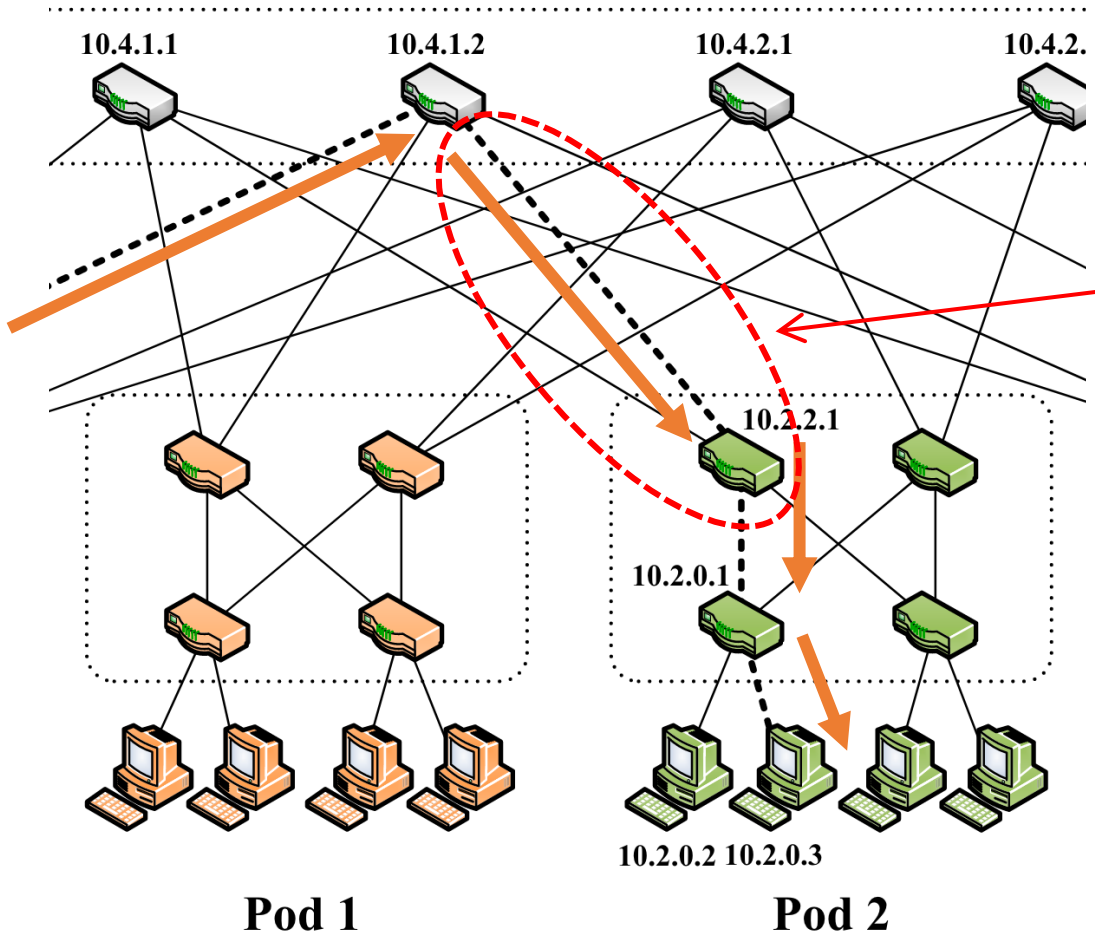
此部分故障则主机无法发送数据，可采用冗余 edge 交换机

网络容错（3）



此部分故障则 core 交换机通知 agg 交换机避免通过其发送数据，agg 交换机标记此路径不可用

网络容错（4）



此部分故障则 agg 交换机通知 core 交换机避免通过其发送数据，core 交换机进而通知其他 agg 交换机

中山大學 软件工程学院

SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

谢谢

陈壮彬

软件工程学院

https://zbchern.github.io/sse316.html