Lecture 7: Data Center Networks

CSE 222A: Computer Communication Networks
Alex C. Snoeren

Thanks: Nick Feamster
Lecture 7 Overview

- Project discussion
- Data Centers overview
- “Fat Tree” paper discussion
Cloud Computing

- Elastic resources
  - Expand and contract resources
  - Pay-per-use
  - Infrastructure on demand

- Multi-tenancy
  - Multiple independent users
  - Security and resource isolation
  - Amortize the cost of the (shared) infrastructure

- Flexible service management
  - Resiliency: isolate failure of servers and storage
  - Workload movement: move work to other locations
Cloud Service Models

- **Software as a Service (Saas)**
  - Provider licenses applications to users as a service
  - e.g., customer relationship management, email, …
  - Avoid costs of installation, maintenance, patches, …

- **Platform as a Service (Paas)**
  - Provider offers software platform for building applications
  - e.g., Google’s App-Engine
  - Avoid worrying about scalability of platform

- **Infrastructure as a Service (Iaas)**
  - Provider offers raw computing, storage, and network
  - e.g., Amazon’s Elastic Computing Cloud (EC2)
  - Avoid buying servers and estimating resource needs
Multi-Tier Applications

- Applications consist of tasks
  - Many separate components
  - Running on different machines

- Commodity computers
  - Many general-purpose computers
  - Not one big mainframe
  - Easier scaling

CSE 222A – Lecture 7: Data Center Networks
Host Virtualization

- Multiple virtual machines on one physical machine
- Applications run unmodified as on real machine
- VM can migrate from one computer to another
VMM Virtual Switches
Top-of-Rack Architecture

- Rack of servers
  - Commodity servers
  - And top-of-rack switch

- Modular design
  - Preconfigured racks
  - Power, network, and storage cabling

- Aggregate to the next level
Extreme Modularity

- Containers

- Many containers
Data Center Challenges

- Traffic load balance
- Support for VM migration
- Achieving bisection bandwidth
- Power savings / Cooling
- Network management (provisioning)
- Security (dealing with multiple tenants)
Data Center Costs

- James Hamilton published basic 2008 breakdown

- Servers: 45%
  - CPU, memory, disk

- Infrastructure: 25%
  - UPS, cooling, power distribution

- Power draw: 15%
  - Electrical utility costs

- Network: 15%
  - Switches, links, transit
Traditional DC Topology

- **Internet**
- **Data Center**
  - **Core**
  - **Aggregation**
  - **Access**
  - **Layer-2 switch**
  - **Layer-2/3 switch**
  - **Layer-3 router**
  - **Servers**
DC Network Requirements

- Scalability
  - Incremental build out?

- Reliability
  - Loop free forwarding

- VM migration

- Reasonable management burden
  - Humans in the loop?
Traditional Topologies

- Over subscription of links higher up in the topology
- Tradeoff between cost and provisioning
- Single point of failure
Capacity Bottlenecks

~ 200:1

~ 40:1

~ 5:1
Management: L2 vs. L3

- Ethernet switching (layer 2)
  - Cheaper switch equipment
  - Fixed addresses and auto-configuration
  - Seamless mobility, migration, and failover

- IP routing (layer 3)
  - Scalability through hierarchical addressing
  - Efficiency through shortest-path routing
  - Multipath routing through equal-cost multipath

- Data centers often connect layer-2 islands by IP routers
Advantages of Layer 2

- Certain monitoring apps require server with same role to be on the same VLAN
- Using same IP on dual homed servers
- Allows organic growth of server farms
- VM migration is easier
Layer 2 Pods w/L3 Backbone

Key
- CR = Core Router (L3)
- AR = Access Router (L3)
- S = Ethernet Switch (L2)
- A = Rack of app. servers

~ 1,000 servers/pod == IP subnet
FAT Tree-Based Solution

- An all Layer-3 solution

- Connect end-host together using a “fat-tree” topology
  - Infrastructure consist of cheap devices
    » Each port supports same speed as endhost
  - All devices can transmit at line speed if packets are distributed along existing paths
  - A k-port fat tree can support $k^{3/4}$ hosts
“Fat-Tree” Topology

Pod 0
10.0.1.1
10.0.1.2

Pod 1
10.4.1.1
10.4.1.2

Pod 2
10.4.2.1
10.4.2.2

Pod 3
10.2.2.1
10.2.0.1
10.2.0.2 10.2.0.3

Core
Aggregation
Edge
Fat-Tree Challenges

- Layer 3 will only use one of the existing equal cost paths
- Packet re-ordering occurs if layer 3 blindly takes advantage of path diversity
  - E.g., ECMP
Modified Fat Tree

- Enforce special addressing scheme in DC
  - Allows host attached to same switch to route only through switch
  - Allows inter-pod traffic to stay within pod
  - unused.PodNumber.switchnumber.Endhost

- Use two level look-ups to distribute traffic and maintain packet ordering.
Two-Level Lookups

- First level is prefix lookup
  - Used to route down the topology to endhost

- Second level is a suffix lookup
  - Used to route up towards core
  - Diffuses and spreads out traffic
  - Maintains packet ordering by using the same ports for the same endhost
Diffusion Optimizations

- Flow classification
  - Eliminates local congestion
  - Assign to traffic to ports on a per-flow basis instead of a per-host basis

- Flow scheduling
  - Eliminates global congestion
  - Prevent long lived flows from sharing the same links
  - Assign long lived flows to different links
Discussion

- VM mobility
- Data center build out
- Internet connectivity
- Address space allocation
  - Doing flow placement anyway?
For Next Class…

- Read and review PortLand paper

- Work on project proposals
  - Details posted to the web page this week