More is Less: Reducing Latency via Redundancy

Ashish Vulimiri (UIUC)

Oliver Michel (U. Vienna)

Brighten Godfrey (UIUC)

Scott Shenker (ICSI/UC Berkeley)

Latency

Latency

- Online services

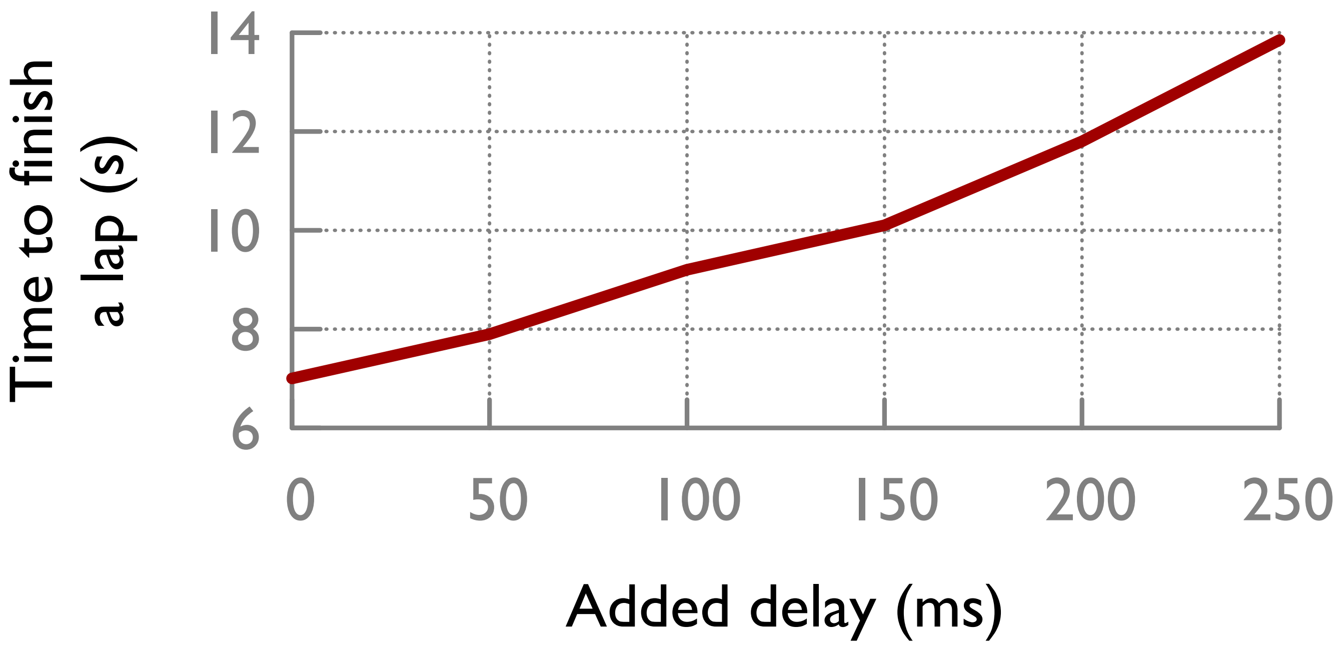
	Delay	Result
Amazon	+100ms	-1% revenue
Bing	+500ms	-1.2% revenue
Google	+400ms	-0.6% searches

- HCI studies

Latency

- Online services

User performance in a racing game [*]



- HCI studies

[*] L. Pantel, L.C. Wolf, "On the impact of delay on real-time multiplayer games", NOSSDAV '02

Latency

Controlling latency is difficult:

Latency

Controlling latency is difficult:

1. Pervasive uncertainty

Latency

Controlling latency is difficult:

1. Pervasive uncertainty
   - Link congestion
   - Cache miss
   - Slow disk lookup
   - Delay due to virtualization
   - ...

Latency

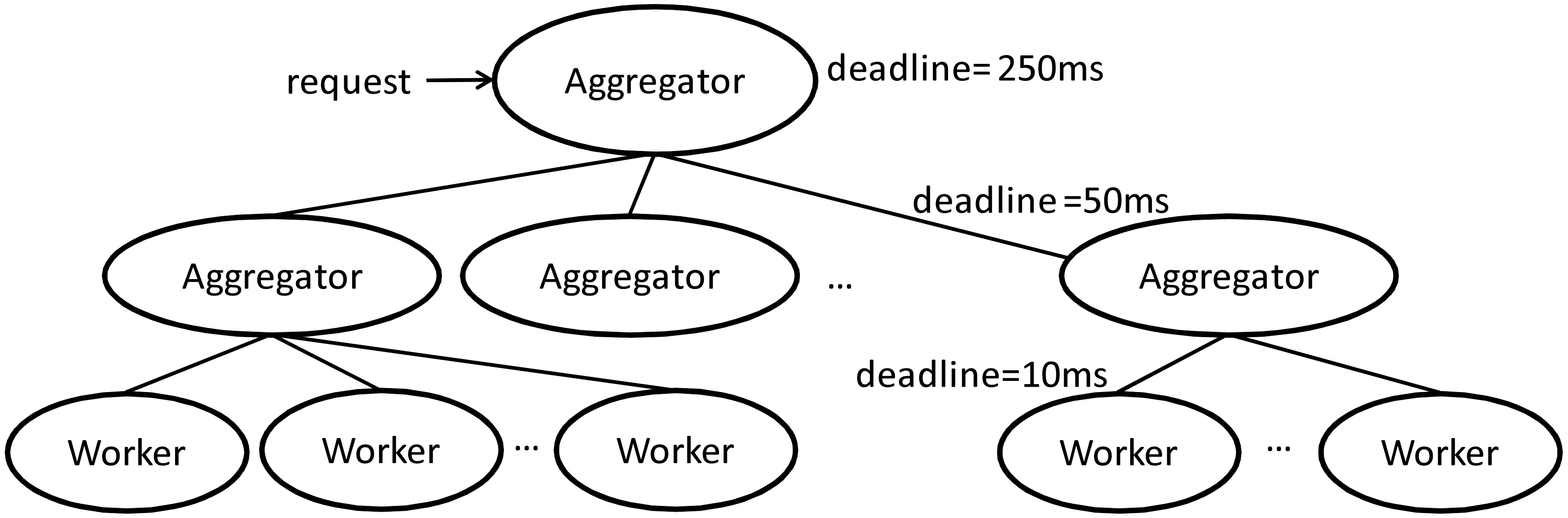
Controlling latency is difficult:

2. Application structure

Latency

Controlling latency is difficult:

2. Application structure



Partition/aggregate pattern

Alizadeh et al., “Data center TCP”, SIGCOMM’10

Latency

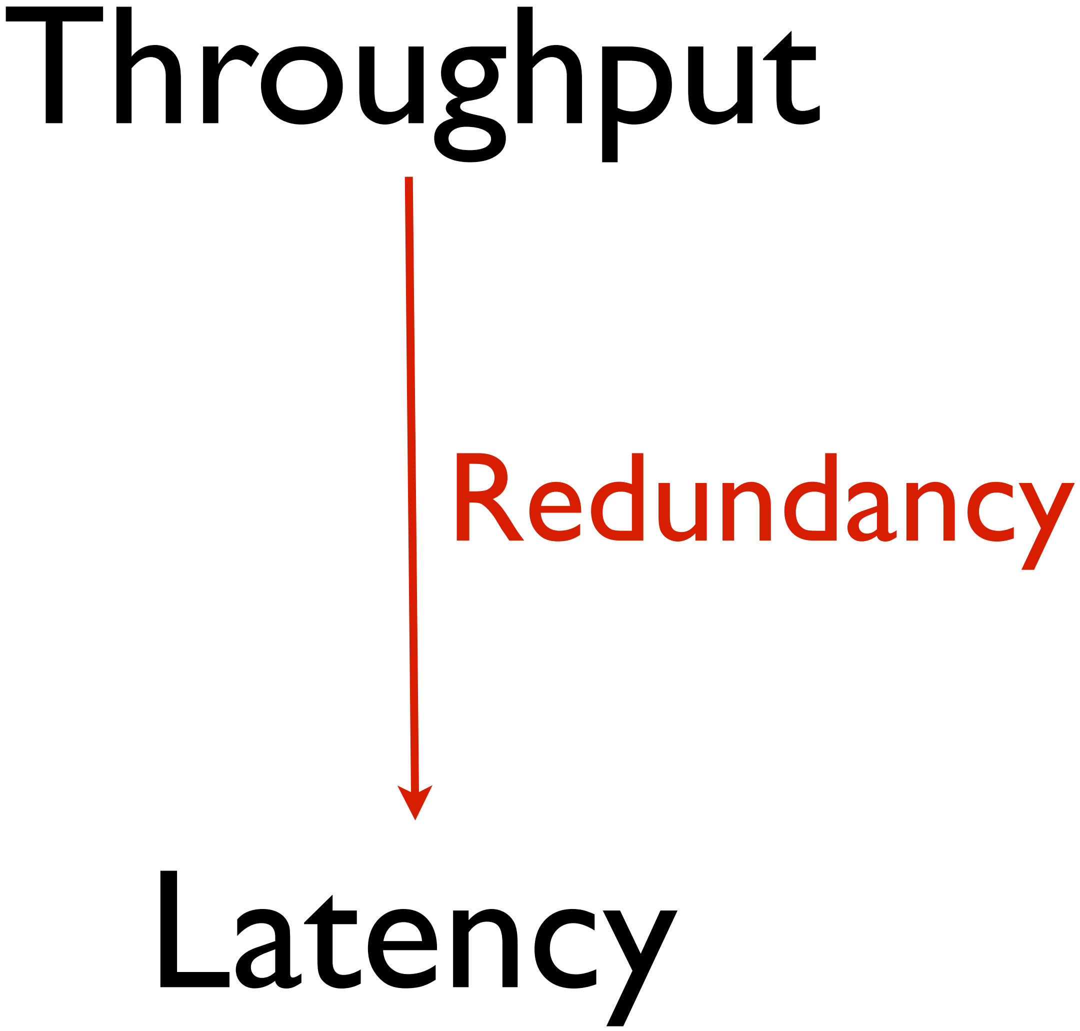
Controlling latency is difficult:

1. Pervasive uncertainty

2. Application structure

Throughput

Latency



Throughput

Redundancy

Latency

Redundancy

Redundancy

- Some past uses:
  - Distributed jobs (speculative execution)[1]
  - DTNs [2]
  - DHT queries [3]

[1] Ananthanarayanan et al.,"Why let resources idle? Aggressive cloning of jobs using Dolly", HotCloud '12

[2] Soljanin,"Reducing delay with coding in multi-agent information transfer", Allerton '10

[3] Li et al.,"Bandwidth efficient management of DHT routing tables", NSDI '10

Argument

1. Overhead should be tolerable

2. When is cost < benefit?

3. Example applications

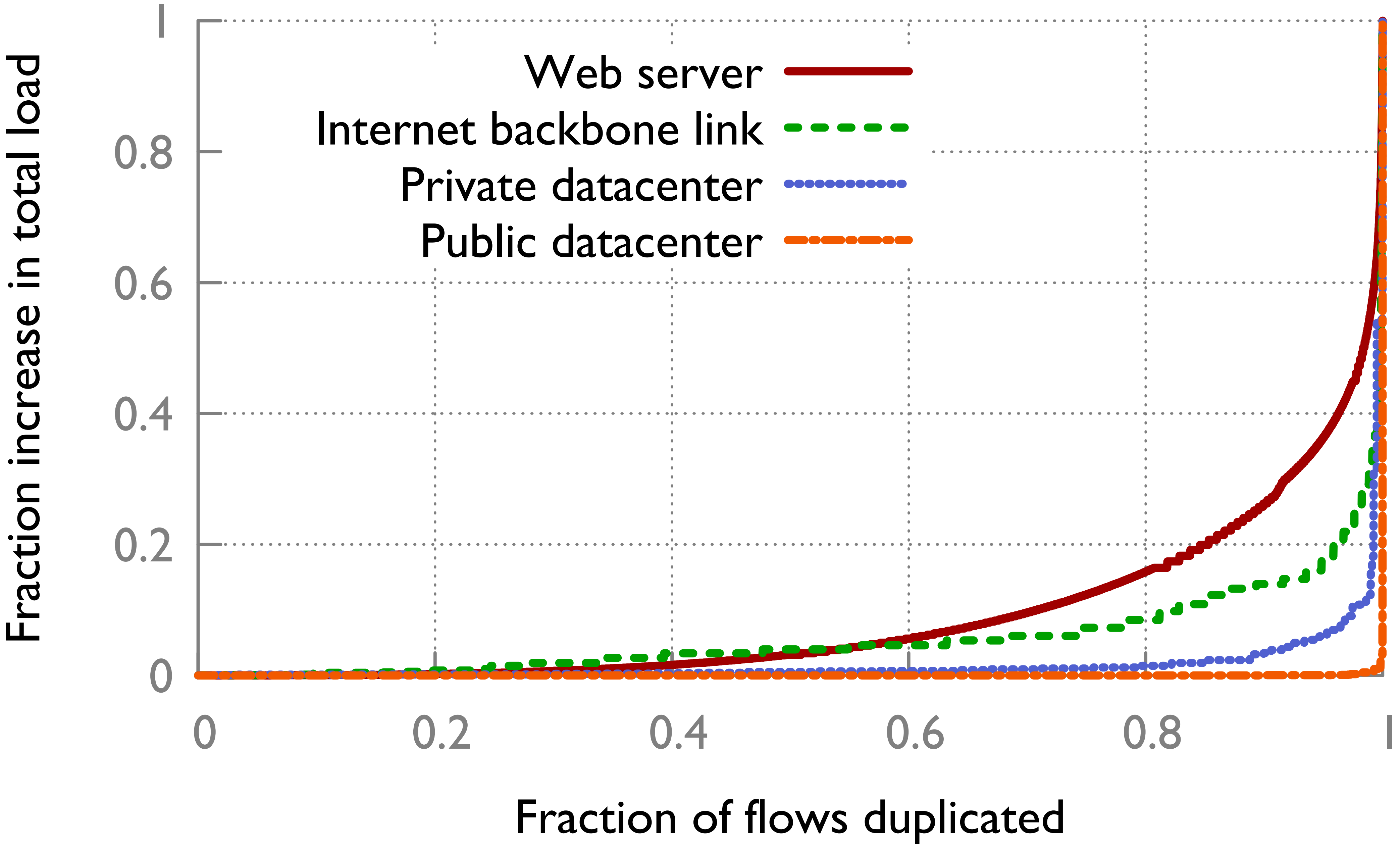
Overhead

Intuitively, overhead should be low because

1. Latency-sensitive tasks likely to be small

2. Heavy tails are pervasive

What is the overhead from replicating the x% smallest flows?



Cost vs benefit

Redundancy is only useful if

cost < benefit

Cost vs benefit

Redundancy is only useful if

cost	<	benefit		
cost of overhead ($/KB)	<	latency savings (ms/KB)	X	value of time ($/ms)

cost of overhead < latency savings X value of time

cost of overhead < latency savings X **value of time**

- Hard to estimate

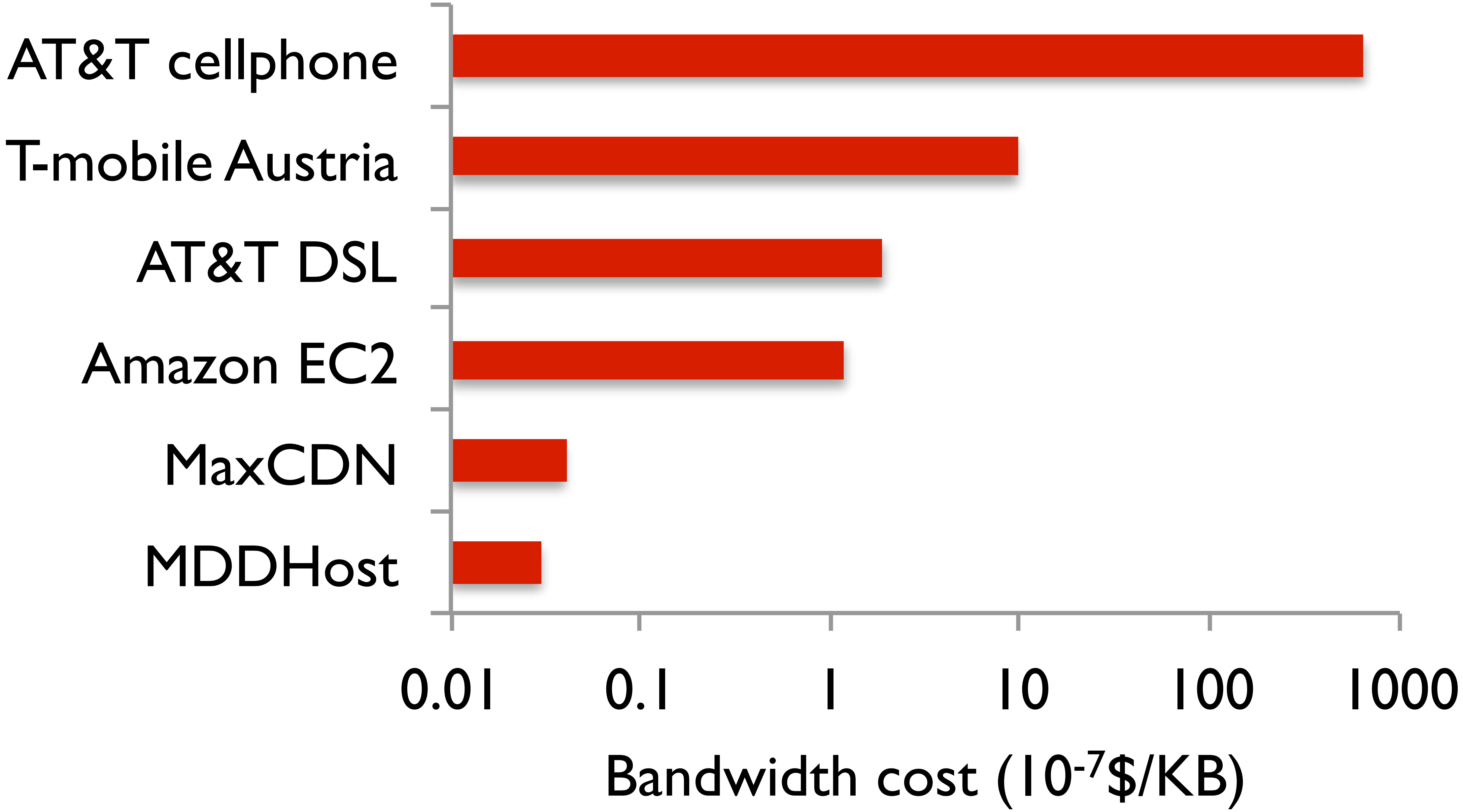
cost of overhead < latency savings X value of time

- Hard to estimate

- As first approximation, we will use

US median wage = 23.5 $/hr

cost of overhead < latency savings X value of time



Cost vs benefit

- Redundancy is useful even with the most expensive cell phone plan if

cost of overhead ($/KB) < latency savings (ms/KB) X value of time ($/ms)

Cost vs benefit

- Redundancy is useful even with the most expensive cell phone plan if

$$\begin{array}{ccccc} 6.5 \text{ x } 10^{-5} & & \text{latency} & & 6.5 \text{ x } 10^{-6} \\ \text{\$/KB} & < & \text{savings} & \text{X} & \text{\$/ms} \end{array}$$

Cost vs benefit

- Redundancy is useful even with the most expensive cell phone plan if

10 ms/KB < latency savings

Cost vs benefit

- Redundancy is useful even with the most expensive cell phone plan if

10 ms/KB < latency savings

- Redundancy is useful with a DSL plan if

0.03 ms/KB < latency savings

Specific applications

1. DNS
2. Multipath overlay
3. Memcached

Targets: 10 ms/KB (cell phone)
0.03 ms/KB (DSL)

DNS

- Replicate DNS queries to multiple servers in parallel
- Evaluation: PlanetLab experiments

DNS: Experiment

DNS: Experiment

Server
Local DNS
Level3
Google
OpenDNS

DNS: Experiment

Server	Avg Response Time (s)
Local DNS	
Level3	
Google	
OpenDNS	

DNS: Experiment

Stage 1: Measure, Rank

Server	Avg Response Time (s)
Local DNS	0.27
Level3	0.61
Google	0.16
OpenDNS	0.37

DNS: Experiment

Stage 1: Measure, Rank

Server	**Avg Response Time (s)**
Google	0.16
Local DNS	0.27
OpenDNS	0.37
Level3	0.61

DNS: Experiment

Stage 1: Measure, Rank

Server	Avg Response Time (s)
Google	0.16
Local DNS	0.27
OpenDNS	0.37
Level3	0.61

DNS: Experiment

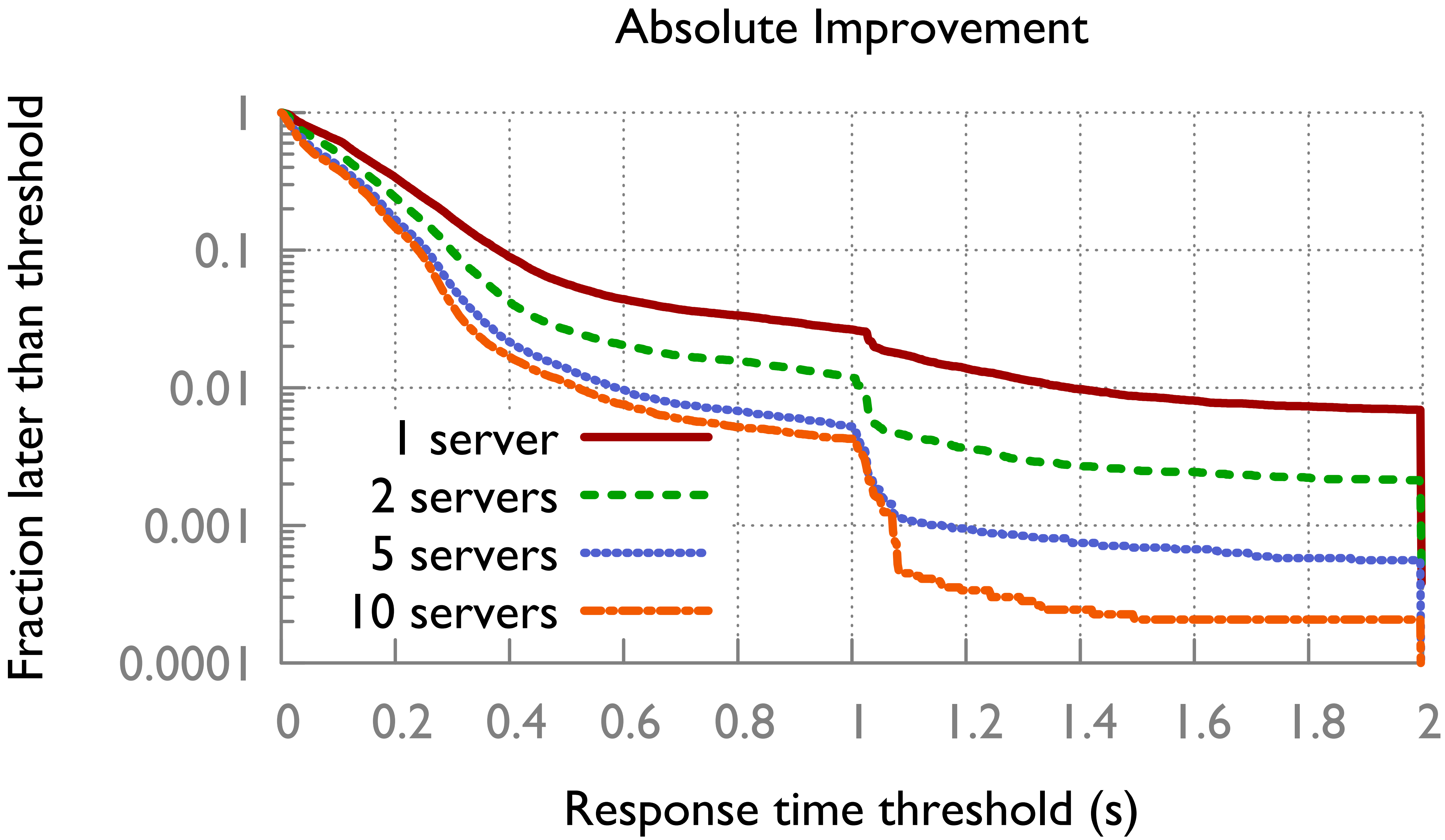
Stage 1: Measure, Rank

Server	Avg Response Time (s)
Google	0.16
Local DNS	0.27
OpenDNS	0.37
Level3	0.61

Stage 2: Evaluate

Try different levels of replication, using servers in the ranked order

DNS


Absolute Improvement
Fraction later than threshold
1
0.1
0.01
0.001
0.0001
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Response time threshold (s)
1 server
2 servers
5 servers
10 servers

DNS

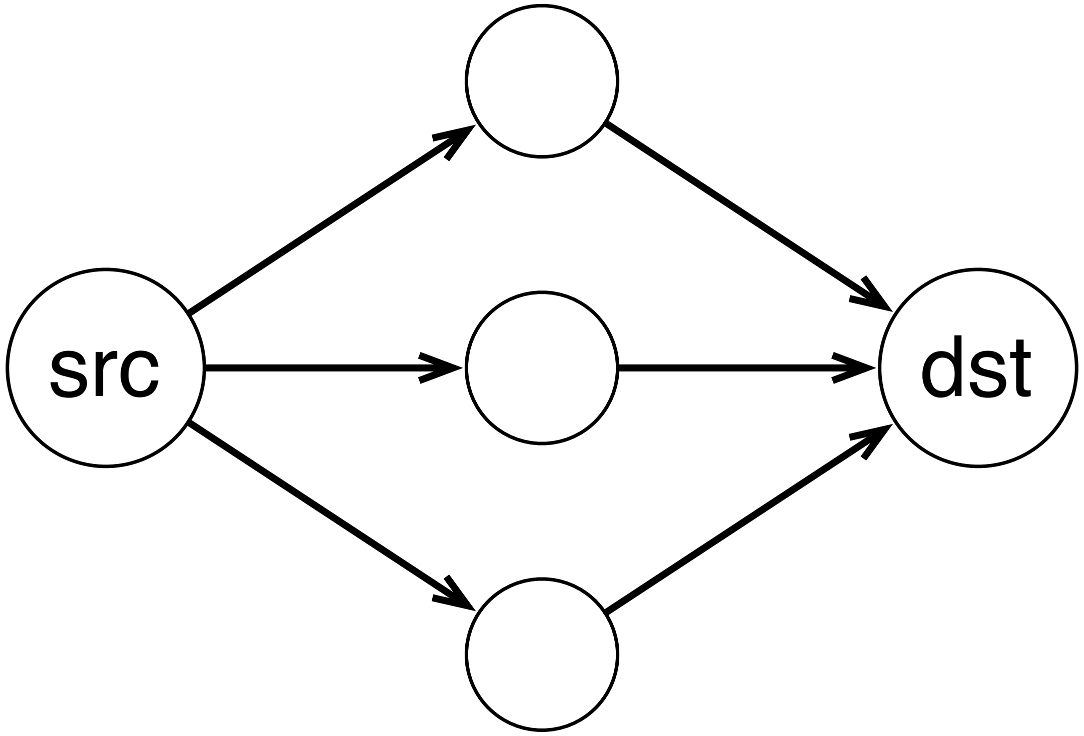
Client location	Optimal number of servers per query	Average latency improvement
Cell phone	5	90ms
DSL	10	100ms

Multipath Overlay

- Send copies of packets on different overlay paths

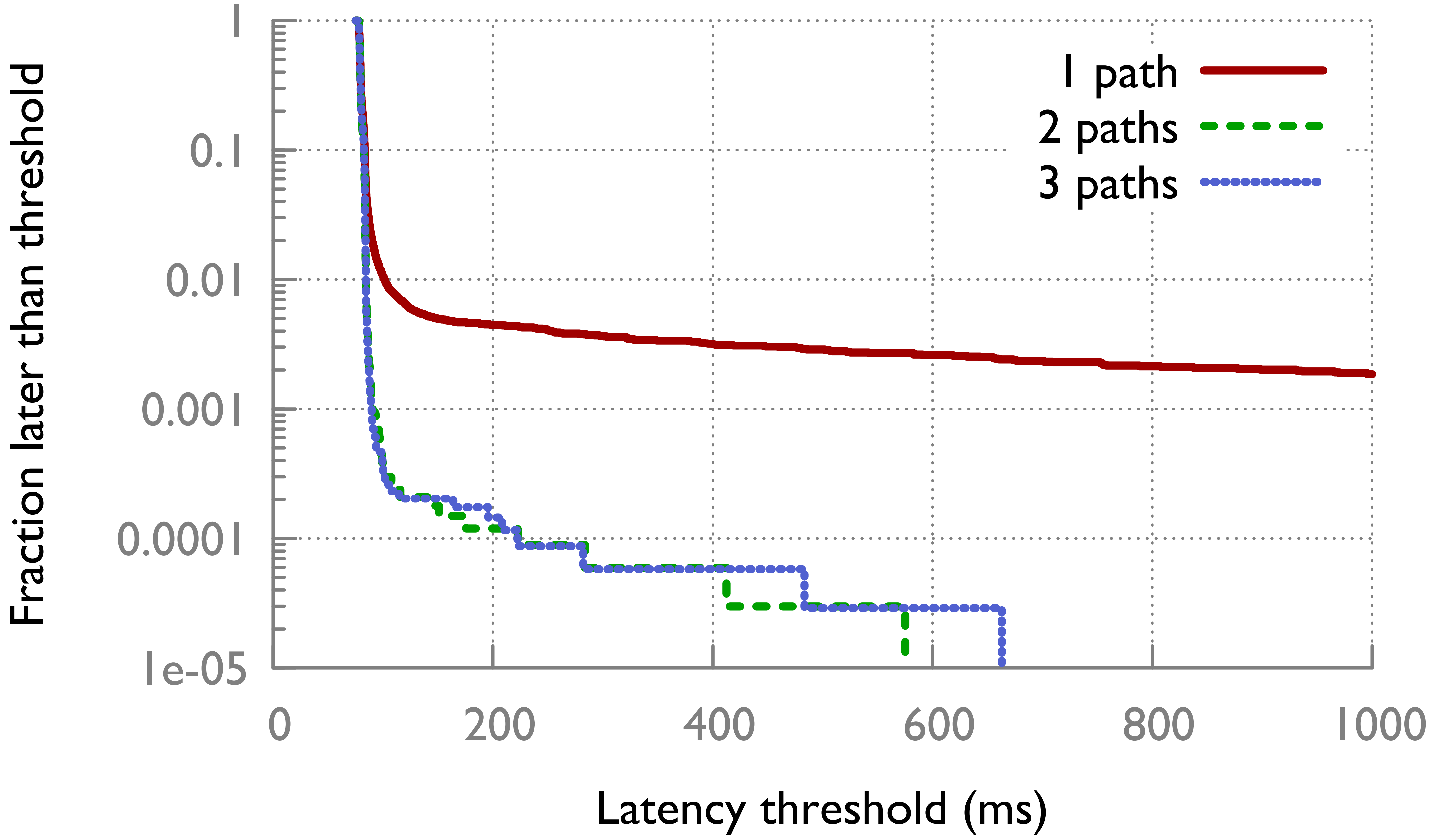
- Evaluation: PlanetLab experiments
  - Note: limited set of topologies

Multipath Overlay



- Data rate: 32kbps-56kbps
- Topology, data rate both match Skype

Multipath Overlay



Multipath Overlay

	2 paths	3 paths
Mean latency savings (ms/KB)	0.8	0.4
99.9th %ile latency savings (ms/KB)	260	130

Multipath Overlay

	2 paths	3 paths
Mean latency savings (ms/KB)	0.8	0.4
99.9th %ile latency savings (ms/KB)	260	130

Targets: 10 ms/KB (cell phone)
0.03 ms/KB (DSL)

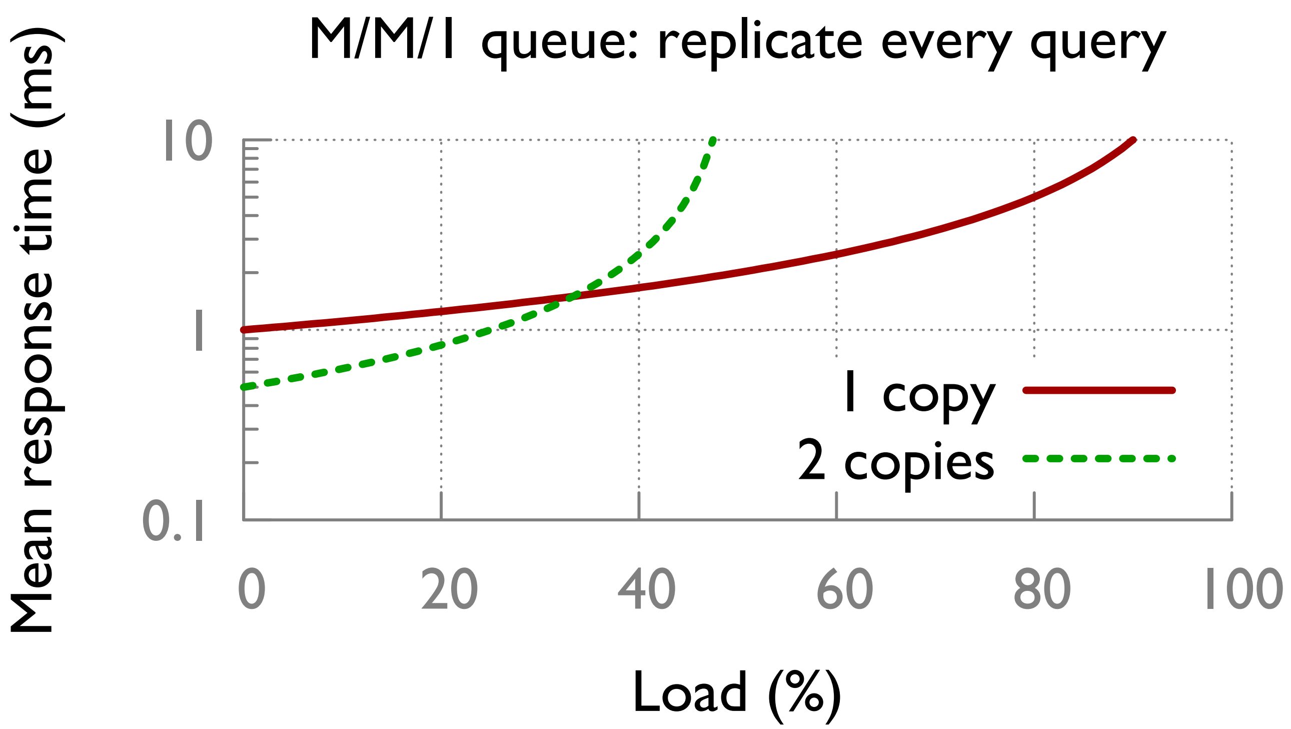
Beyond selfishness

- So far: when should an individual user selfishly replicate?

- Now: look at whole system

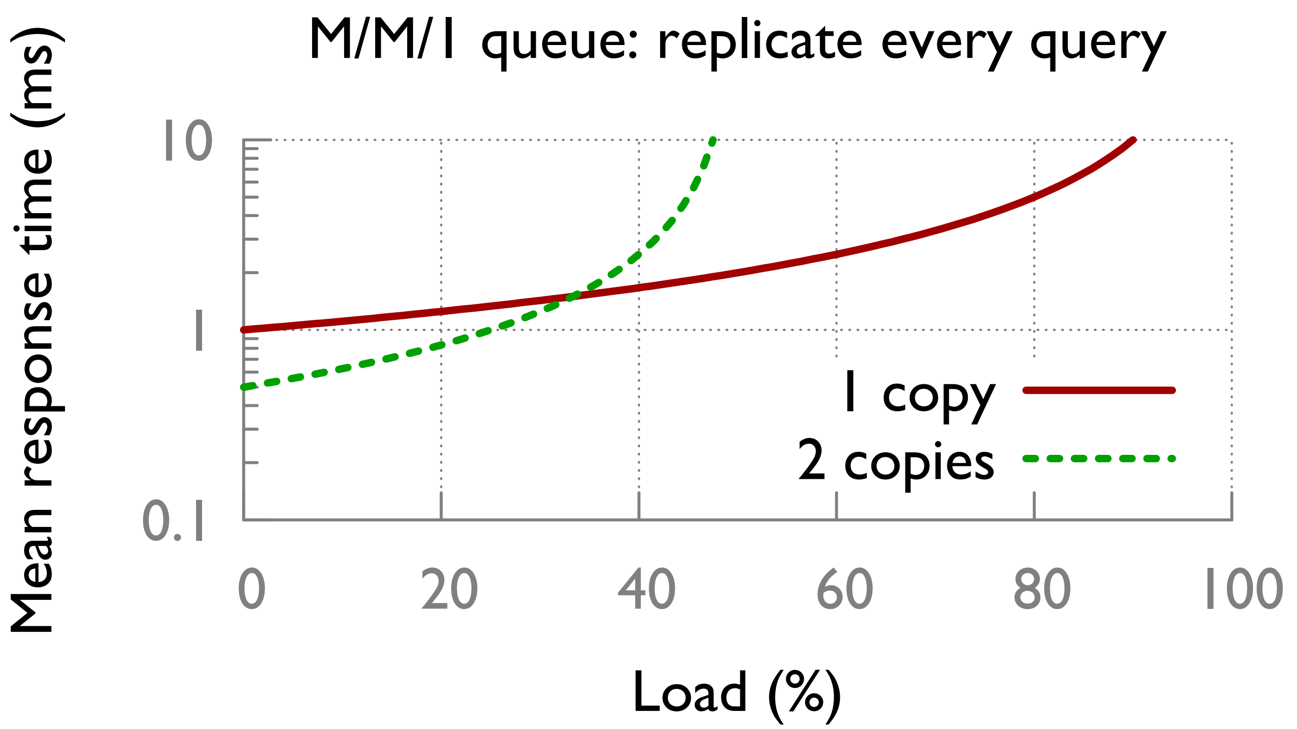
Beyond selfishness

Queueing analysis: threshold effect



Beyond selfishness

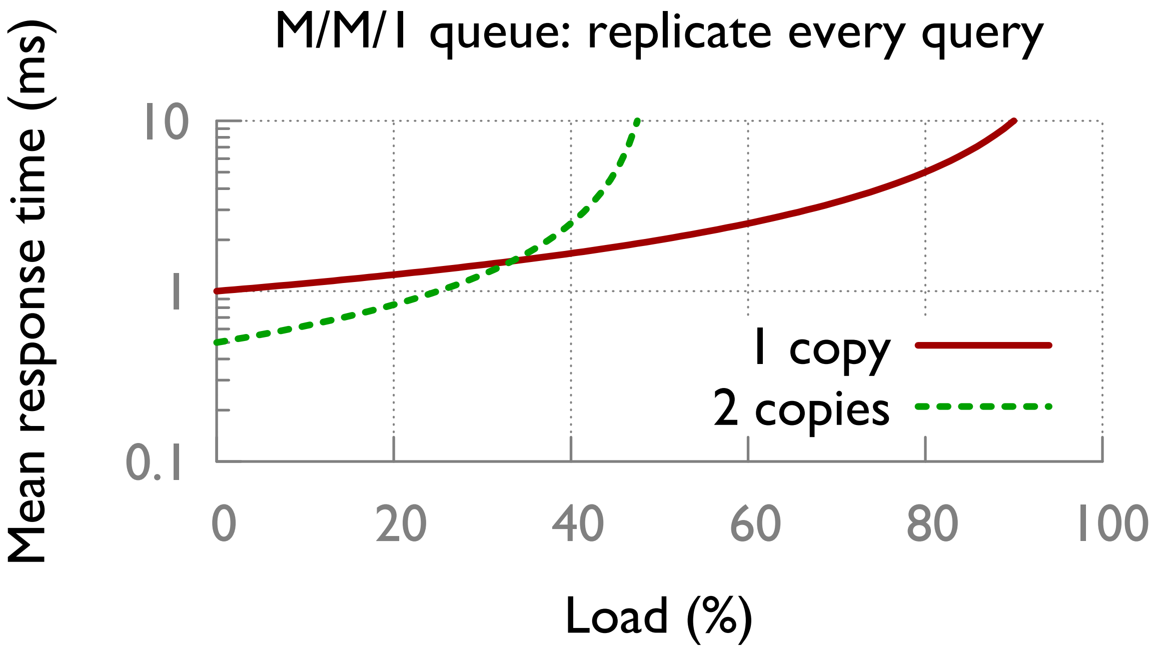
Queueing analysis: threshold effect



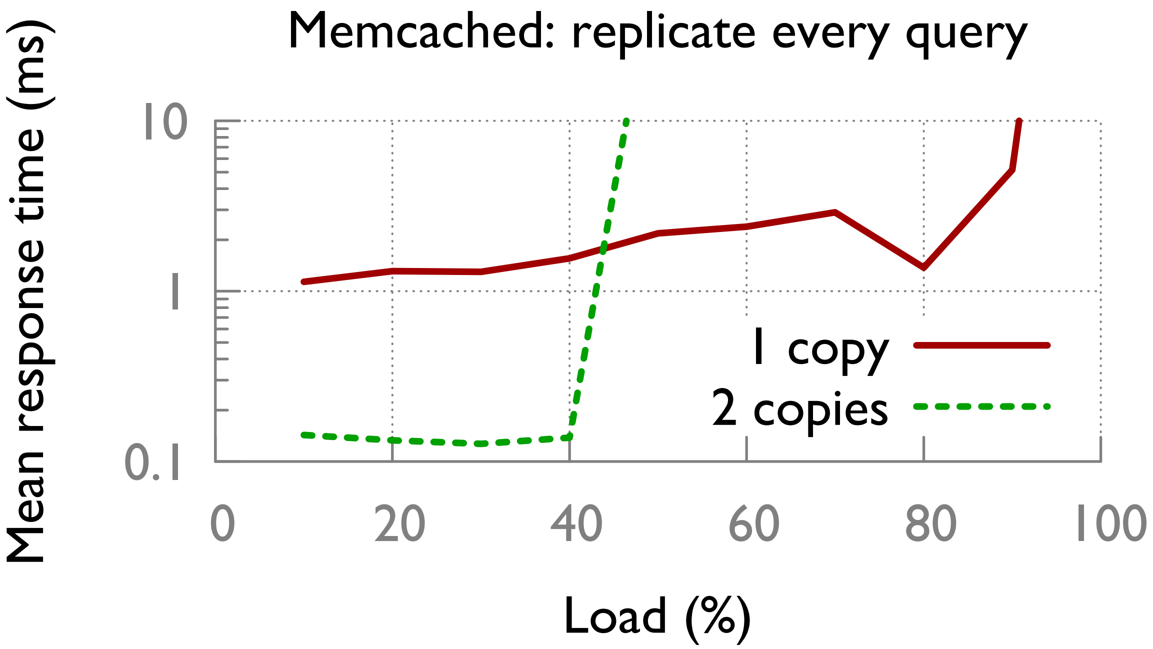
Conjecture: threshold load > 30%
irrespective of service time distribution

Beyond selfishness

Queueing analysis (low variance service time)



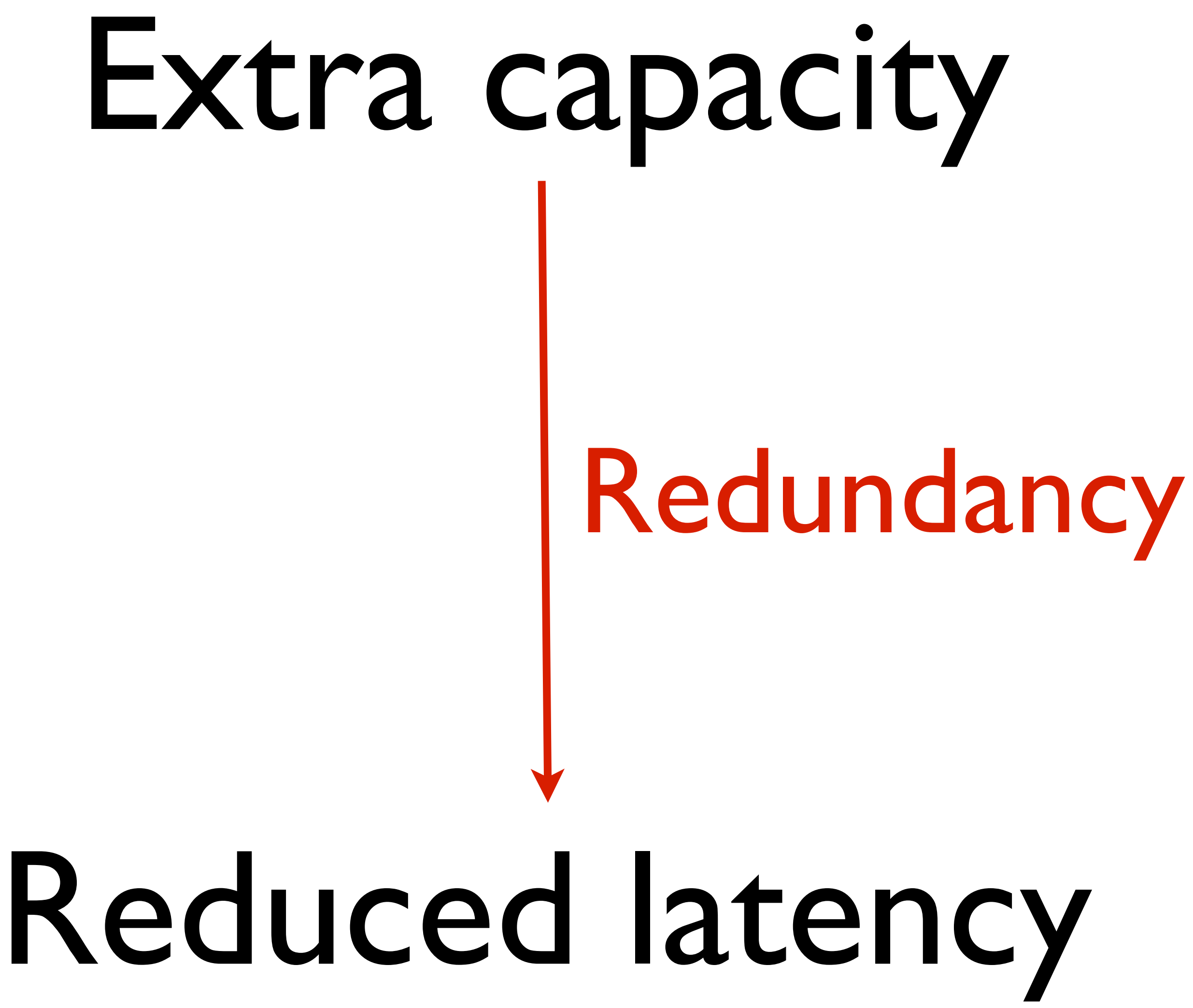
Real system (Memcached/ProtoGENI)



GENI resources

Application	Need	Experiments on
DNS	A	PlanetLab
Multipath Overlay	A, B, C	PlanetLab, ProtoGENI, OpenFlow
Memcached	B	ProtoGENI

A: Realistic background traffic
B: Isolated environment
C: Novel Internet architecture



Extra capacity

Redundancy

Reduced latency

A. Vulimiri, O. Michel, P. B. Godfrey, S. Shenker
“More is less: Reducing latency via redundancy”
HotNets 2012

Supported by NSF grant CNS 1050146

Thank you!

Backup slides

How can you mitigate overhead?

- Strict prioritization
- Redundancy elimination[*]
- Network coding (fractional replication)

[*] Han et al., "RPT: re-architecting loss protection for content-aware networks", NSDI '12