Datacenters

CS 168, Fall 2014
Sylvia Ratnasamy

http://inst.eecs.berkeley.edu/~cs168/
What you need to know

• Characteristics of a datacenter environment
  – goals, constraints, workloads, etc.
• How and why DC networks are different (vs. WAN)
  – e.g., latency, geo, autonomy, ...
• How traditional solutions fare in this environment
  – e.g., IP, Ethernet, TCP, ARP, DHCP
• Specific design approaches we cover in class
  – next lecture
Disclaimer

• Material is emerging (not established) wisdom

• Material is incomplete
  – many details on how and why datacenter networks operate aren’t public
Plan

Today
- Characteristics and goals of datacenter networks
- Focus on differences relative to the Internet

Next lecture
- Emerging solutions
What goes into a datacenter (network)?

- Servers organized in racks
What goes into a datacenter (network)?

- Servers organized in racks
- Each rack has a ‘Top of Rack’ (ToR) switch
What goes into a datacenter (network)?

- Servers organized in racks
- Each rack has a `Top of Rack’ (ToR) switch
- `Aggregation switches interconnect ToR switches
What goes into a datacenter (network)?

- Servers organized in racks
- Each rack has a `Top of Rack’ (ToR) switch
- `Aggregation switches interconnect ToR switches
- Connected to the outside via `core’ switches
  – note: blurry line between aggregation and core
- With 2x redundancy for fault-tolerance
E.g., Brocade Reference Design
E.g., Cisco Reference Design

Internet

\[ \text{\textsc{CR}} \] \quad \text{\textsc{CR}}

\[ \text{\textsc{AR}} \] \quad \text{\textsc{AR}} \quad \ldots \quad \text{\textsc{AR}} \quad \text{\textsc{AR}}

\[ \text{\textsc{S}} \quad \text{\textsc{S}} \]

\[ \text{\textsc{S}} \quad \text{\textsc{S}} \quad \text{\textsc{S}} \quad \text{\textsc{S}} \]

\[ \text{\textsc{A}} \quad \text{\textsc{A}} \quad \ldots \quad \text{\textsc{A}} \quad \text{\textsc{A}} \quad \text{\textsc{A}} \quad \ldots \quad \text{\textsc{A}} \]

\[ \sim 40-80 \text{ servers/rack} \]
Datacenters have been around for a while

1961, Information Processing Center at the National Bank of Arizona
What’s new?
What’s new?

• Scale

• Applications
  – Large-scale computations ("big data")
  – Customer-facing, revenue generating services

• Service model
  – Clouds (jargon: SaaS, PaaS, DaaS, IaaS, …)
  – Multi-tenancy
SCALE!
How big exactly?

- 1M servers/site [Microsoft/Amazon/Google]
- > $1B to build one site [Facebook]
- >$20M/month/site operational costs [Microsoft ’09]

But only O(10-100) sites
Implications (1)

• Scale
  - Need scalable designs (duh): e.g., avoid flooding
  - Low cost designs: e.g., use commodity technology
  - High utilization (efficiency): e.g., >80% avg. utilization
    • Contrast: avg. utilization on Internet links often ~30%
  - Tolerate frequent failure
    • Large number of (low cost) components
  - Automate
Implications (2)

- Service model: clouds / multi-tenancy
  - performance guarantees
  - isolation guarantees
  - portability

- How?
  - “network virtualization” (lecture on SDN)
Applications

• Common theme: parallelism
  – Applications decomposed into tasks
  – Running in parallel on different machines

• Two common paradigms
  – “Partition Aggregate”
  – Map Reduce
Partition-Aggregate

user requests from the Internet

Router / Load Balancer

Front-End Proxy

Web Server

Data Cache

Data Cache

Database

Database
“North – South” Traffic

• Interactive / query-response exchange between external clients and datacenter
• Handled by front-end (web) servers, mid-tier application servers, and back-end databases
Map-Reduce

Distributed Storage

Map Tasks

Reduce Tasks

Distributed Storage
Map-Reduce

Distributed Storage

Map Tasks

Reduce Tasks

Distributed Storage

Often doesn’t cross the network

Some fraction (typically 2/3) crosses the network

Always goes over the network
“East-West” Traffic
“East-West” Traffic

- Traffic between servers in the datacenter
- Communication within “big data” computations
- Traffic may shift on small timescales (< minutes)
Common traffic pattern: “Elephants” and “Mice”

- Web search, data mining (Microsoft) [Alizadeh 2010]
Implications (3)

- Applications
  - High bandwidth any-to-any communication ("bisection bandwidth")
  - Low latency is critical
  - Worst-case ("tail") latency is critical
High Bandwidth

• Ideal: Each server can talk to any other server at its full access link rate

• Conceptually: DC network as one giant switch
DC Network: Just a Giant Switch!

Slides from: Alizadeh, HotNets 2012
DC Network: Just a Giant Switch!
DC Network: Just a Giant Switch!

Slides from: Alizadeh, HotNets 2012
DC Network: Just a Giant Switch!

Slides from: Alizadeh, HotNets 2012
High Bandwidth

• Ideal: Each server can talk to any other server at its full access link rate

• Conceptually: DC network as one giant switch
  – Would require a 10 Pbits/sec switch!
    • 1M ports (one port/server)
    • 10Gbps per port

• Practical approach: build a network of switches (“fabric”) with high “bisection bandwidth”
  – Each switch has practical #ports and link speeds
Bisection Bandwidth

• Partition a network into two equal parts
• Minimum bandwidth between the partitions is the bisection bandwidth

• **Full** bisection bandwidth: bisection bandwidth in an $N$ node network is $N/2$ times the bandwidth of a single link

  → **nodes of any two halves can communicate at full speed with each other.**
Achieving Full Bisection Bandwidth

\[ O(40 \times 10 \times 100) \text{ Gbps} \]

\[ O(40 \times 10) \text{ Gbps} \]

10Gbps links

\[ \sim 40 \text{ servers/rack} \]

“scale up” approach
Achieving Full Bisection Bandwidth

• Problem: “Scaling up” a traditional tree topology is expensive!
  – requires non-commodity / impractical / link and switch components

• Solutions?
  – Over-subscribe (i.e., provision less than full BBW)
  – Better topologies
Oversubscription

Need techniques to avoid congesting oversubscribed links!
Better topologies?

“scale up” approach

“scale out” approach
Better topologies?

• E.g., ‘Clos’ topology
  – Multi-stage network
  – All switches have k ports
  – k/2 ports up, k/2 down
• E.g., with 3 stages, k=48
  – $k^3/4$ hosts = 27,648 servers
• All links have same speed

“scale out” approach
Challenges in scale-out designs?

• **Topology** offers high bisection bandwidth

• All other system components must be able to exploit this available capacity
  – Routing must use all paths
  – Transport protocol must fill all pipes (fast)
Low Latency

Two (related) issues:

1) Very low RTTs within the DC (approaching 1µsec)

   Implications?
   - BW x delay: 10Gbps x 1µsec = 10000 bits = 2.5 packets
   - Consider TX 500B @ 10Gbps = 0.4µs per hop = 2µs if a packet traverses 5 hops and waits behind one packet at every hop
   - What does this mean for buffering and switch design?
   - What does this mean for congestion control?
Low Latency

Two (related) issues:

1) Very low RTTs within the DC (approaching $1\mu$sec)

2) Applications want low latency
   - predictable / guaranteed bounds on flow completion time, including the worst-case!
     (recall: `best effort’ vs. `guaranteed service’ debates)
   - How is still an open question
What’s different about DC networks?

**Characteristics**

• Huge scale
  – ~20,000 switches/routers
  – *contrast*: AT&T ~500 routers
What’s different about DC networks?

Characteristics

• Huge scale

• Limited geographic scope
  – High bandwidth: 10/40/100G (Contrast: DSL/WiFi)
  – Very low RTT: 1-10s μsecs. (Contrast: 100s msecs)
What’s different about DC networks?

Characteristics

• Huge scale
• Limited geographic scope
• Limited heterogeneity
  – link speeds, technologies, latencies, …
What’s different about DC networks?

Characteristics

• Huge scale
• Limited geographic scope
• Limited heterogeneity
• Regular/planned topologies (e.g., trees)
  – Contrast: ad-hoc evolution of wide-area topologies
What’s different about DC networks?

Goals

• Extreme bisection bandwidth requirements
  – recall: all that east-west traffic
  – target: any server can communicate at its full link speed
  – How: next lecture
What’s different about DC networks?

**Goals**

- Extreme bisection bandwidth requirements
- Extreme latency requirements
  - real money on the line
  - current target: 1μs RTTs
  - how? Next lecture
What’s different about DC networks?

Goals

- Extreme bisection bandwidth requirements
- Extreme latency requirements
- *Predictable, deterministic* performance
  - “your packet will reach in Xms, or not at all”
  - “your VM will always see at least YGbps throughput”
  - How is still an open question
What’s different about DC networks?

Goals

• Extreme bisection bandwidth requirements
• Extreme latency requirements
• *Predictable, deterministic* performance
• Differentiating between tenants is key
  – e.g., “No traffic between VMs of tenant A and tenant B”
  – “Tenant X cannot consume more than XGbps”
  – “Tenant Y’s traffic is low priority”
• How: lecture on SDN (Nov 24)
What’s different about DC networks?

**Goals**

- Extreme bisection bandwidth requirements
- Extreme latency requirements
- *Predictable, deterministic* performance
- Differentiating between tenants is key
- Scalability (of course)
What’s different about DC networks?

Goals

• Extreme bisection bandwidth requirements
• Extreme latency requirements
• *Predictable, deterministic* performance
• Differentiating between tenants is key
• Scalability (of course)
• Cost/efficiency
  – focus on commodity solutions, ease of management
What’s different about DC networks?

New degrees of (design) freedom

• Single administrative domain
  – Can deviate from standards, invent your own, etc.
  – “Green field” deployment is still feasible
What’s different about DC networks?

New degrees of (design) freedom

- Single administrative domain
- Control over network \textit{and} endpoint(s)
  - can change (say) addressing, congestion control, \textit{etc.}
  - can add mechanisms for security/policy/etc. at the endpoints (typically in the hypervisor)
What’s different about DC networks?

New degrees of (design) freedom

• Single administrative domain
• Control over network and endpoint(s)
• Control over the placement of traffic source/sink
  – e.g., map-reduce scheduler chooses where tasks run
  – Can control what traffic crosses which links
Summary

• Recap: datacenters
  – new characteristics and goals
  – some liberating, some constraining
  – scalability is the baseline requirement
  – more emphasis on performance
  – less emphasis on heterogeneity
  – less emphasis on interoperability