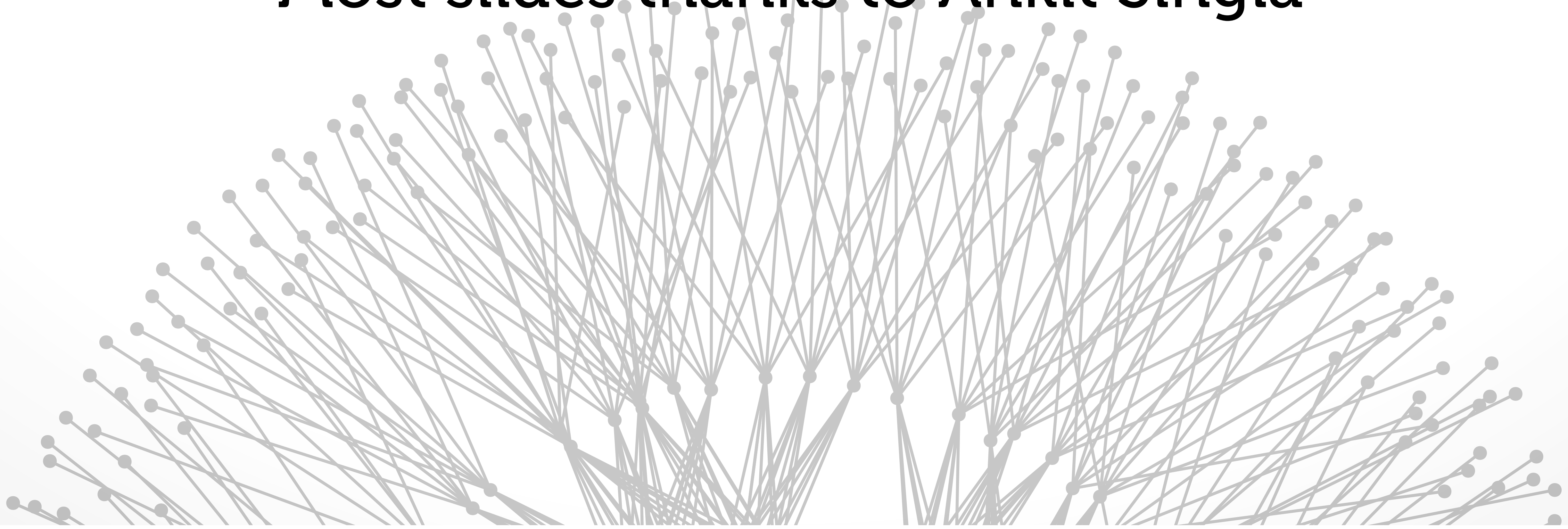
Cloud Workloads & Performance

Brighten Godfrey
CS 538 April 9 2018

Most slides thanks to Ankit Singla



Applications and network traffic

Dropbox

coursera

amazon

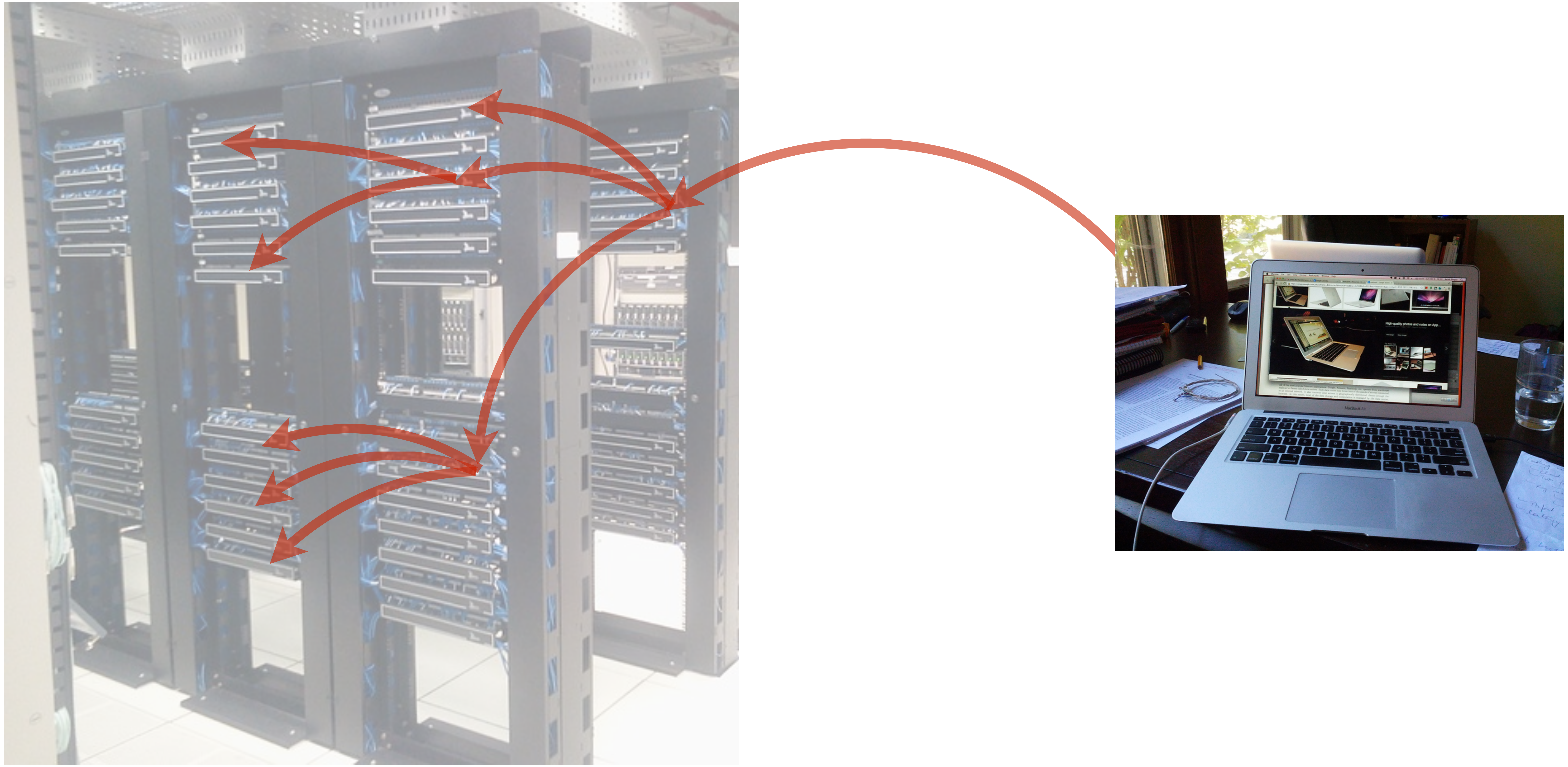
Google

NETFLIX

bing

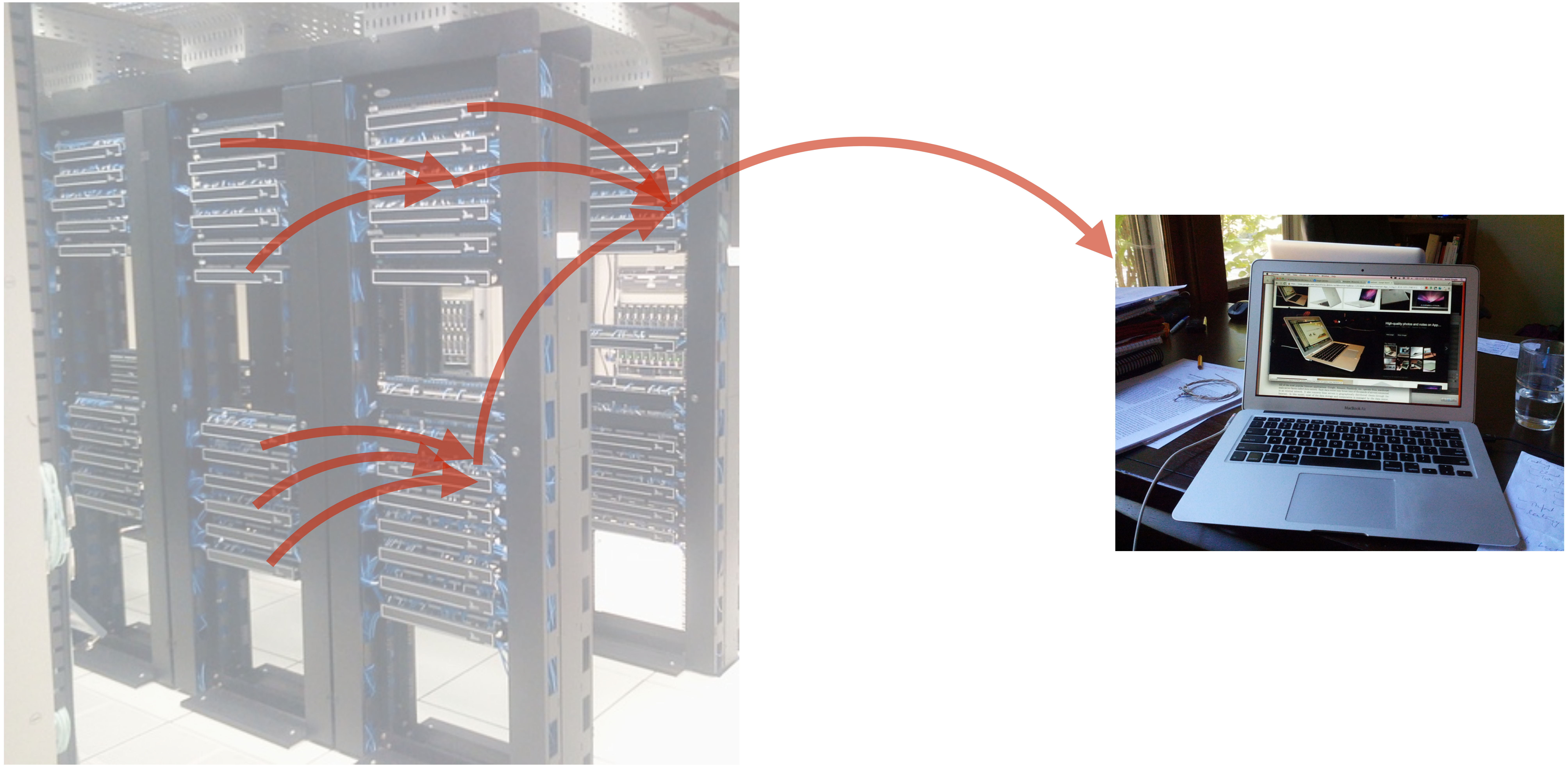
How a Web search works

How a Web search works



Scatter-gather traffic pattern

How a Web search works



Extremely short response deadlines for each server — 10ms

"Up to 150 stages, degree of 40, path lengths of 10 or more"

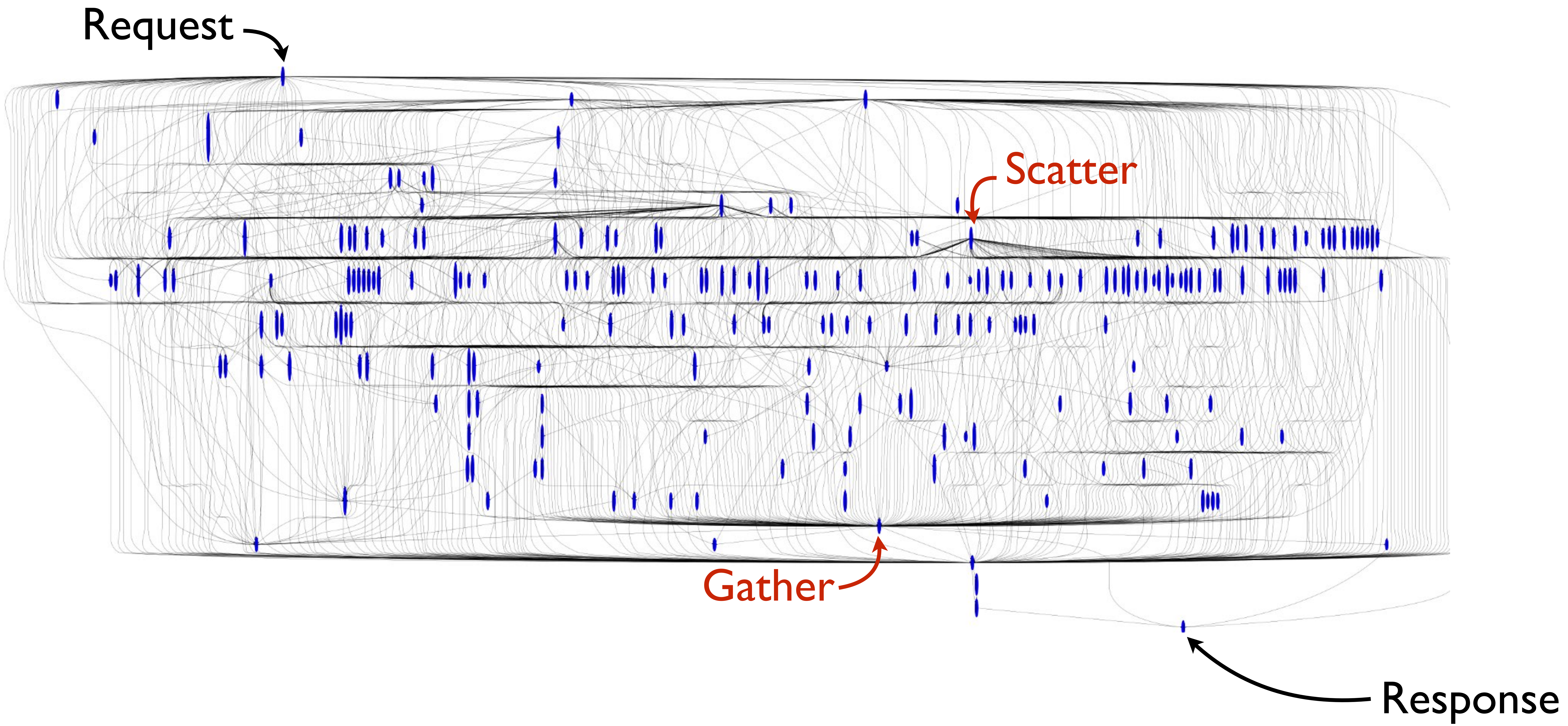


Image source: Talk on "Speeding up Distributed Request-Response Workflows"
by Virajith Jalaparti at ACM SIGCOMM'13

Other Web application traffic

USENIX NSDI, 2013

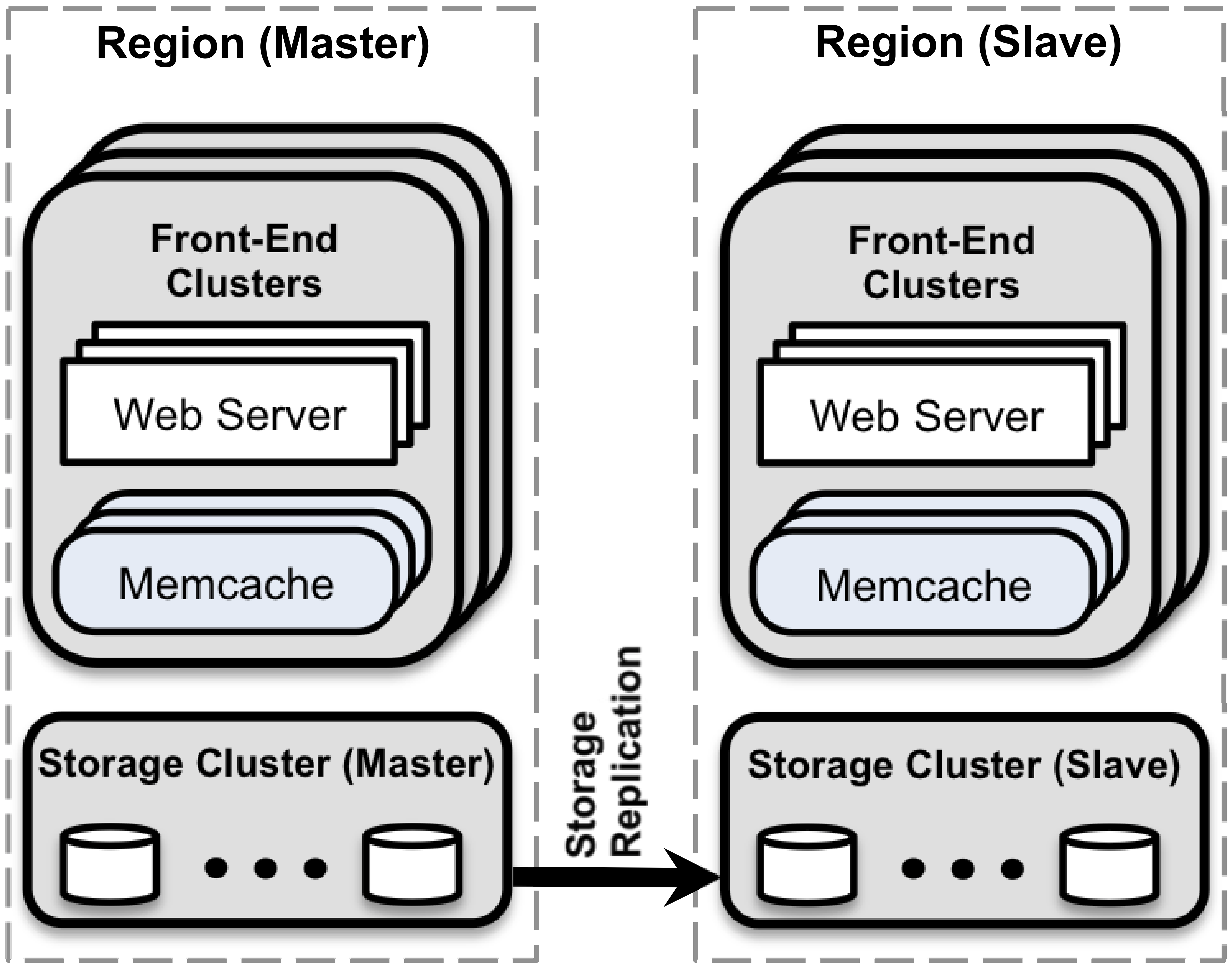
Scaling Memcache at Facebook

Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C. Li,
Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, David Stafford, Tony Tung,
Venkateshwaran Venkataramani
{rajeshn,hans}@fb.com, {sgrimm, marc}@facebook.com, {herman, hcli, rm, mpal, dpeek, ps, dstaff, ttung, veeve}@fb.com
Facebook Inc.

One popular page loaded ⇒ average of **521** distinct `memcache` fetches

95th percentile: **1740** distinct `memcache` fetches

Facebook service architecture



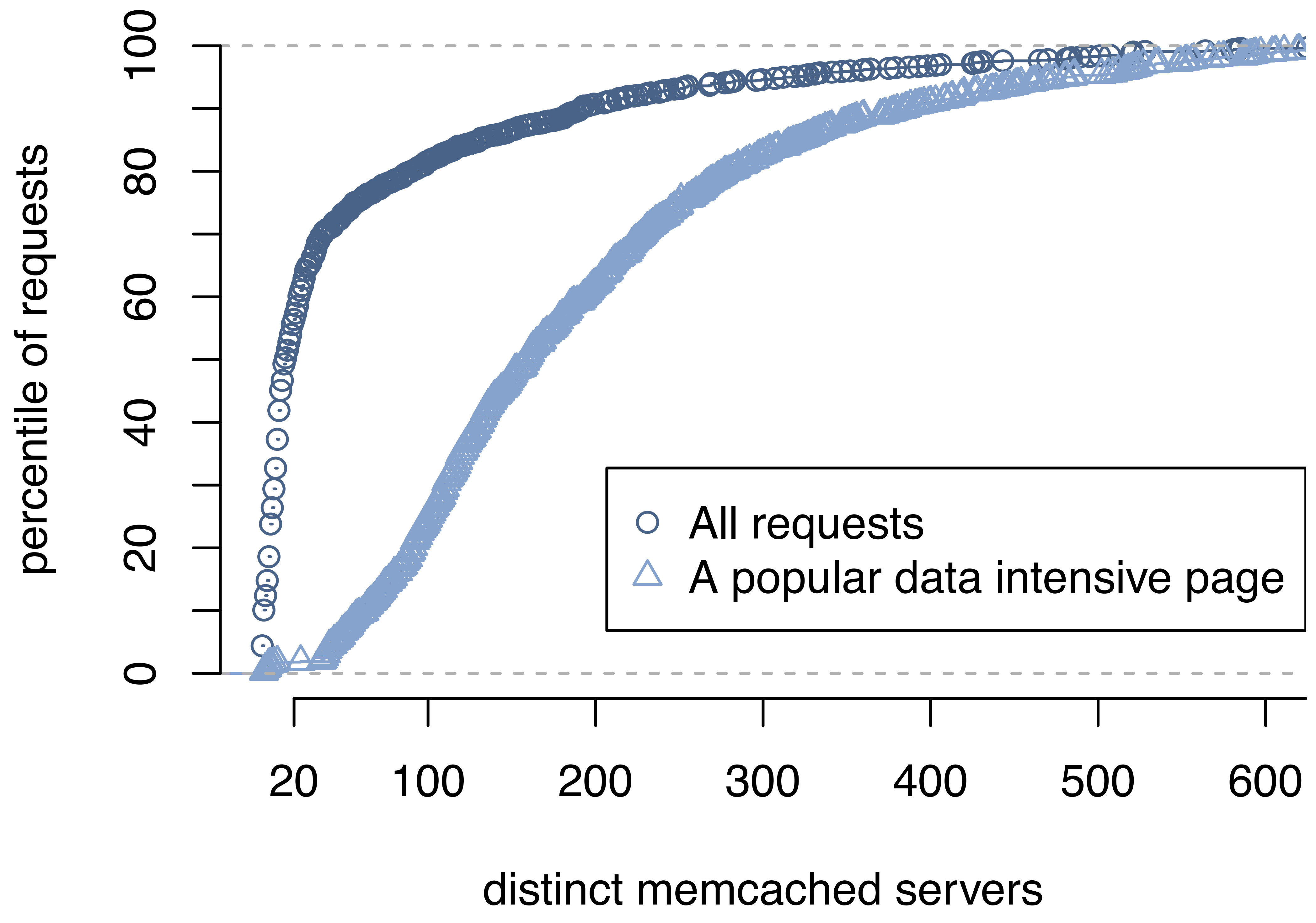
[from Nishtala et al., Scaling Memcache at Facebook, NSDI 2013]

Memcached: service characteristics

O(billions) scale

Wide "fan-out"

- 100s of memcached servers per request
- Causes all-to-all traffic from web to memcached servers



(from web server for single web request)

[from Nishtala et al., Scaling Memcache at Facebook, NSDI 2013]

Memcached: service characteristics

O(billions) scale

App workflows have wide "fan-out"

- 100s of memcached servers per request
- Causes all-to-all traffic from web to memcached servers

App workflows need multiple rounds per request

- Service tasks according to the DAG of dependencies
- Example of needing multiple rounds?

Memcached: service characteristics

O(billions) scale

App workflows have wide "fan-out"

- 100s of memcached servers per request
- Causes all-to-all traffic from web to memcached servers

App workflows need multiple rounds per request

- Service tasks according to the DAG of dependencies
- Example of needing multiple rounds?

Implications

- Need extreme performance
- Exceptional conditions become the common case

Memcached: scaling to billions

A cornucopia of systems optimizations

- Aggregate queries across threads, compression, batching requests in one packet, custom malloc, use UDP, client flow control to avoid incast, …
- One master region handles writes, others read-only

Keep memcache servers simple

- Only talk to web clients
- Web clients handle complexity (e.g., installing cached values, carrying tokens, error recovery)

Pr[stale] is tunable, not a correctness problem

Interesting observations

Warmup takes hours!

- Bring up new cluster fast by moving content from already-warm memcache cluster
- memcached servers store cached values semi-persistently
  - in shared memory region
  - doesn’t die when memcached process is killed or upgraded!

Intriguing questions

- What would happen if you shut off Facebook and turned it back on again?
- What if you shut off the Internet and turned it back on again?

Big data analytics

Hadoop

Spark

Dryad

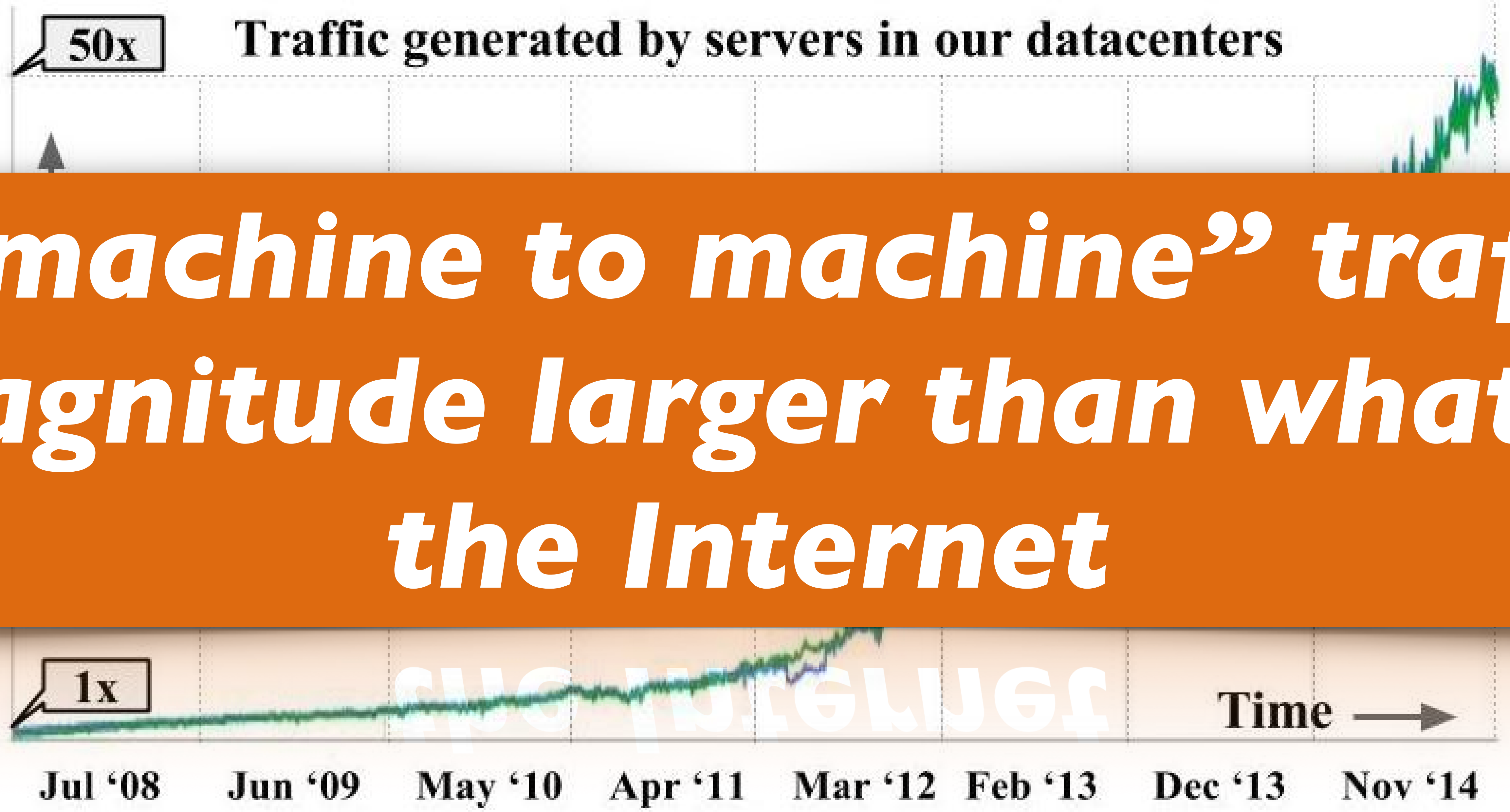
Database *joins*

⋮

What does data center traffic look like?

It depends … on applications, scale, network design, …

Traffic characteristics: growing volume



Facebook: *"machine to machine" traffic is several orders of magnitude larger than what goes out to the Internet*

"Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network", Arjun Singh et al. @ **Google**, ACM SIGCOMM'15

Traffic characteristics: rack locality

Facebook

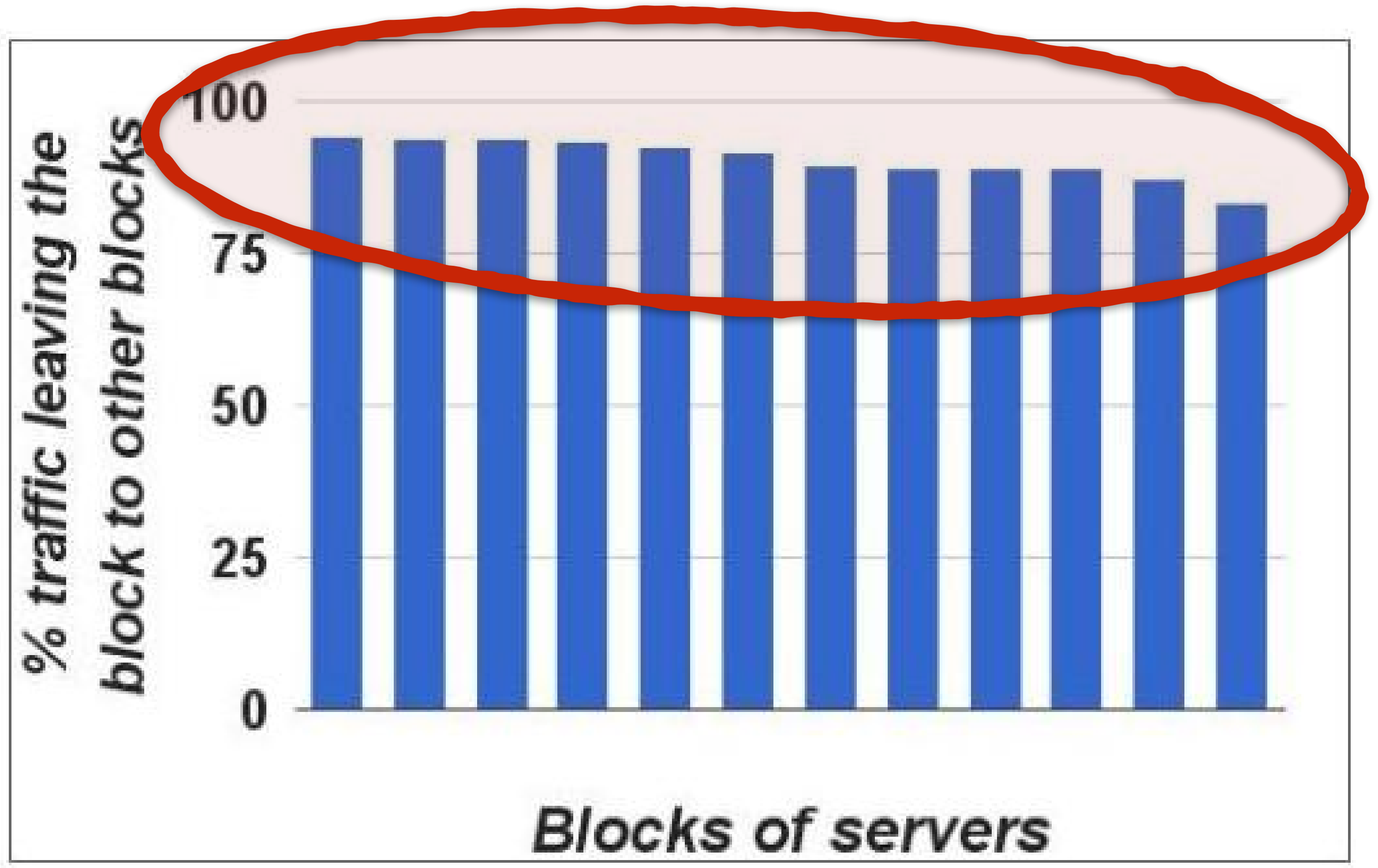
"Inside the Social Network's (Datacenter) Network"
Arjun Roy et al., ACM SIGCOMM'15

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

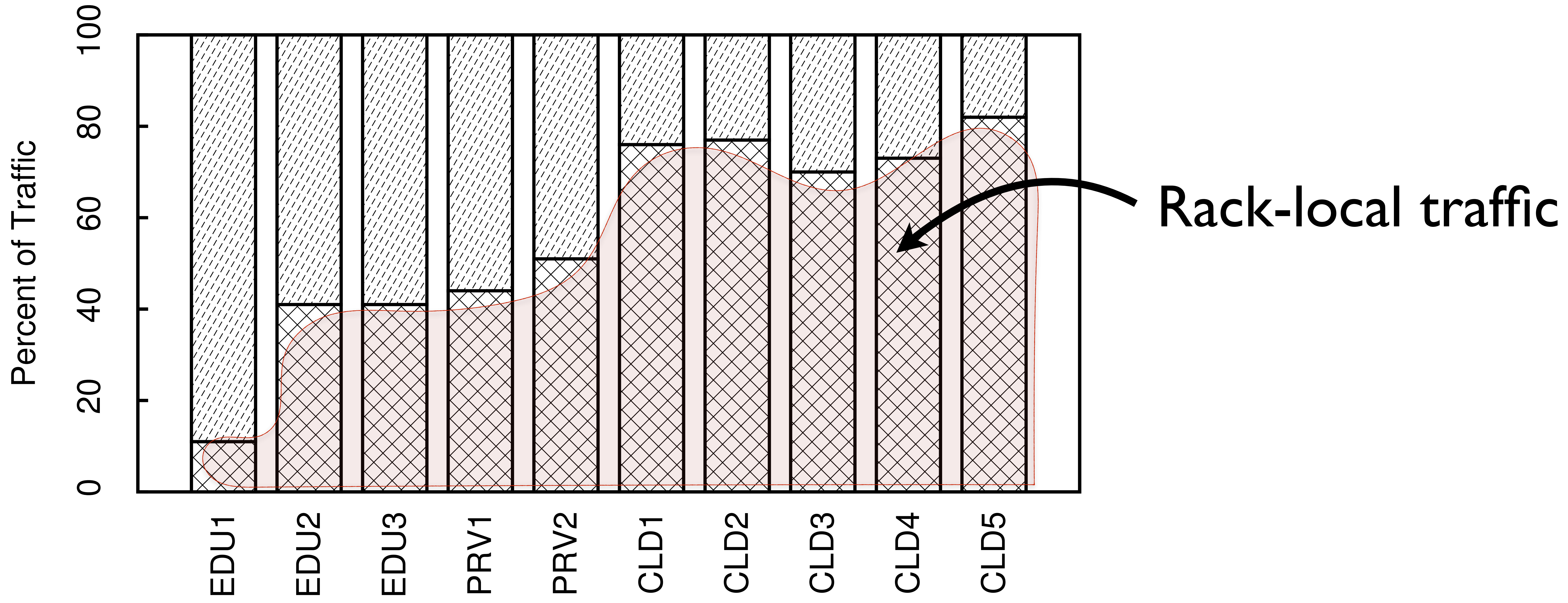
Google

"Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network"
Arjun Singh et al., ACM SIGCOMM'15

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



Traffic characteristics: rack locality



"Network Traffic Characteristics of Data Centers in the Wild"
Theophilus Benson et al., ACM IMC'10

Traffic characteristics: concurrent flows

Facebook

"Inside the Social Network's (Datacenter) Network"
Arjun Roy et al., ACM SIGCOMM'15

*"Web servers and cache hosts have **100s to 1000s** of concurrent connections"*

*"Hadoop nodes have approximately **25** concurrent connections on average."*

1500 server cluster @ ??

"The Nature of Datacenter Traffic: Measurements & Analysis"
Srikanth Kandula et al. (Microsoft Research), ACM IMC'09

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

Traffic characteristics: flow arrival rate

Facebook

"Inside the Social Network's (Datacenter) Network"
Arjun Roy et al., ACM SIGCOMM'15

"median inter-arrival times of approximately 2ms" at a server

1500 server cluster @ ??

"The Nature of Datacenter Traffic: Measurements & Analysis"
Srikanth Kandula et al. (Microsoft Research), ACM IMC'09

< 0.1x Facebook's rate

Traffic characteristics: flow sizes

Facebook

"Inside the Social Network's (Datacenter) Network"
Arjun Roy et al., ACM SIGCOMM'15

Hadoop: median flow <1KB
<5% exceed 1MB or 100sec

Caching: most flows are long-lived
… but bursty internally

Heavy-hitters ≈ median flow, not persistent

1500 server cluster @ ??

"The Nature of Datacenter Traffic: Measurements & Analysis"
Srikanth Kandula et al. (Microsoft Research), ACM IMC'09

> 80% of the flows last <10sec

> 50% bytes are in flows lasting less <25sec

What does data center traffic look like?

It depends … on applications, scale, network design, …

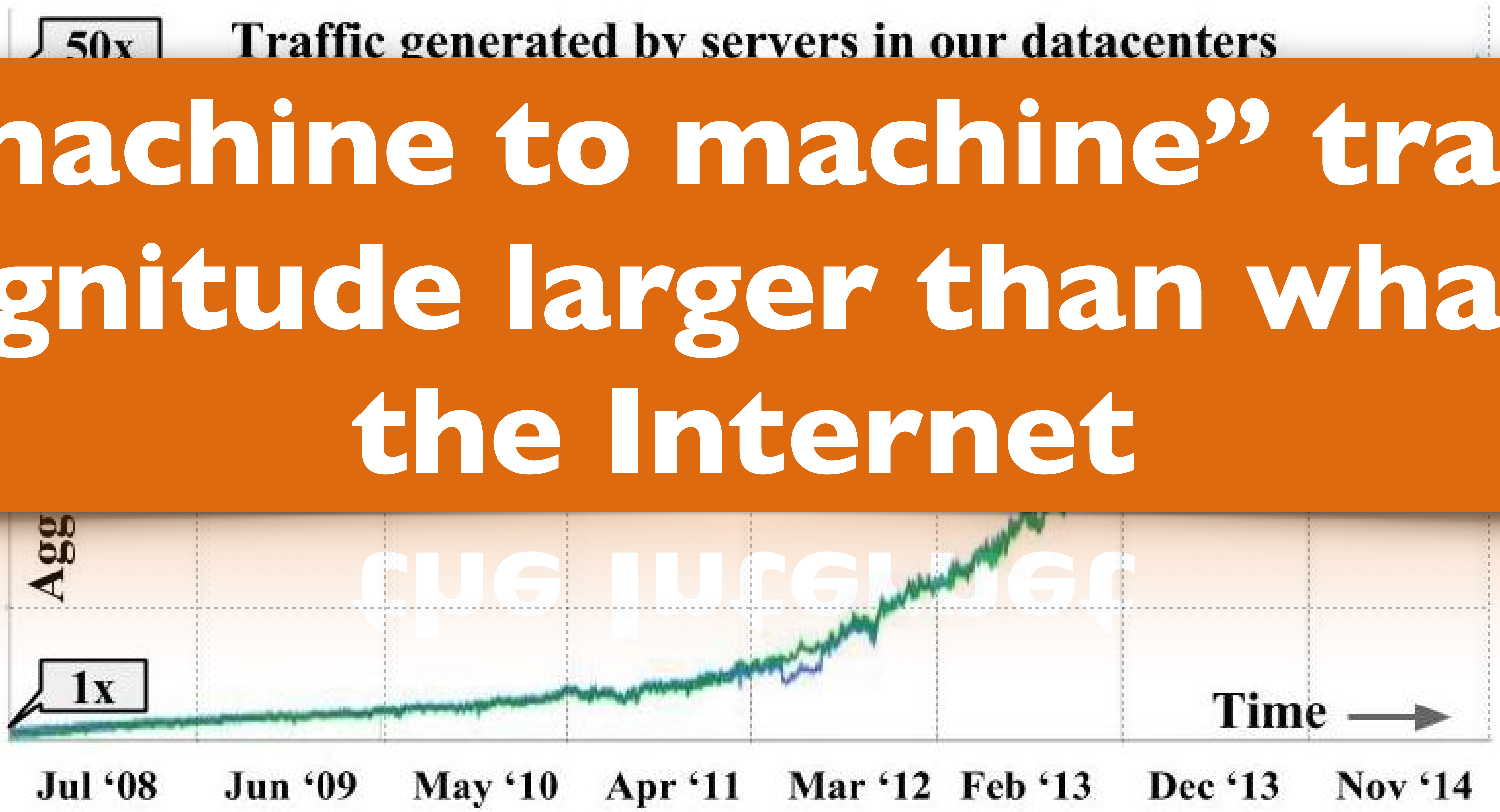
… and right now, not a whole lot of data is available.

Implications for networking

1. Data center internal traffic is BIG
2. Tight deadlines for network I/O
3. Congestion and TCP incast
4. Need for isolation across applications
5. Centralized control at the flow level may be difficult

Implications for networking

- Data center internal traffic is BIG

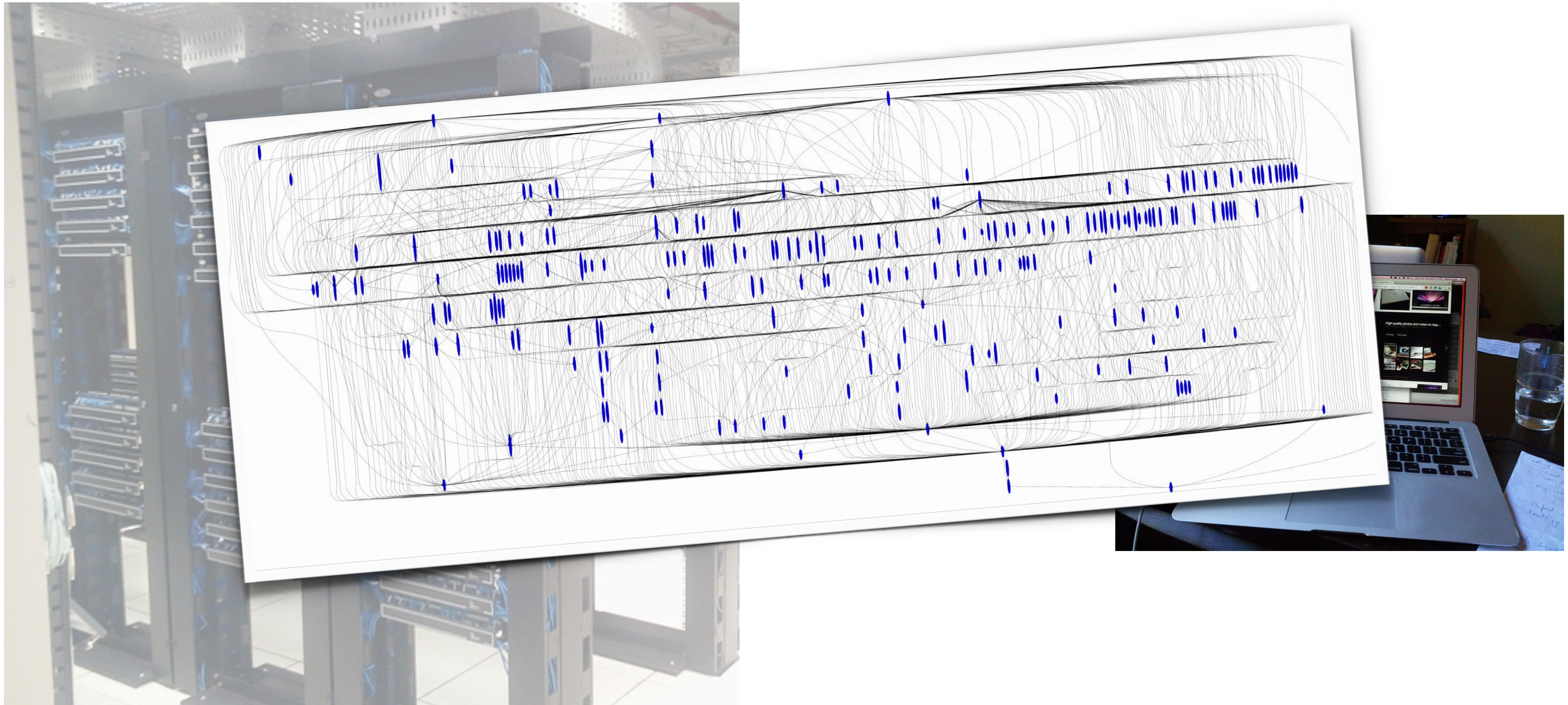


Facebook: "machine to machine" traffic is several orders of magnitude larger than what goes out to the Internet

"Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network", Arjun Singh et al. @ **Google**, ACM SIGCOMM'15

Implications for networking

2 Tight deadlines for network I/O



Implications for networking

2 Tight deadlines for network I/O

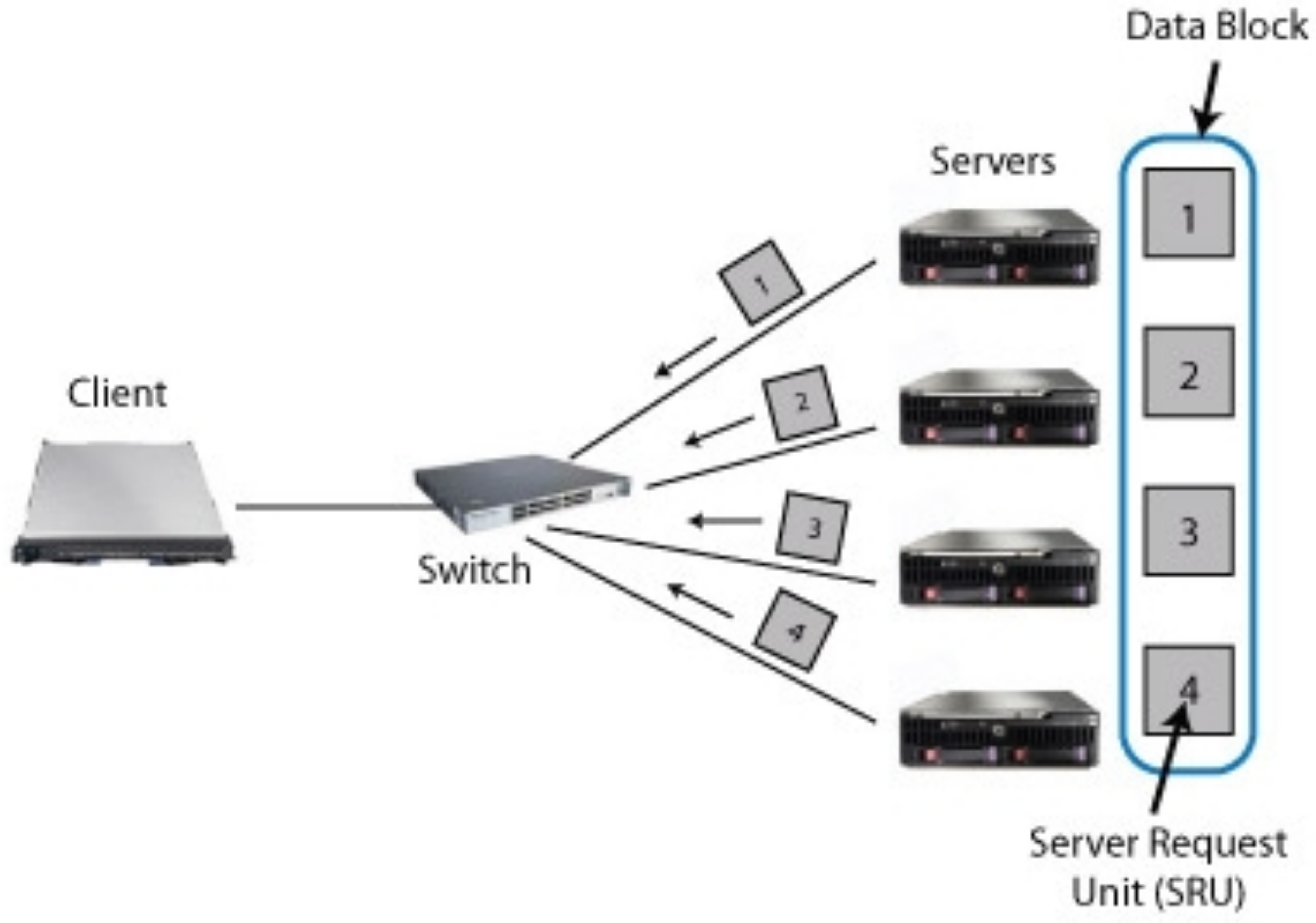
Suppose: server response-time is 10ms for 99% of requests; 1s for 1%

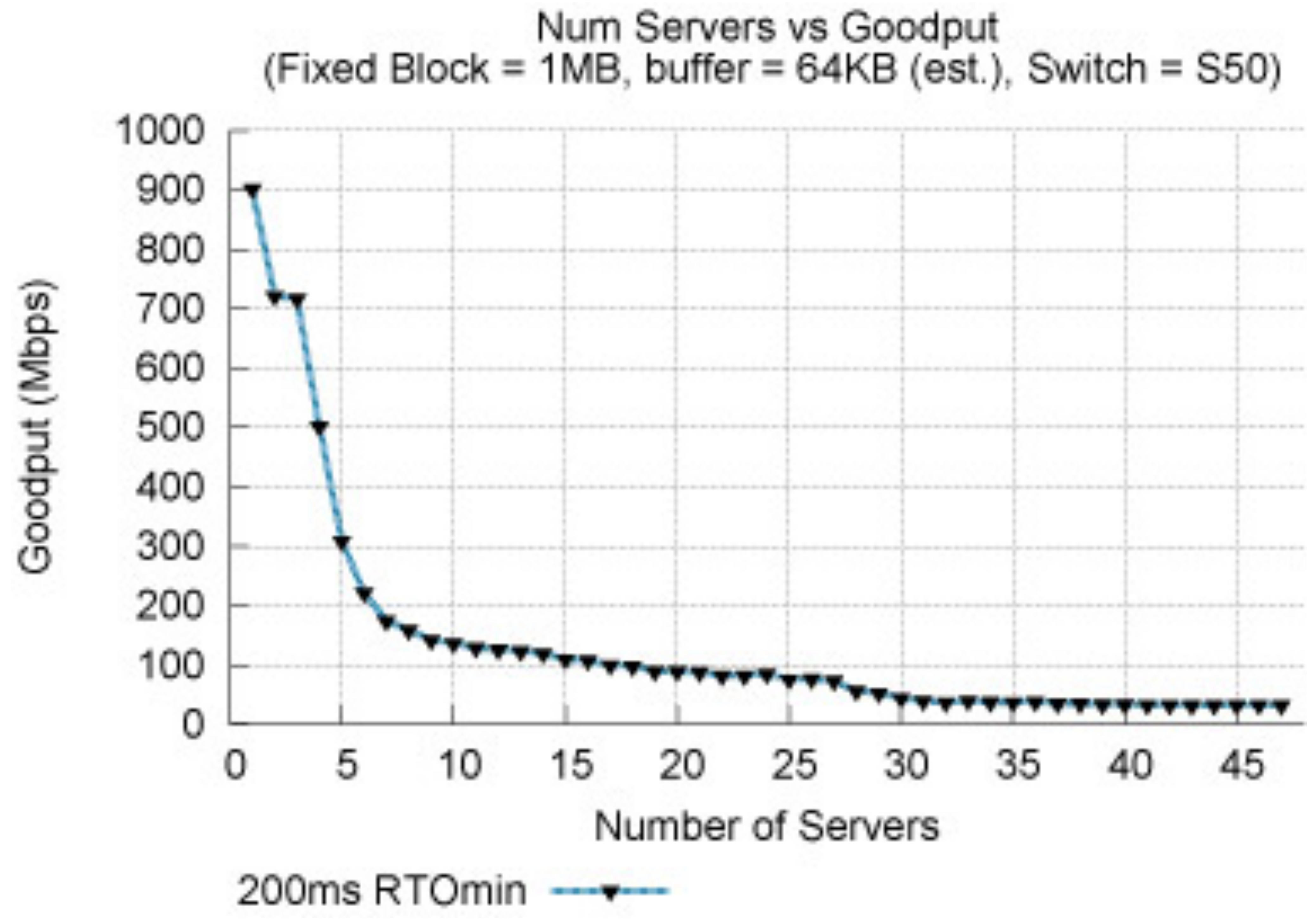
#Servers	Requests 1s or slower
1	1%
100	63%

Need to reduce variability and tolerate *some* variation

Implications for networking

3 Congestion and TCP incast

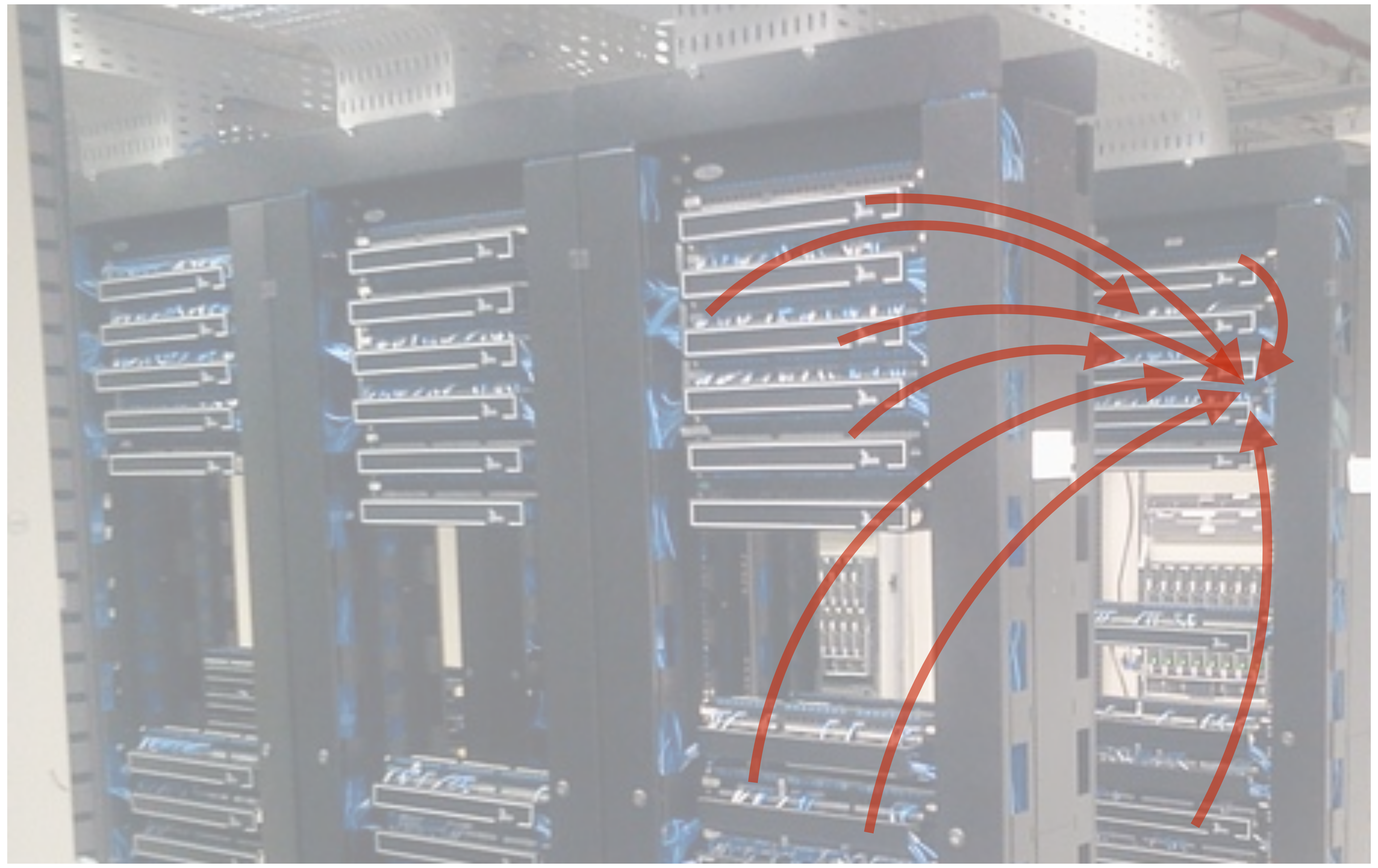


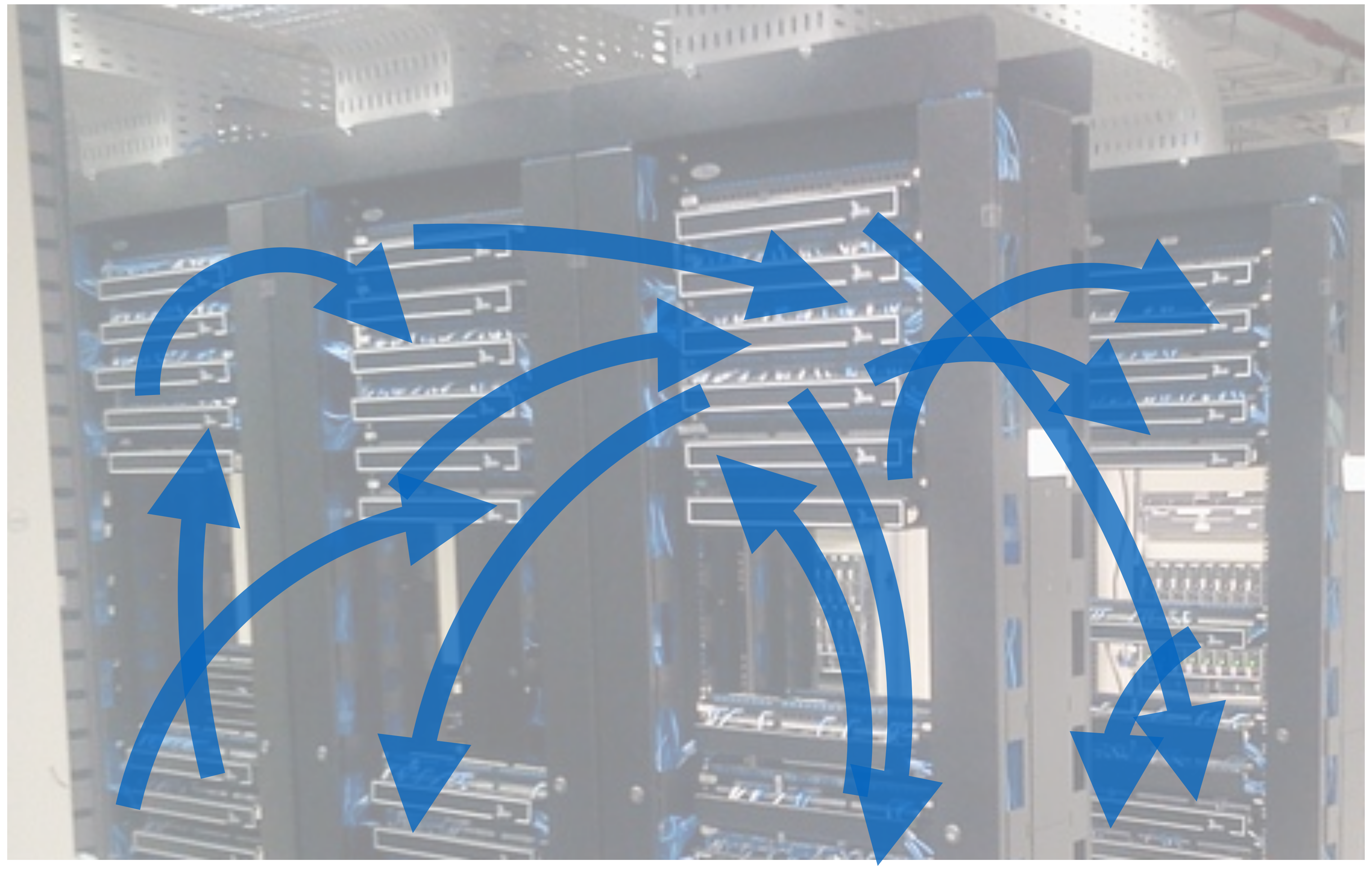


Figures from CMU PDL INCAST project:
http://www.pdl.cmu.edu/Incast/

Implications for networking

4 Need for isolation across applications

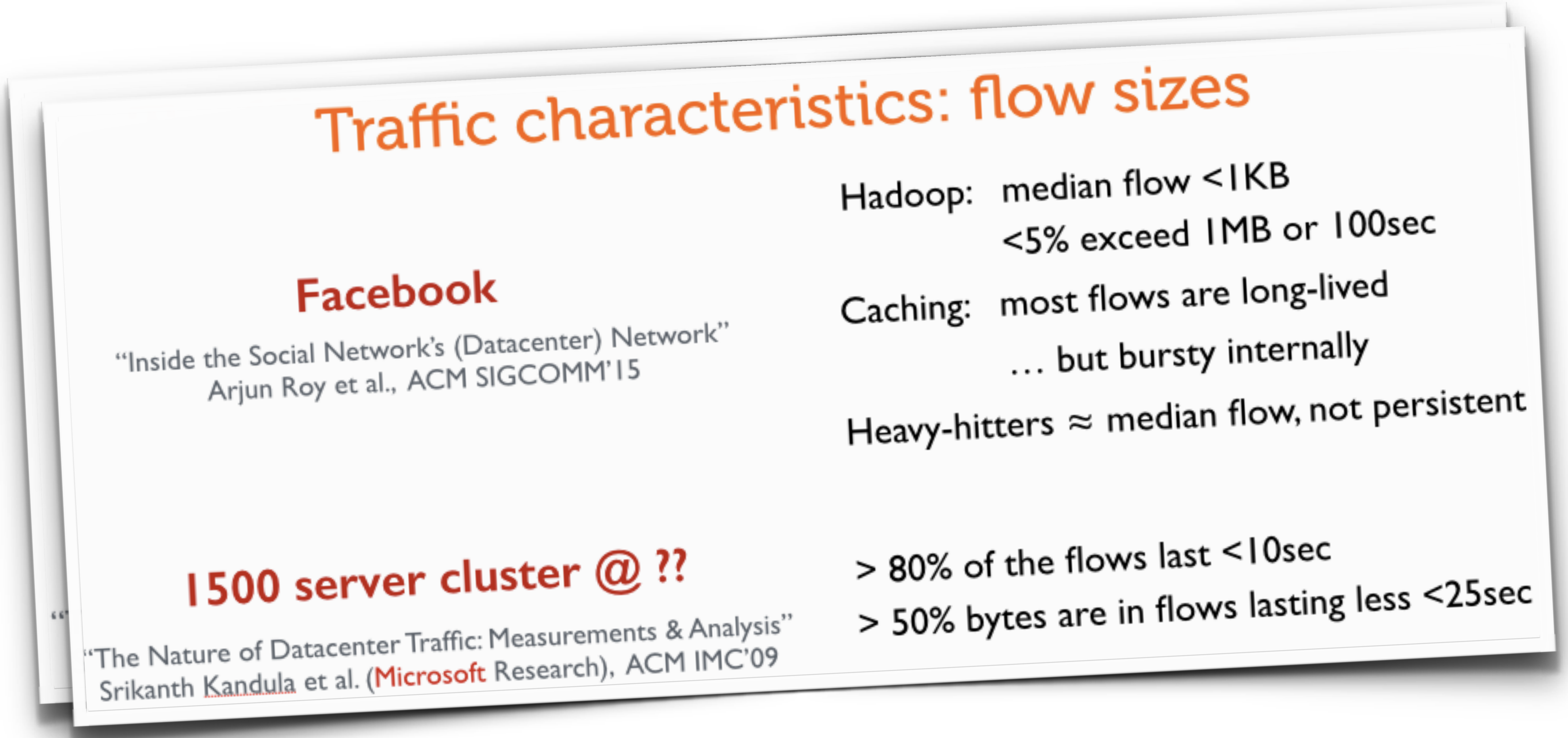




Applications with different objectives sharing the network

Implications for networking

5 Centralized control at the flow level may be difficult



Distributed control, perhaps with some centralized tinkering

NDP

Re-architecting datacenter networks and stacks for low latency and high performance

Handley, Raiciu, Agache, Voinescu, Moore, Antichi, Wójcik

SIGCOMM 2017

Key ideas:

- Get the most out of our nonblocking network
  - Send at line rate! Not even an RTT for connection setup!
  - Spread packets across all paths, round-robin
- Recover from loss quickly
  - Packet trimming and prioritization of control packets
  - Result: super fast notification of loss
- Avoid loss as quickly as possible
  - Receiver-driven pacing

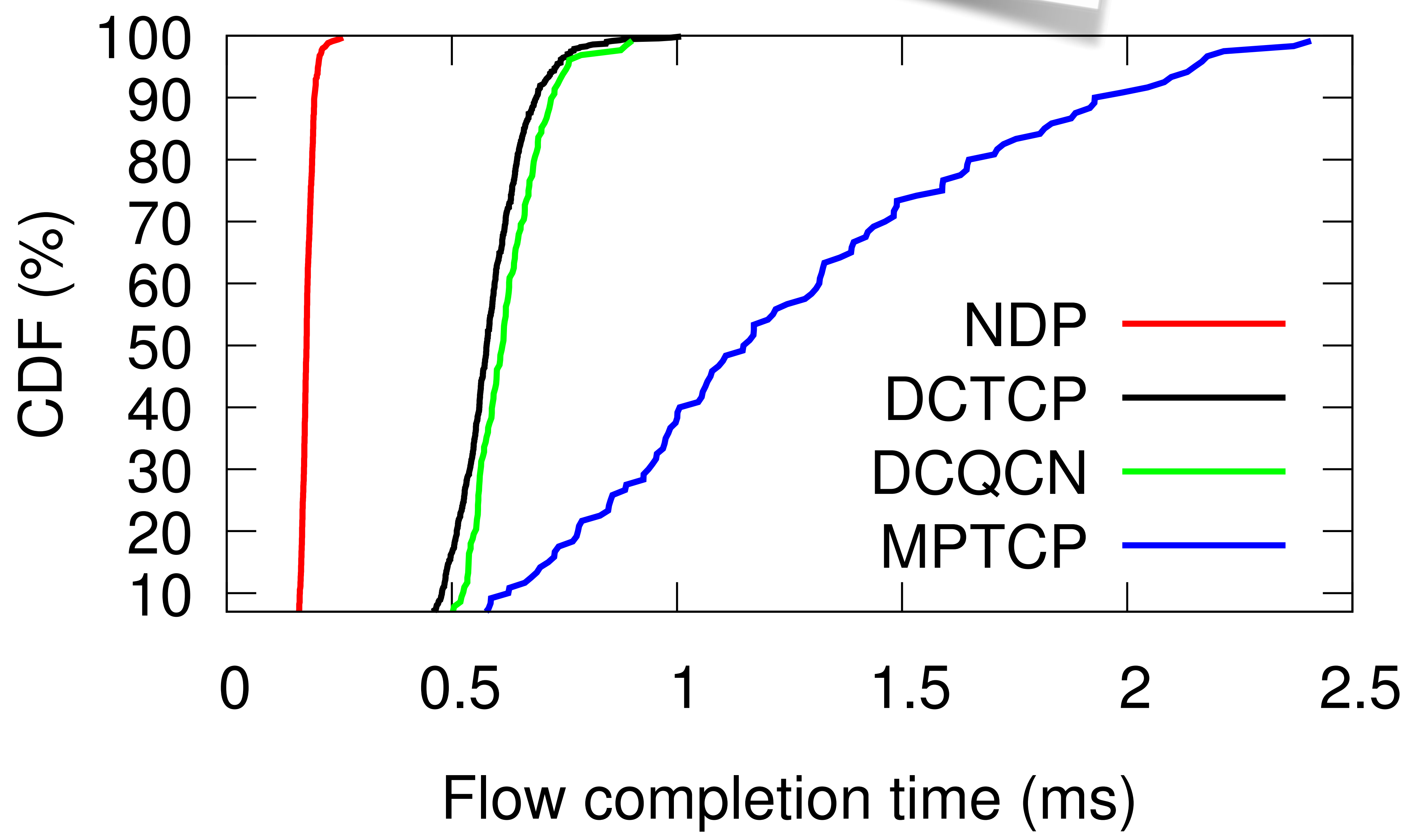


Figure 15: FCT for 90KB flows with random background load, 432 node FatTree.

NDP Discussion

How do we deal with out-of-order arrival before the connection has even started?

Could end-host buffers overflow with control packets?

Why is source routing better than per-packet random forwarding?

Zero-RTT setup and the security problem

Next time

Software-defined data centers: handling multi-tenancy

Assignment 2 released