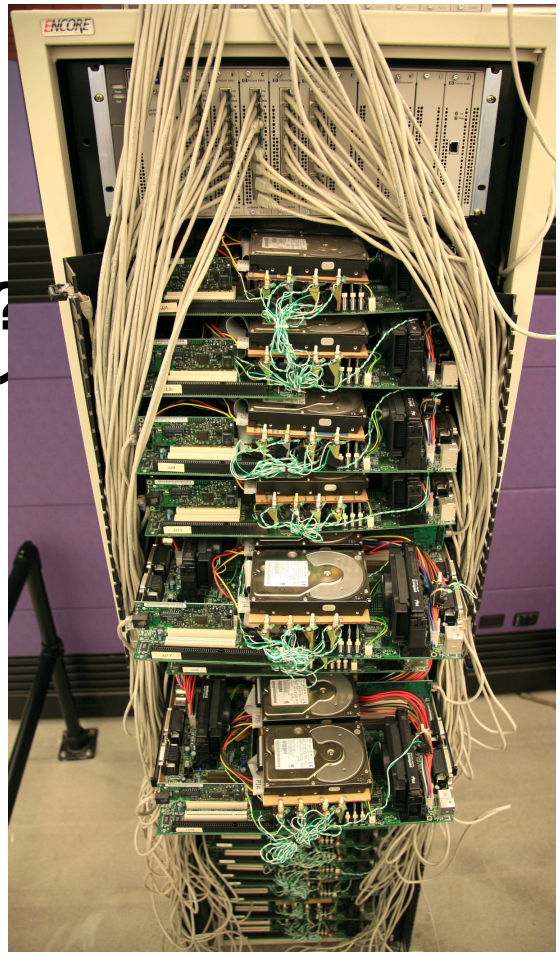
TECHNISCHE HOCHSCHULE DEGGENDORF

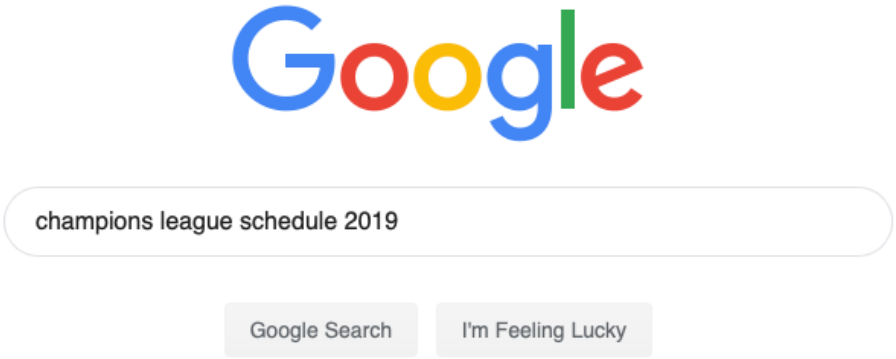
# Datacenter Network Programming

## Introduction to Datacenter Networking

# WebSearch – Whats behind the scenes

- What happens when we search the Web?
- How does a search engine find results so fast?
- Do we crawl the whole Web in real time?
- Can we store the whole web on a single server?
- How many machines do we need for that? 10? 1

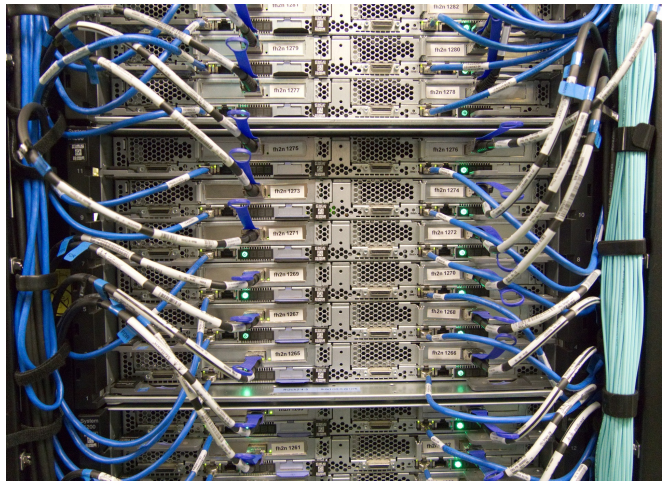



https://www.flickr.com/photos/44124348109@N01/157722937

# Answer: We use Data centers to search the Web

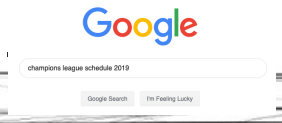
- Warehouse computing on millions of commodity
- Have massive amount of CPU and disc
- Carry massive amount of traffic



Tight Integration between Compute, Storage, Network

# The journey of a websearch

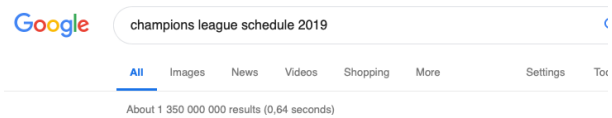
Request






→ Many Co-Flows and multiple parallel TCP incasts

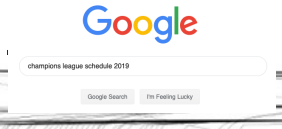
Response




Source: Talk on “Speeding up Distributed Request-Response Workflows” by Virajith Jalaparti at ACM SIGCOMM’13

# The journey of a websearch

Request

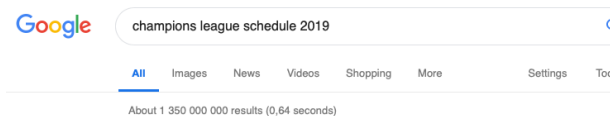



Datacenter Networking performance is crucial

- Google: 20% traffic reduction from an extra 500 ms of latency,
- Amazon: every additional 100ms of latency costs 1% loss in revenue

→ Many Co-Flows and multiple parallel TCP incasts

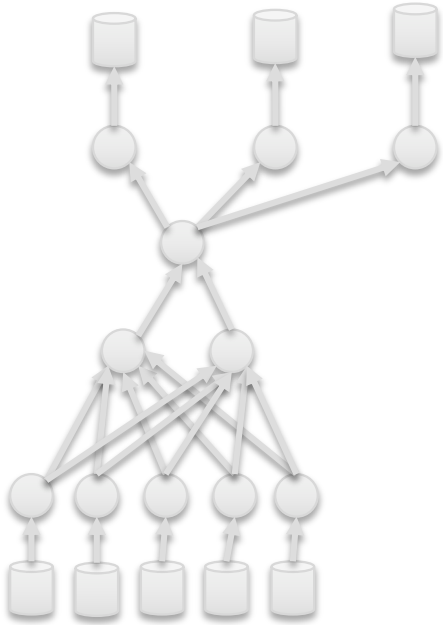
Response




Source: Talk on “Speeding up Distributed Request-Response Workflows” by Virajith Jalaparti at ACM SIGCOMM’13

# In Data Center, Communication is crucial

- **Application Performance for Data Center Apps crucial for the service to be accepted**
- **Facebook analytics jobs spend 33% of their runtime in communication**
- ***As in-memory systems proliferate, the network MUST scale for not become the bottleneck***
- ***Appliction flow:***
  - A sequence of packets between two endpoints
  - Independent unit of allocation, sharing, load balancing, and prioritization

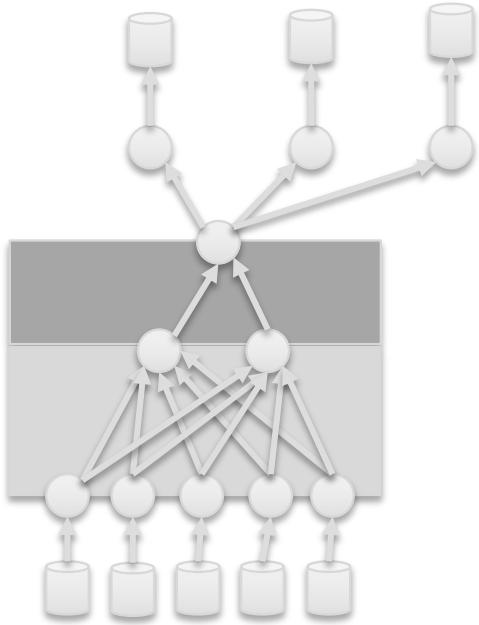


| Framework | # peers |
|---|---|
| Spark | 6 |
| Hadoop | 10 |
| Yarn | 20 |

# Coflow and performance

- **Coflow Definition:**
  - A collection of parallel flows with distibuted endpoints
  - Each flow within the CoFlow set is independent
- **Performance Aspects**
  - Job completion time depends on the last flow within a CoFlow set to complete!
  - Many Incoming flows lead to TCP Incast problem
    - Switch attached to server typically has small buffers
    - many incoming small requests may lead to buffer overflow

  → Switch drops packets → bursty retransmit and TCP RTO
    - May result in idle time and 90% throughput drop



| Framework | # peers |
|---|---|
| Spark | > 6 |
| Hadoop | > 10 |
| Yarn | > 20 |

# What is important to a data center?

- High availability
- High efficiency
- Smart agility
- Security

# Datacenter main characteristic: elasticity

- **Manage the Workload!**
  - Allows to rapidly install service capacity to scale with demands
  - Virtual Machines, docker containers, disk images → deploy, migrate
  - Re-assign Resources to match demands → horizontal and vertical scaling
- **Manage the storage!**
  - Allows a server access to persistent data
  - Highly distributed filesystems (e.g. HDFS) and key-value stores (e.g. Memcache)
- **Manage the Network!**
  - Allows a server to communicate with other servers, regardless of location
  - Vision: provide remote data access as fast as local, many techniques inside OS → RDMA, Infiniband Verbs, kernel bypass, DPDK, … → How about the network?

# Datacenter traffic characteristics

**Inside the Social Network's (Datacenter) Network**

ACM SIGCOMM 2015

Arjun Roy, Hongyi Zeng†, Jasmeet Bagga†, George Porter, and Alex C. Snoeren

Department of Computer Science and Engineering
University of California, San Diego

†Facebook, Inc.

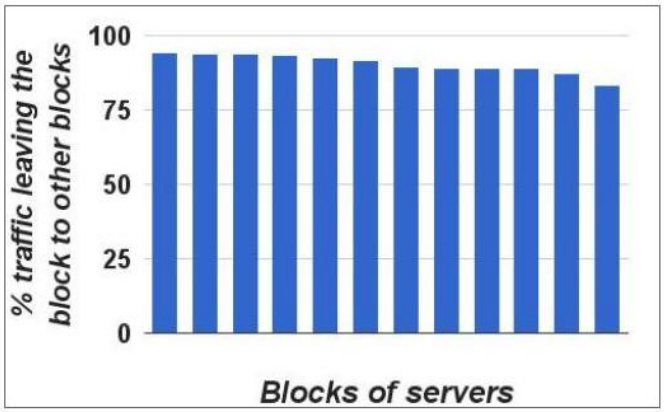
| Locality | All | Hadoop | FE | Svc. | Cache | DB |
|---|---|---|---|---|---|---|
| Rack | 12.9 | 13.3 | 2.7 | 12.1 | 0.2 | 0 |
| Cluster | 57.5 | 80.9 | 81.3 | 56.3 | 13.0 | 30.7 |
| DC | 11.9 | 3.3 | 7.3 | 15.7 | 40.7 | 34.5 |
| Inter-DC | 17.7 | 2.5 | 8.6 | 15.9 | 16.1 | 34.8 |
| Percentage | | 23.7 | 21.5 | 18.0 | 10.2 | 5.2 |

**Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network**

ACM SIGCOMM 2015

Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat
Google, Inc.
jupiter-sigcomm@google.com

| Job Category | B/w (%) |
|---|---|
| Storage | 49.3 |
| Search Serving | 26.2 |
| Mail | 7.4 |
| Ad Stats | 3.8 |
| Rest of traffic | 13.3 |



# Concurrent flows

**Inside the Social Network's (Datacenter) Network**

ACM SIGCOMM 2015

Arjun Roy, Hongyi Zeng†, Jasmeet Bagga†, George Porter, and Alex C. Snoeren

Department of Computer Science and Engineering
University of California, San Diego

†Facebook, Inc.

- ***"Web servers and cache hosts have 100s to 1000s of concurrent connections"***
- ***"Hadoop nodes have approximately 25 concurrent connections on average."***
- ***"median inter-arrival times of approximately 2ms"***

**The Nature of Datacenter Traffic: Measurements & Analysis**

ACM IMC 2009

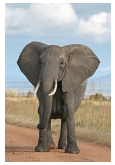
Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, Ronnie Chaiken

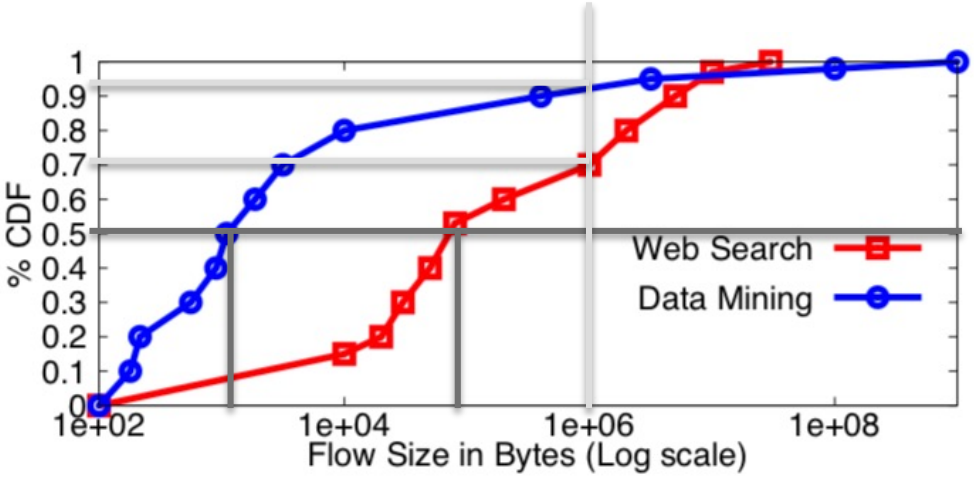
Microsoft Research

- ***"median numbers of correspondents for a server are two (other) servers within its rack and four servers outside the rack"***

# Datacenter traffic charcteristics

- **Different application flow types**
  - Large Flows (Elephant)
    - Few, carry most volume
  - Small Flows (Mice)
    - Many, small volume in total
- **Traffic patterns**
  - Highly volatile
    - Changing rapidly even during a day
  - Highly unpredictable
    - Weak correlation






**HULA: Scalable Load Balancing Using Programmable Data Planes**

*SOSR'16, March 14–15, 2016, Santa Clara, CA, USA*

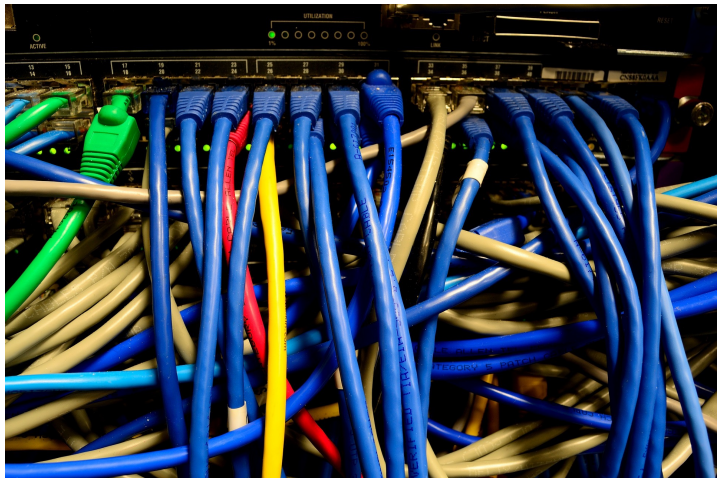
Naga Katta*, Mukesh Hira†, Changhoon Kim‡, Anirudh Sivaraman+, Jennifer Rexford*
*Princeton University, †VMware, ‡Barefoot Networks, +MIT CSAIL
{nkatta, jrex}@cs.princeton.edu, mhira@vmware.com, chang@barefootnetworks.com, anirudh@csail.mit.edu

Traffic-aware optimization needs to be done frequently and rapidly

IMAGES FROM WIKIPEDIA

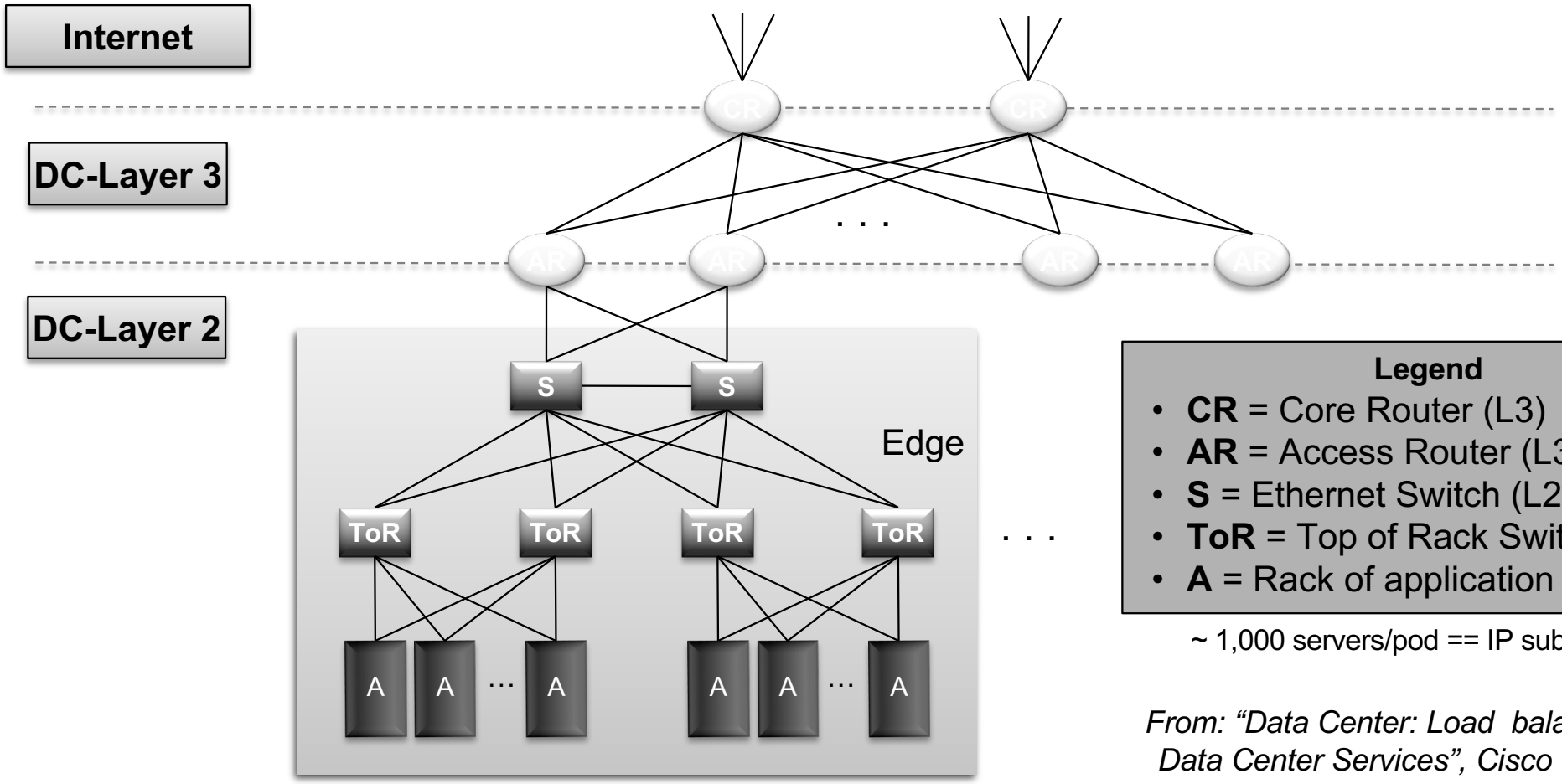
# Datacenter Network

- **Connects servers with each other and the outside world**
- **Built to optimize cost and performance**
- **Tiered Architecture**
  - 3 layers; edge, aggregation, core
  - Cheap devices at edges and expensive devices at core
  - Devices typically proprietary and closed
- **Over-subscription of links closer to the core**
  - Fewer links towards core reduce cost
  - Trade loss/delay for fewer devices and links



Data Center Network must provide High Capacity at Ultra Low Latency

# Conventional Datacenter Network




**Legend**

- **CR** = Core Router (L3)
- **AR** = Access Router (L3)
- **S** = Ethernet Switch (L2)
- **ToR** = Top of Rack Switch (L2)
- **A** = Rack of application servers

~ 1,000 servers/pod == IP subnet

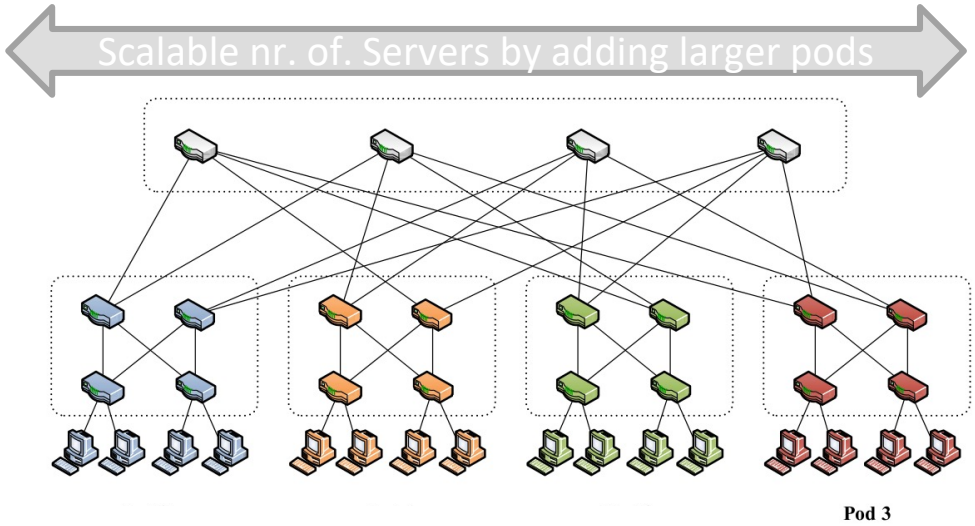
*From: "Data Center: Load balancing Data Center Services", Cisco 2004*

# Conventional Datacenter Network Problems

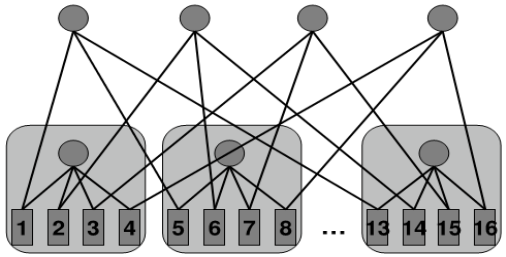
- **Layer 2 Problems**
  - Broadcast must be limited (ARP)
  - Spanning tree protocol does not scale well
- **Layer 3 problems**
  - Complex configuration
  - Cannot migrate containers and VMs easily without changing IP-@
- **Limited server-to-server capacity due to oversubscription**
  - High port number switches expensive → oversubscription
  - Oversubscription ratio: Ratio of ports facing downwards vs. ports facing upwards (with equal bandwidth ports)
    - E.g. Access to core layer: 4:1 oversubscribed, ToR to access 20:1 oversubscribed
    - Switch uplinks get heavily loaded → microbursts and congestion

# Example topology design

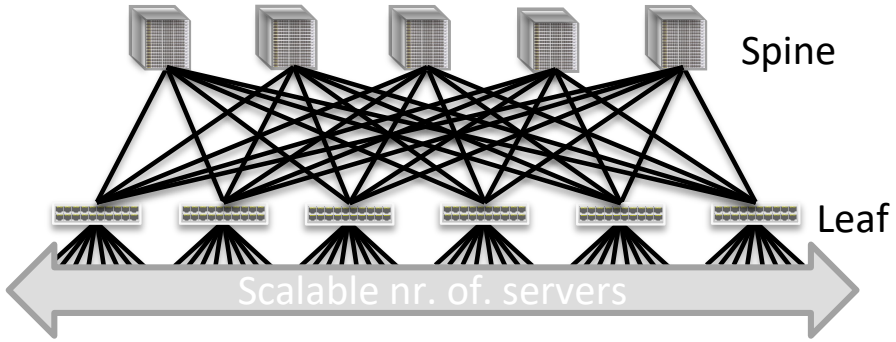
Multi-rooted trees, e.g. Fat-tree [SIGCOMM'08]



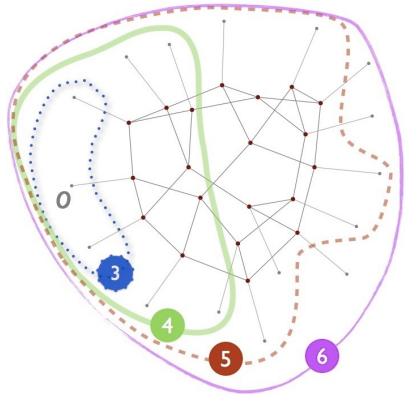

Bcube [SIGCOMM'10]




Leaf-Spine (2-Tier Clos Network)




Jellyfish (random) [NSDI'12]

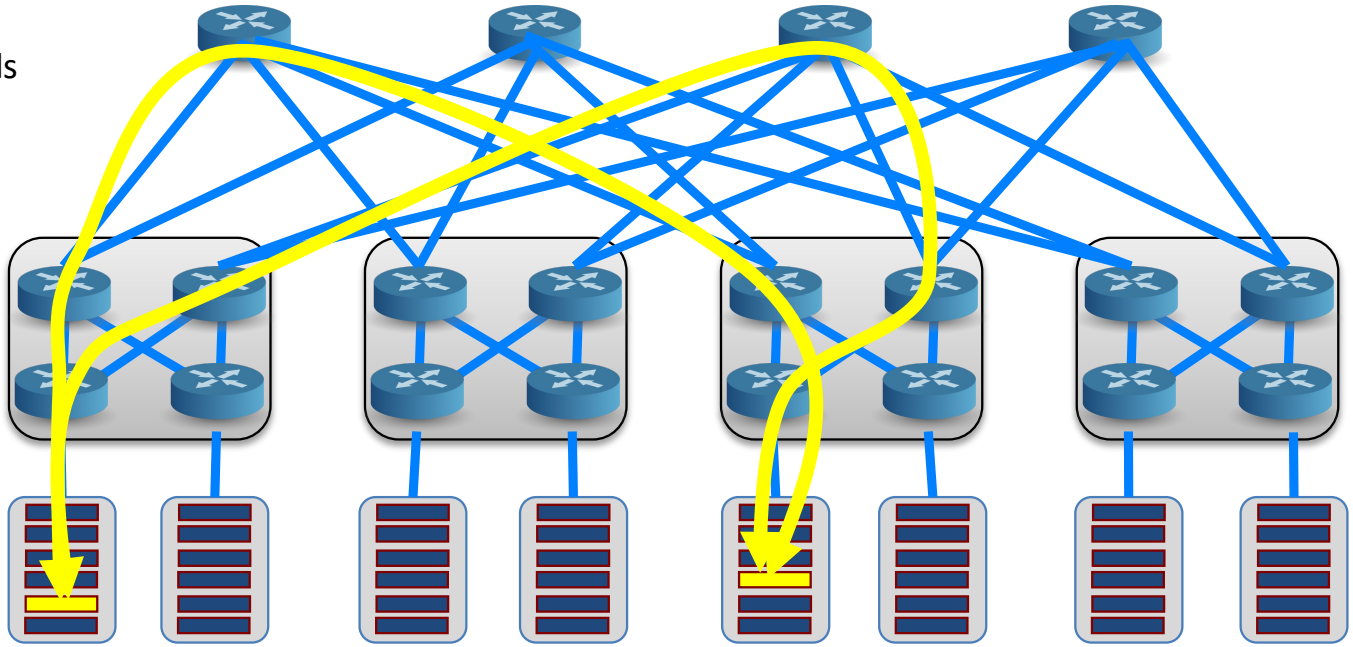


# Multirooted tree properties

Multi-rooted tree [Fat-tree, Leaf-Spine, …] achieve full bisection bandwidth assuming perfect multipathing

Charles E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE Transactions on Computers, Vol. 34, no. 10, Oct. 1985, pp. 892-901.

$(k/2)^2$ core switches
each connects to k pods
→ Supports $k^3/4$ hosts
→ Example: k=4



Core
K Pods
with each K
Switches

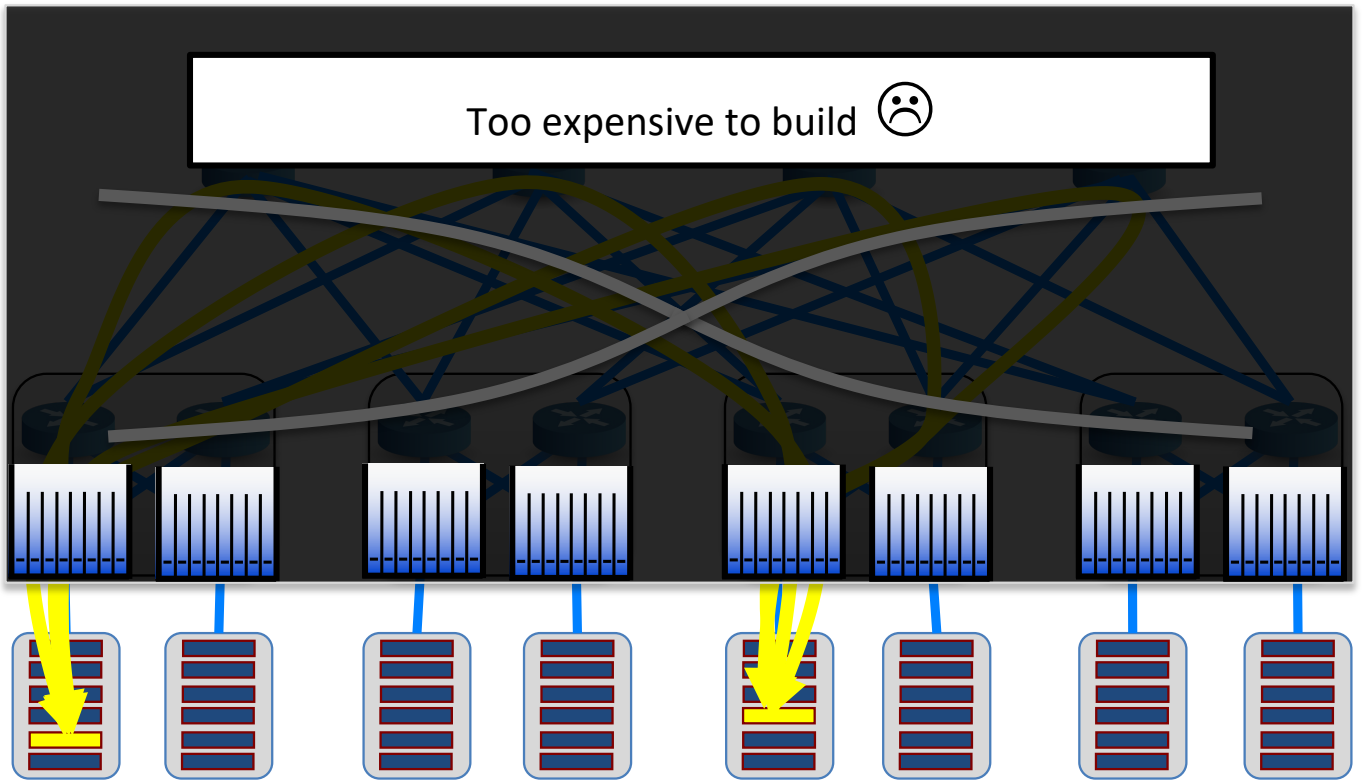
Agg

Access

Server
Racks

# Multirooted tree properties

Multi-rooted tree [Fat-tree, Leaf-Spine, …] try to approximate one big switch abstraction



# Approximating Big Switch

- **Google 2004: Cluster Routers (CR)**
  - 512 ports → expensive
  - Special purpose
  - Oversubscription → bottleneck for intra-rack
- **Challenge**
  - How can we increase capacity?
  - How can we approximate one big switch with cheap commodity switches using clever load-balancing?

**Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network**

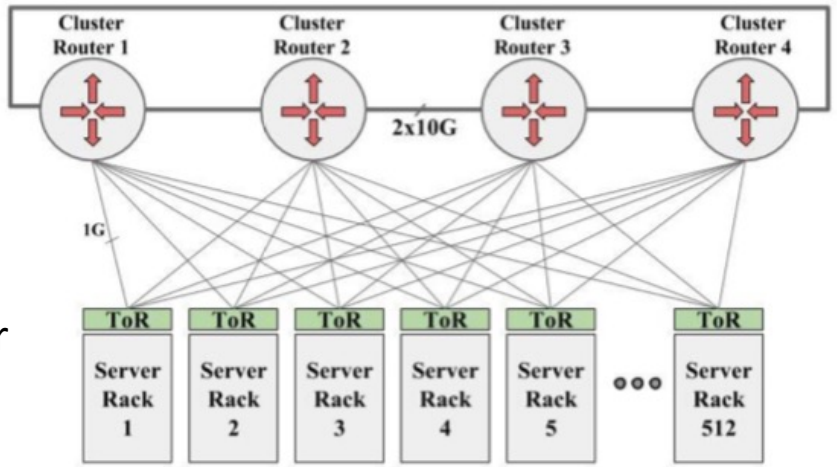
ACM SIGCOMM 2015

Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat
Google, Inc.
jupiter-sigcomm@google.com

Cluster Capacity: 20k servers

2 x 10 G to CR
512 x 1G to ToR

4 x 1G to CR
40x 1G to server

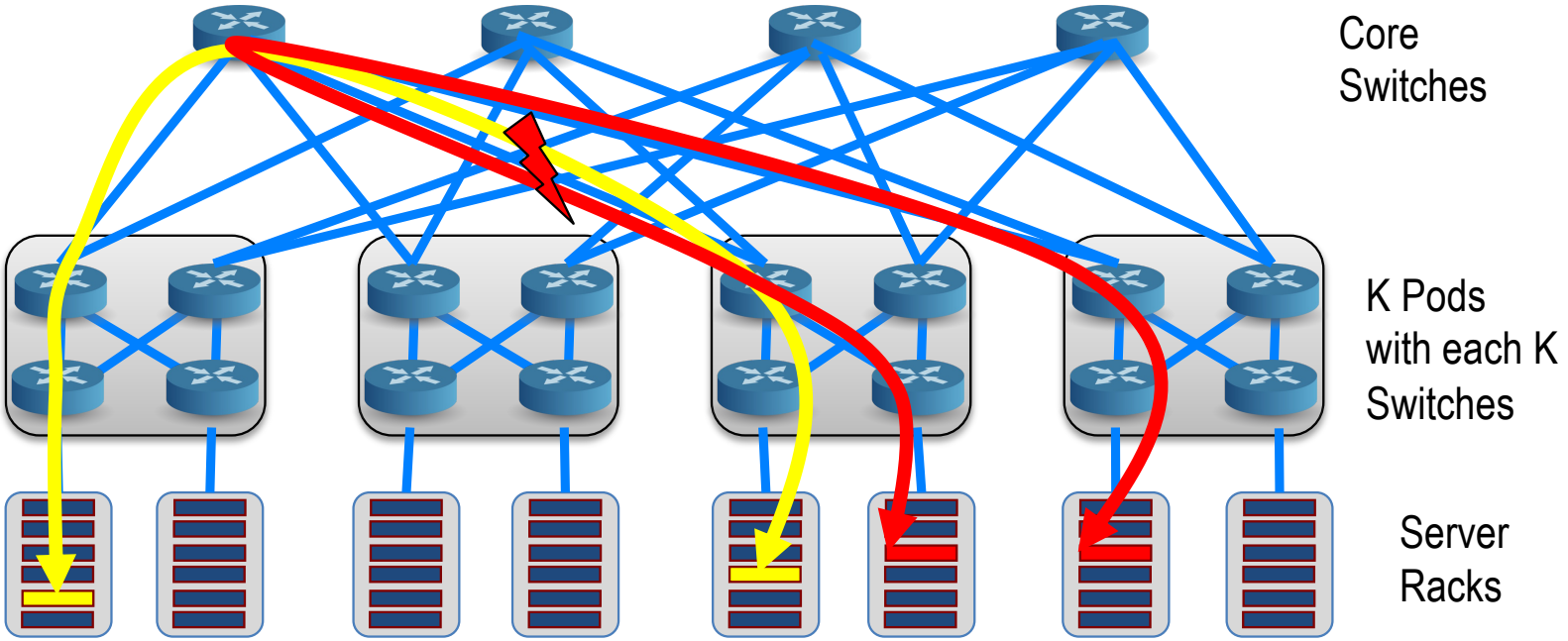


# Multirooted tree problems

**With standard routing, can only exploit a single path**

**Flow collisions result in suboptimal traffic distribution**

**Packet based load-balancing results in reordering (TCP!)**



# Implications for Datacenter Networking

- MASSIVE internal network traffic
- Complex network shared by many applications
- Tight deadlines for network I/O → business impact
- Congestion and TCP incast due to many Co-Flows
- Low latency and high capacity at same time
- Need highly adaptive localized capacity on demand → fine loadbalancing
- cheap, scalable, fault tolerance, ...

# Next lecture

- **Data Center Load Balancing**
  - Network layer
  - ECMP → you will implement in P4
  - Conga
- **P4 based load balancing**
  - Hula → you will (partly) implement in P4
  - MP-Hula Hula with support for Multipath transport protocols)