Long Tail of Latency

Day 15

Agenda

- Why is Latency Important?

- Latency in Data Centers

- Reducing Latency through duplications
  - Duplicate Requests
  - Duplicate Storage

Cost of Additional Latency

- +50ms additional latency is okay
- +100ms or more leads to problems
  - Fewer clicks and follow through
  - Smaller revenue

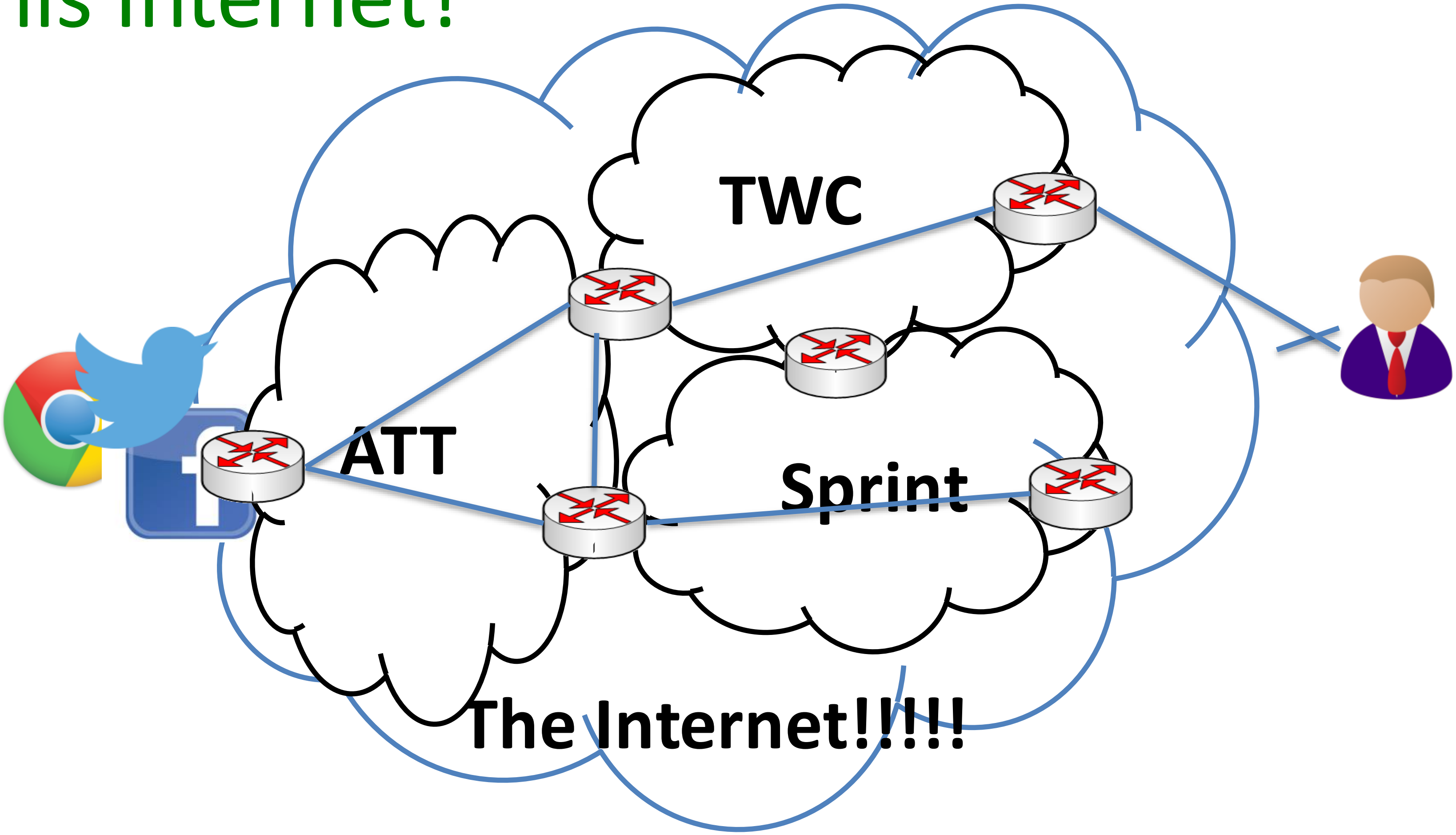
- Important to keep latency low!!!

What Contributes To Latency?

The Internet!!!!!

What Contributes To Latency?

This Internet!



- Within an ISP: RCP, TeXCP
- Across ISP: Overlay Networks, BGP oscillations issues

Ping Times to Some servers

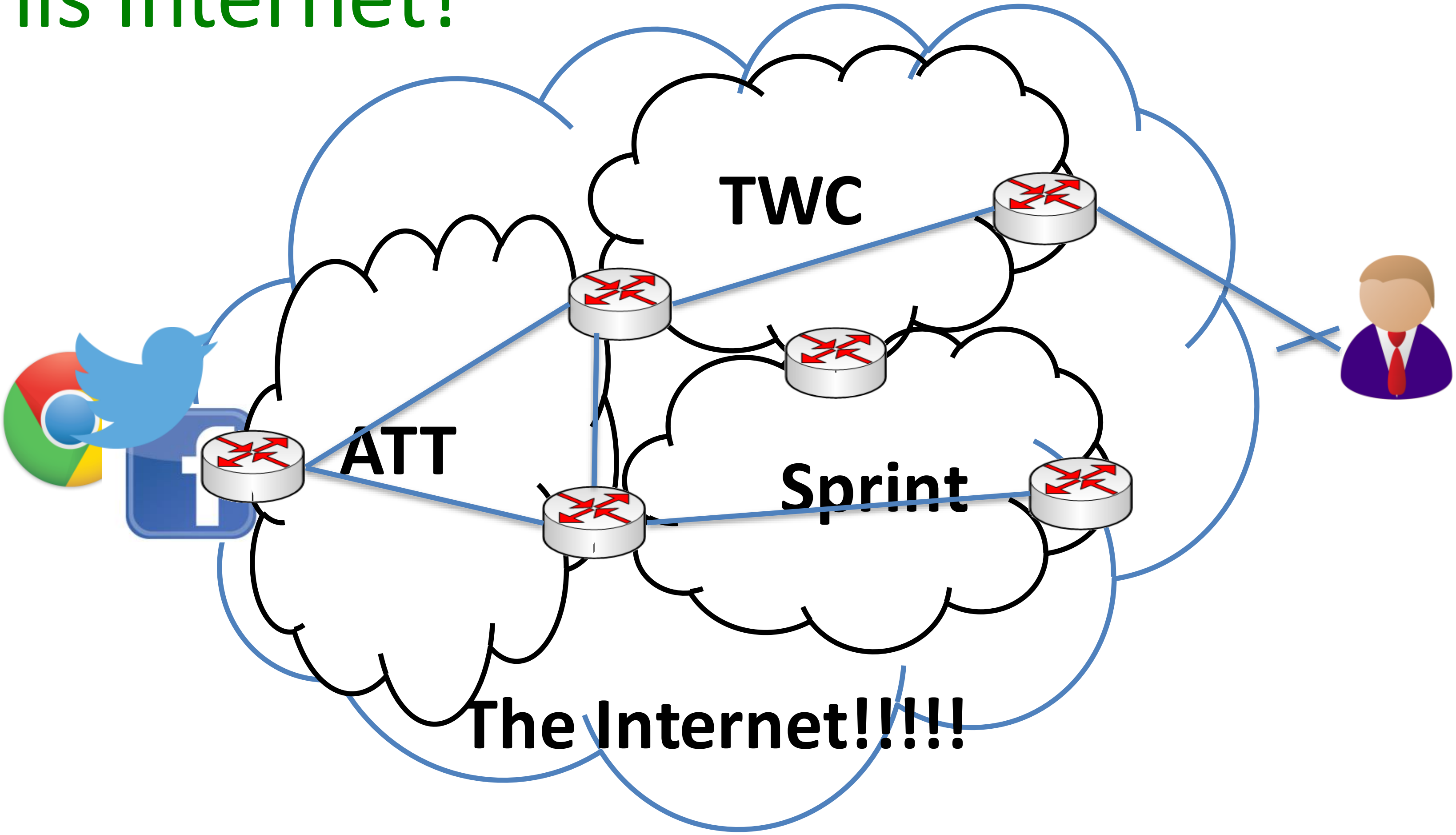
	Round Trip Latency
Google	10ms
Yahoo	37ms
Facebook	16ms
CNN	16ms

Much less than 50ms!!!! Why are we worried about latency?

http://www.factfixx.com/2011/10/13/the-science-of-tangled-cords/

What Contributes To Latency?

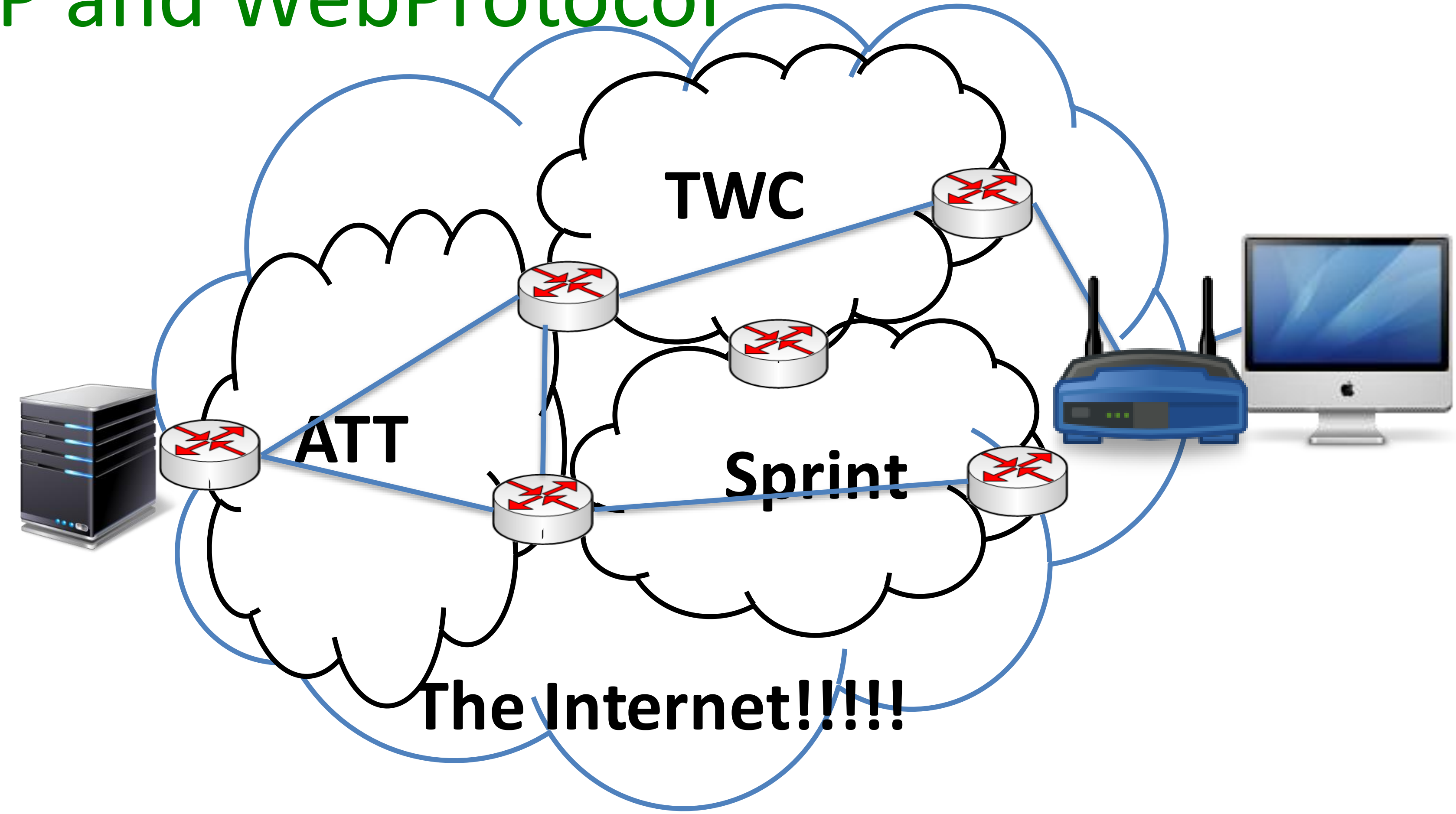
This Internet!



- Within an ISP: RCP, TeXCP
- Across ISP: Overlay Networks, BGP oscillations issues

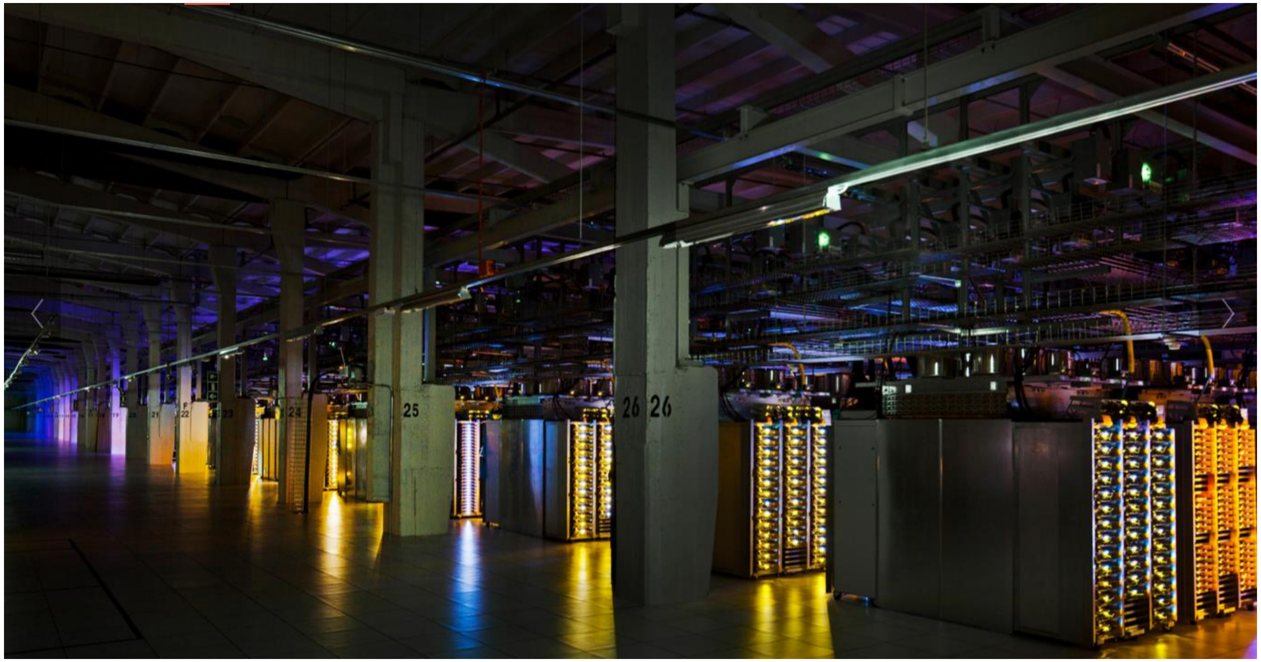
What Contributes To Latency?

TCP and WebProtocol



- Network Multipath: MpTCP
- TCP Overheads: TFO
- Networks with losses: Reducing Web Latency
- Web protocols: SPDY

Server → Data Center

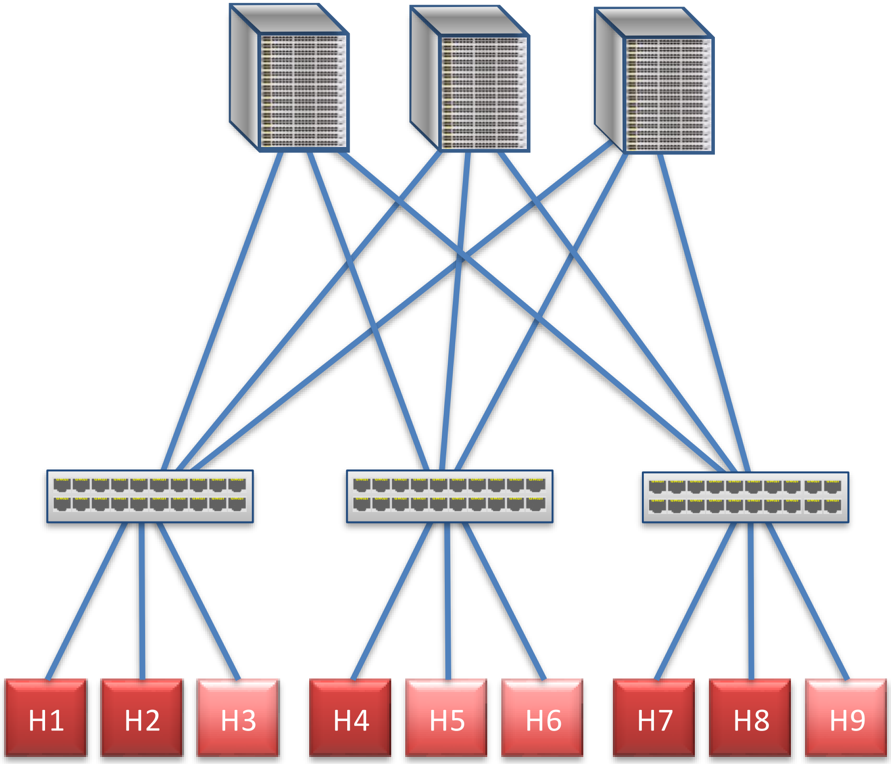


http://www.google.com/about/datacenters/gallery/#/all/14

Agenda

- Why is Latency Important?
- Latency in Data Centers
- Reducing Latency through duplications
  - Duplicate Requests
  - Duplicate Storage

What is a Data Center?



- Servers
  - Run multiple Applications
  - Run background jobs
- Switches
  - Connect servers together

Image courtesy of Mohammad Alizadeh

Source of Latency Within Data Center

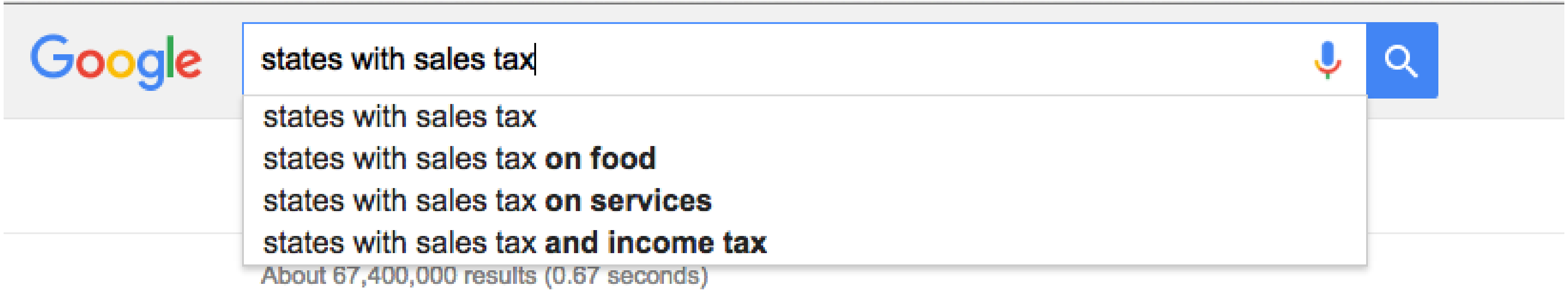
Server Issues

- **Background jobs:**
  - E.g. back-up storage (daemon), clean up garbage, update software (maintenance)
- **Shared resources**
  - Imperfect sharing/scheduling
- **Bad Hardware:**
  - E.g. failing disk
- **Power Saving (energy management):**
  - Slow down CPU to save energy

Network Issues (Global Resource)

- **Not enough resources (.e.g BW)**
  - Network devices are expensive
  - Day 16
- **Inefficient network protocol**
  - Think more TCP overheads
  - Day 17
- **Physical server limits**
  - Many-to-one problem: InCast
  - Day 17

How Do Request Get Processed in a Data Center

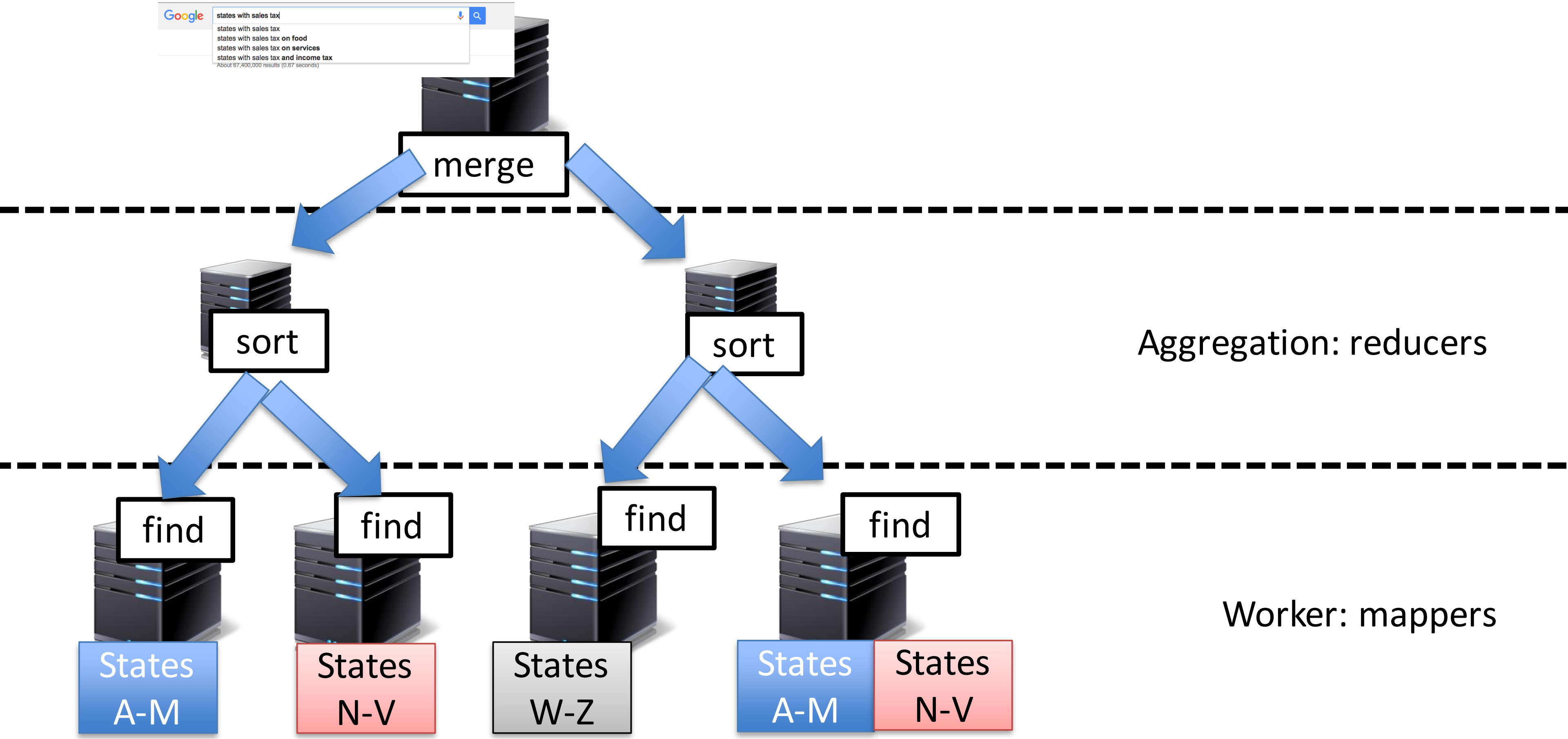


states with sales tax

states with sales tax
states with sales tax **on food**
states with sales tax **on services**
states with sales tax **and income tax**

About 67,400,000 results (0.67 seconds)

How Do Request Get Processed in a Data Center



Google

states with sales tax

states with sales tax
states with sales tax **on food**
states with sales tax **on services**
states with sales tax **and income tax**

About 67,400,000 results (0.67 seconds)

merge

sort

sort

Aggregation: reducers

find

find

find

find

Worker: mappers

States A-M

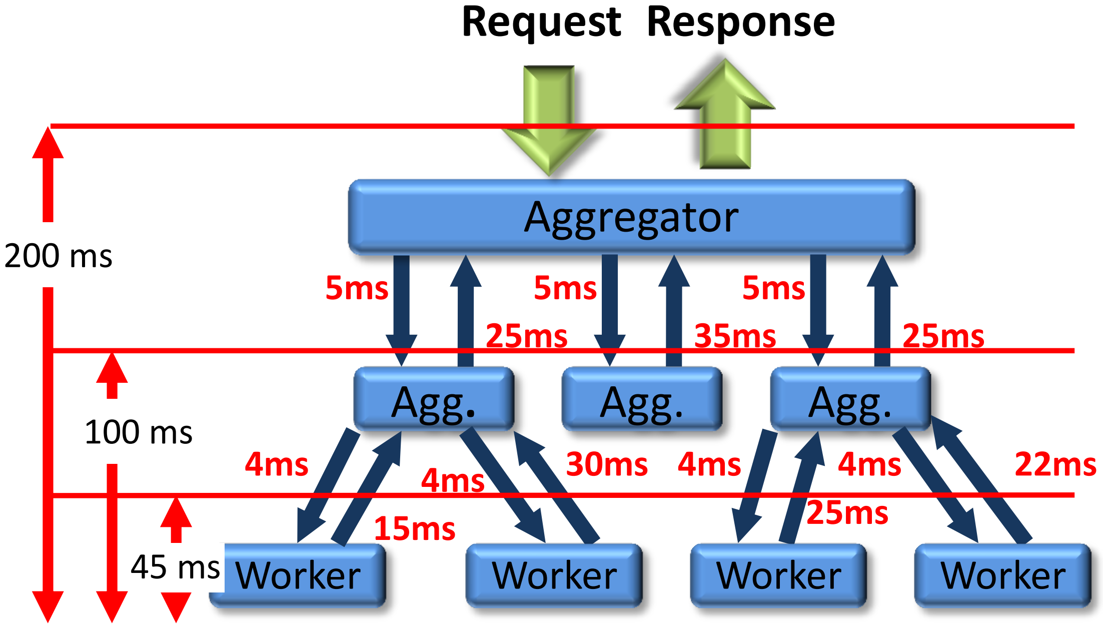
States N-V

States W-Z

States A-M

States N-V

User-facing online services



Application SLAs

Cascading SLAs

SLAs for components at each level of the hierarchy

Network SLAs

Deadlines on communications between components

Importance of Tail Latency

- Tail Latency == 1 in X servers being slow

- Why is this bad?

Importance of Tail Latency

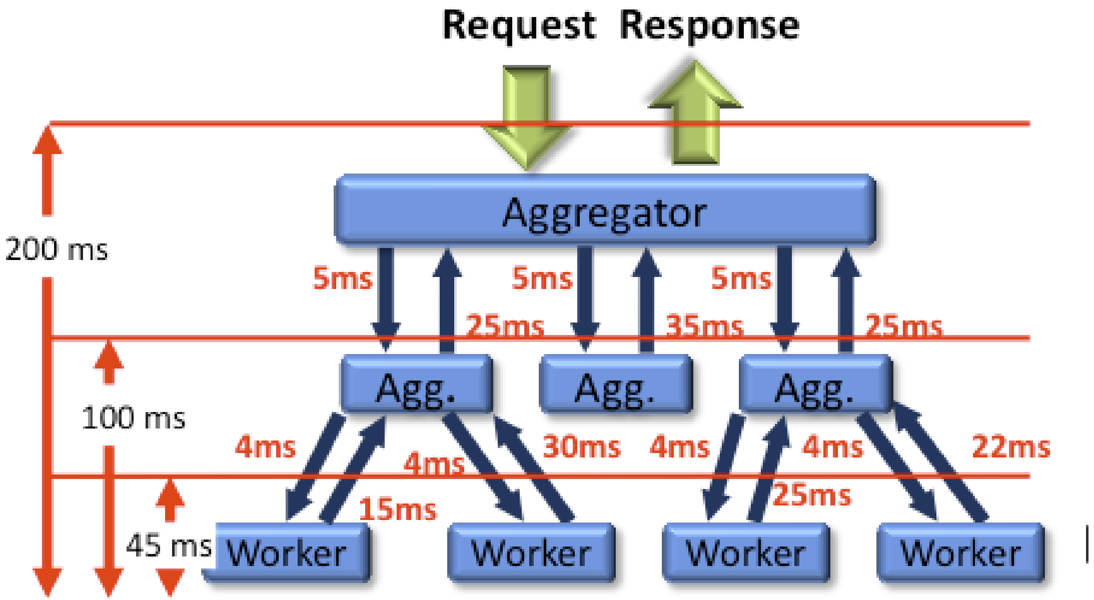
- Tail Latency == 1 in X servers being slow

- Why is this bad?
  - Each user request now needs several servers
  - Changes of experience tail is much higher

- If one in 100 servers has high latency (1% are bad)
  - If users needs 100 partitions then chances of latency is (63%): **MUCH HIGHER!!!!**

Respond with "Good Enough" Results

- Better to give the user less than perfect results rather than loose the user
- If a machine doesn't respond before its deadline ignore it



Basic Latency Reduction Techniques

- Use priority queues
  - (Think HOV lanes)
  - User traffic Higher priority
  - Background traffic low priority

- Reduce head of line blocking
  - Break large requests into smaller ones

- Rate-limit background activity
- Stop low priority until high priority is done

Source of Latency Within Data Center

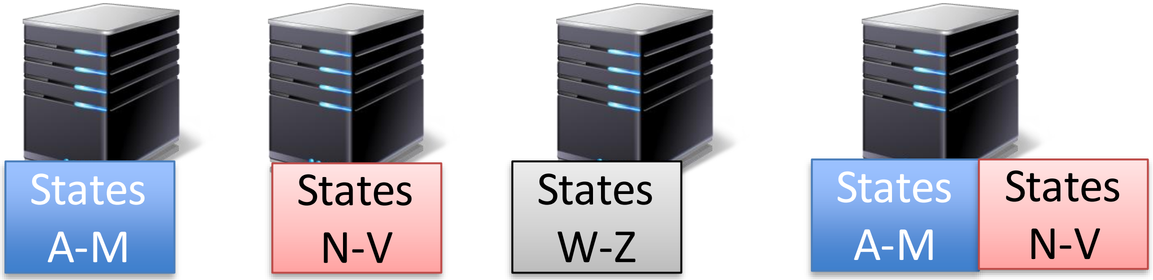
Server Issues	**Solutions**
• **Background jobs:** – E.g. back-up storage (daemon), clean up garbage, update software (maintenance)	• Make them happen at the same time. Only affect jobs running at that time.
• **Shared resources** – Imperfect sharing/scheduling	
• **Bad Hardware:** – E.g. failing disk	• Quarantine bad machines
• **Power Saving (energy management):** – Slow down CPU to save energy	• Minimize power savings

Agenda

- Why is Latency Important?
- Latency in Data Centers
- Reducing Latency through duplications
  - Duplicate Requests
  - Duplicate Storage

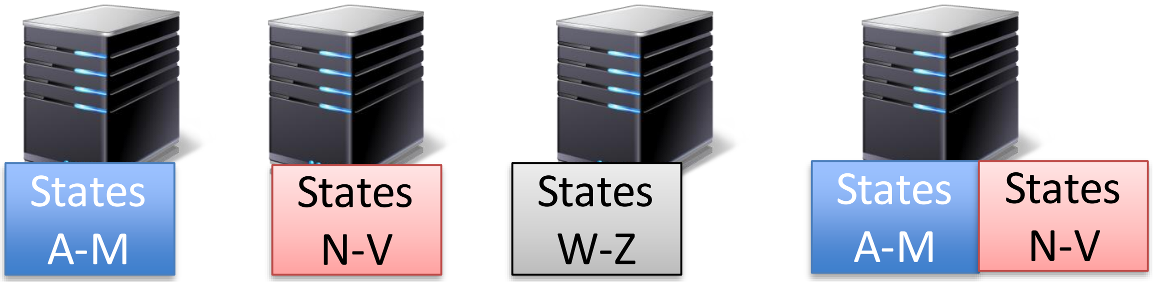
Dealing with Slow Processing With Replication

- Replicate Processing
  - If a request is slow: Start a new one!!
  - New request may run on a machine with no problems
- Why is this insufficient?



Dealing with Slow Processing With Replication

- All requests process data: e.g. queries about state tax processes US state data.
- Duplicating the request may not help if the new request uses the same data.
- **We need to perform data replication also.**



States A-M

States N-V

States W-Z

States A-M

States N-V

Agenda

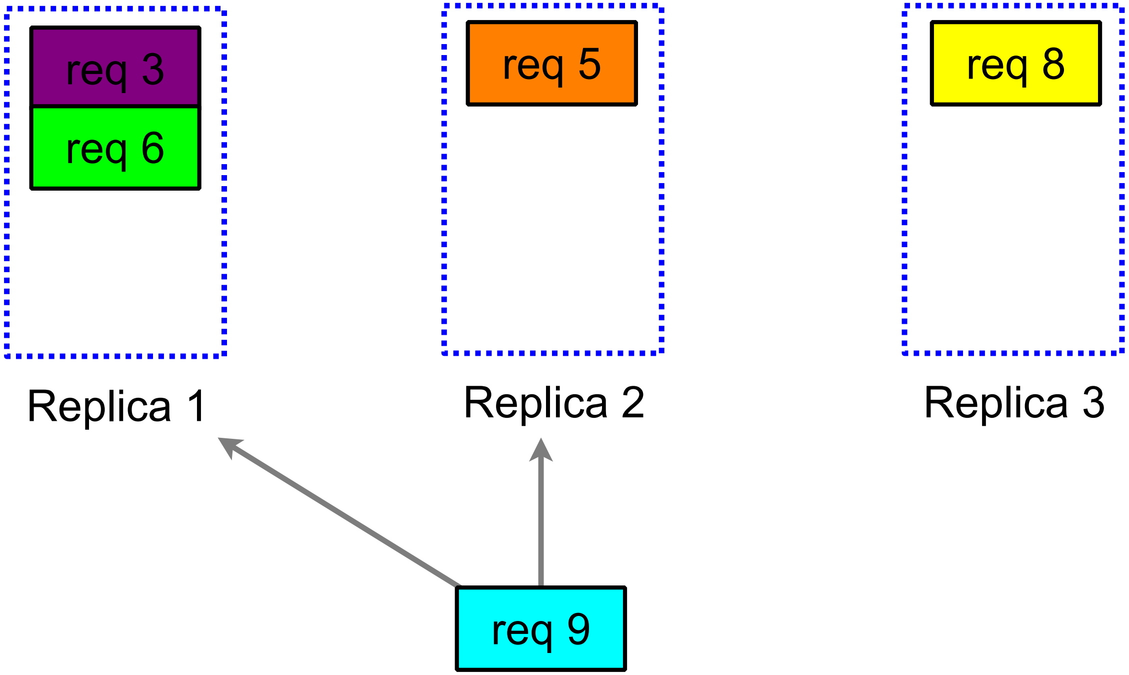
- Why is Latency Important?
- Latency in Data Centers
- Reducing Latency through duplications
  - Duplicate Requests
  - Duplicate Storage

How to Replicate Processing?

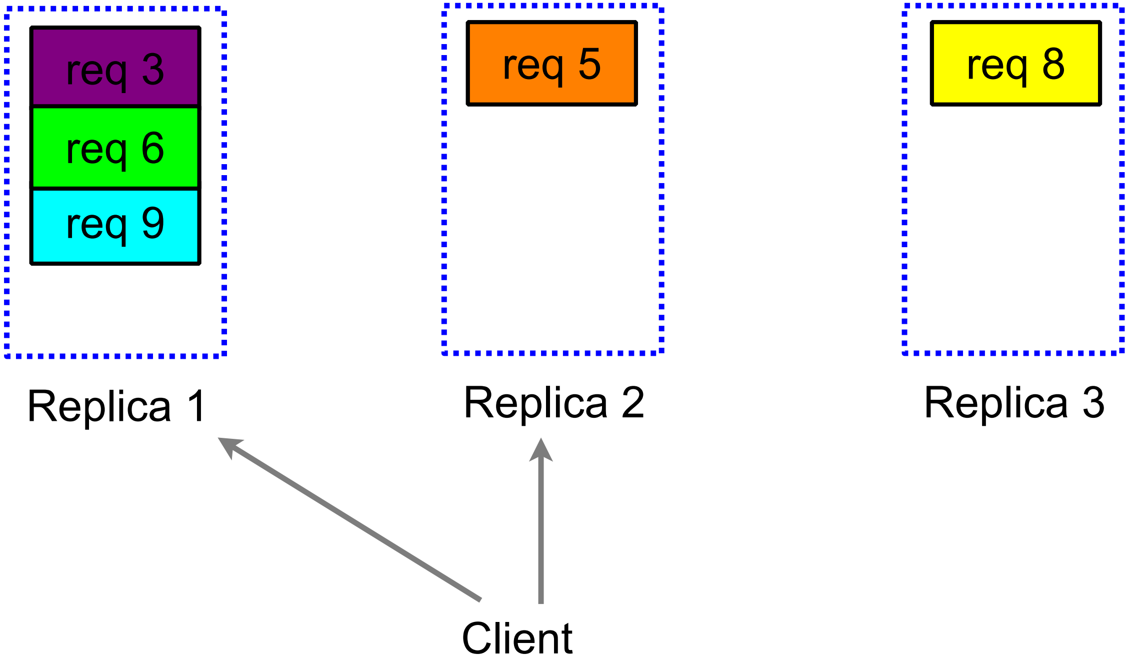
- When to start replication?
- How many replicas to make?
- How to deal with replica results?
- Replicas waste resources: how to minimize waste?

Next set of slides are from Jeff Dean's Achieving Rapid Response Times in Large Online Services

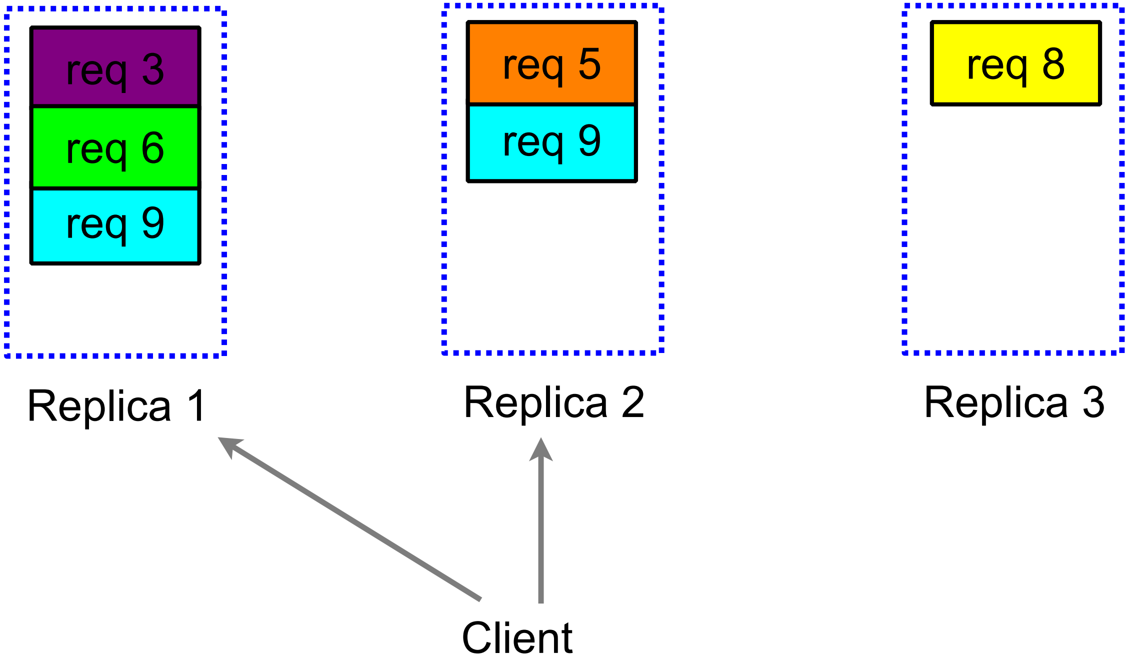
Backup Requests



Backup Requests



Backup Requests



req 3

req 6

req 9

Replica 1

req 5

req 9

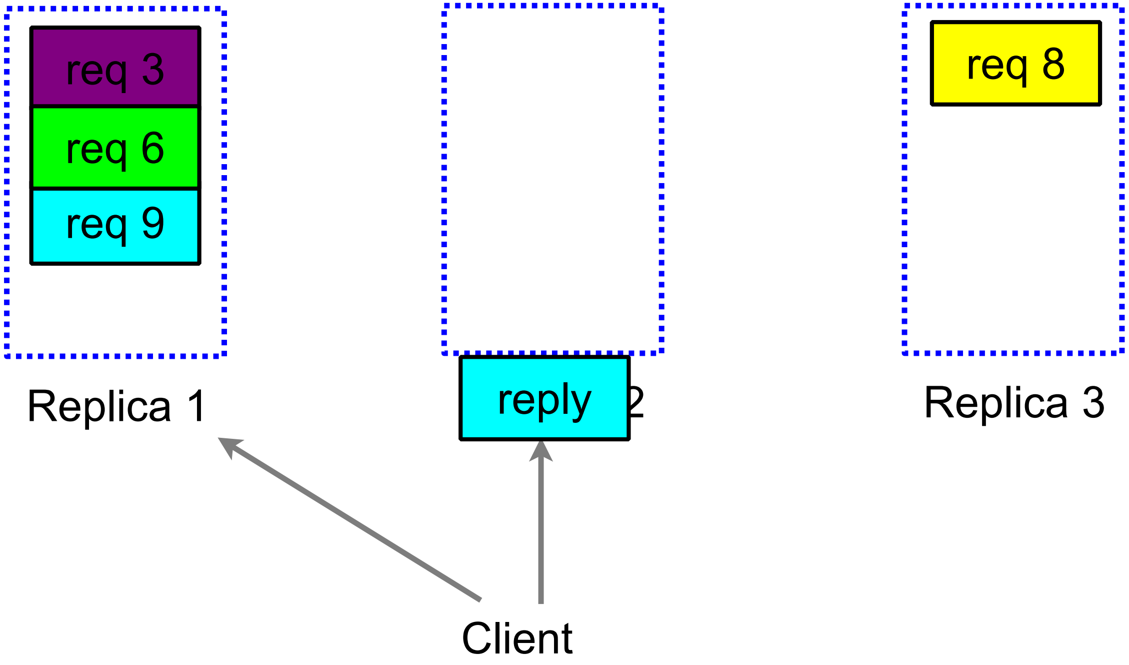
Replica 2

req 8

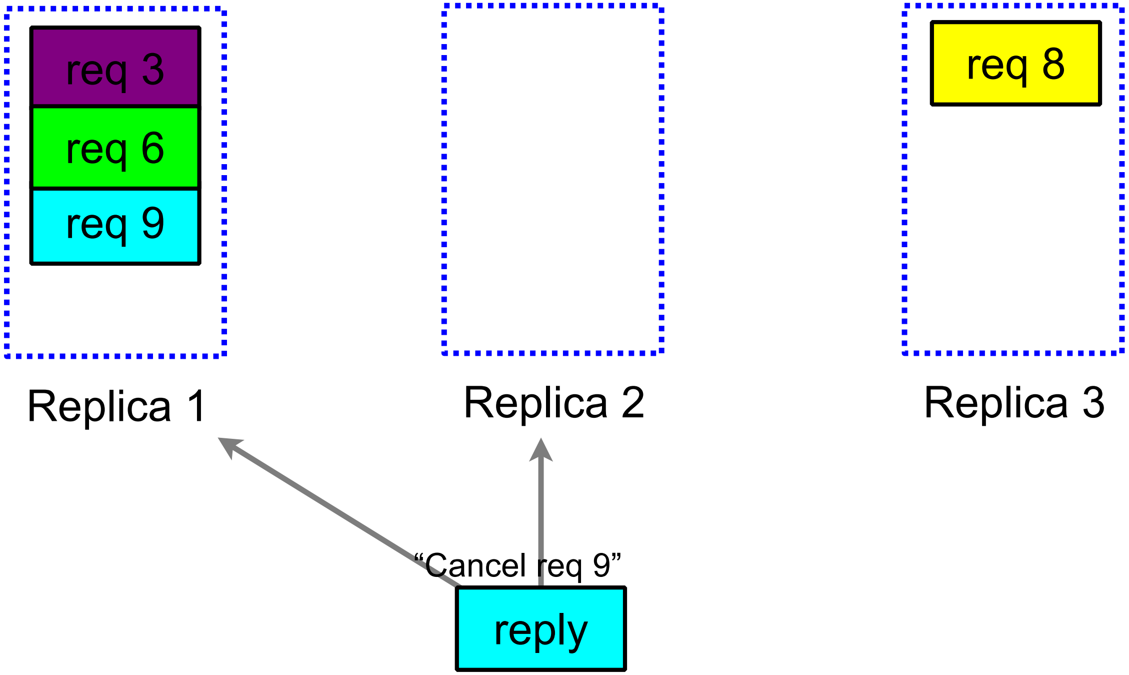
Replica 3

Client

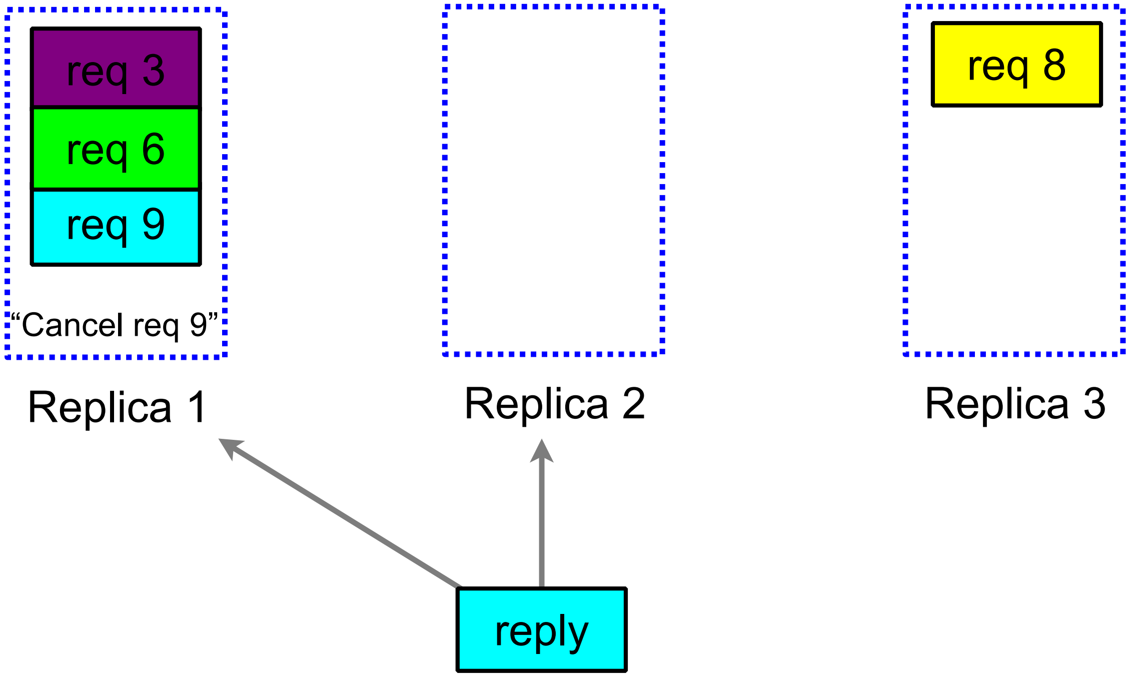
Backup Requests



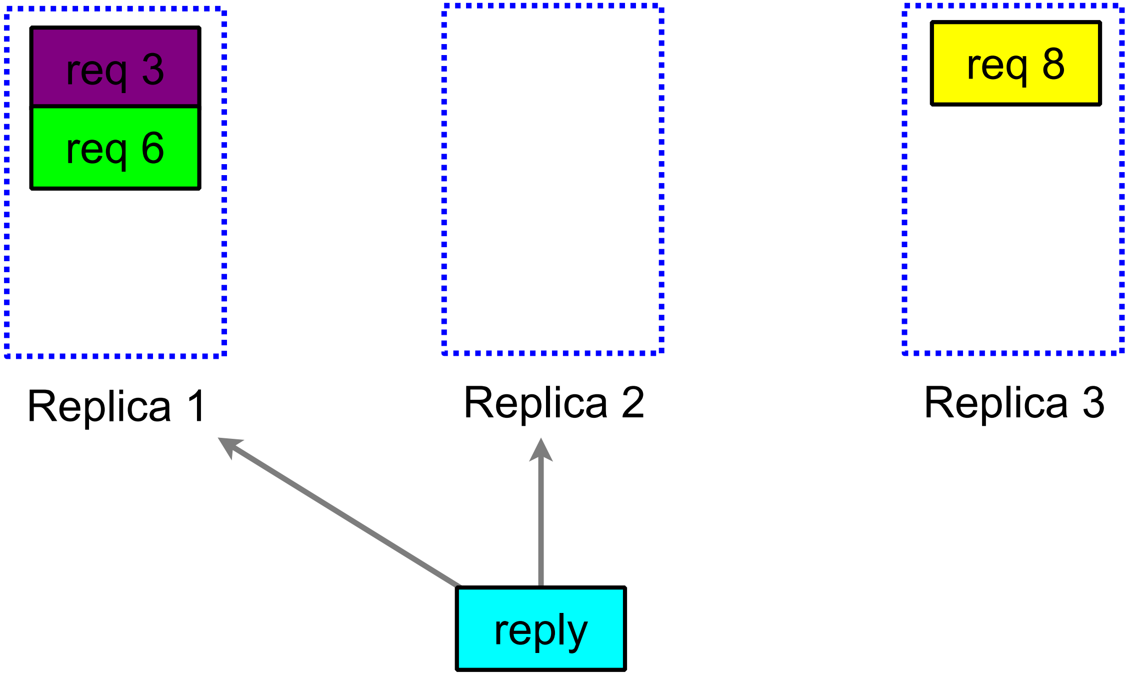
Backup Requests



Backup Requests



Backup Requests



Backup Requests Effects

- In-memory BigTable lookups
  - data replicated in two in-memory tables
  - issue requests for 1000 keys spread across 100 tablets
  - measure elapsed time until data for last key arrives

	Avg	Std Dev	95%ile	99%ile	99.9%ile
No backups	33 ms	1524 ms	24 ms	52 ms	994 ms
Backup after 10 ms	14 ms	4 ms	20 ms	23 ms	50 ms
Backup after 50 ms	16 ms	12 ms	57 ms	63 ms	68 ms

- Modest increase in request load:
  - 10 ms delay: <5% extra requests; 50 ms delay: <1%

Google

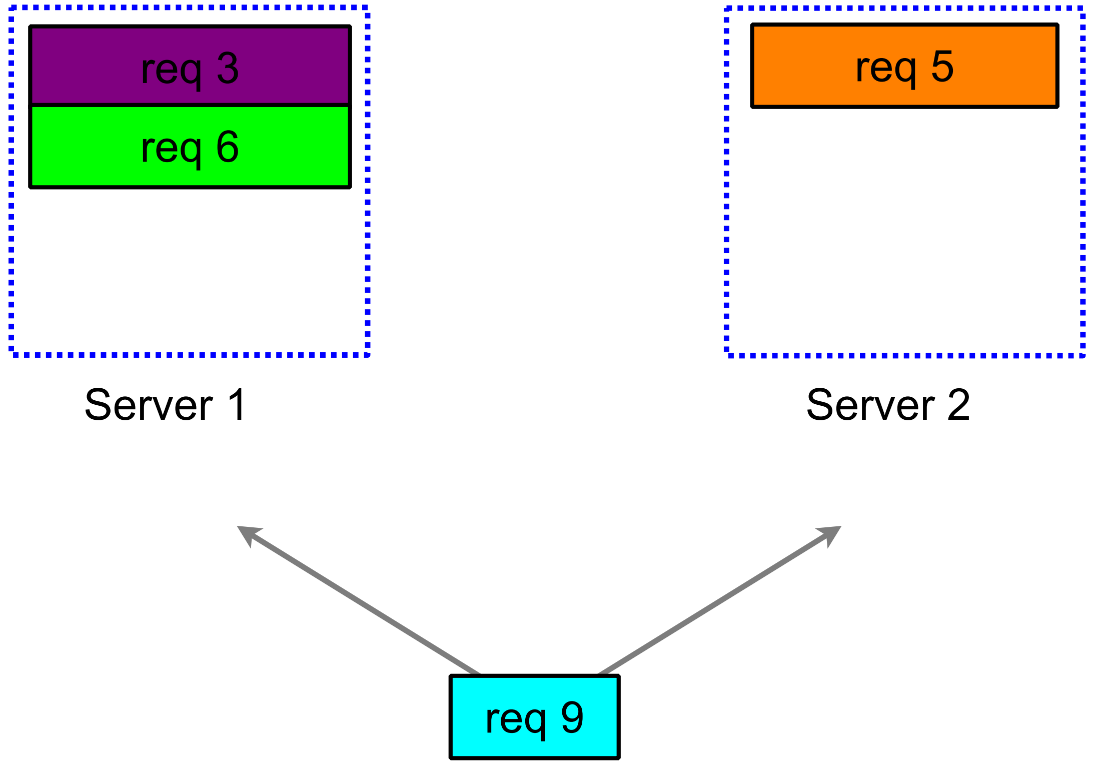
Backup Requests Effects

Can we reduce the back-up time even further? Maybe 0ms?
How do we minimize overheads?

	Avg	Std Dev	95%ile	99%ile	99.9%ile
No backups	33 ms	1524 ms	24 ms	52 ms	994 ms
Backup after 10 ms	14 ms	4 ms	20 ms	23 ms	50 ms
Backup after 50 ms	16 ms	12 ms	57 ms	63 ms	68 ms

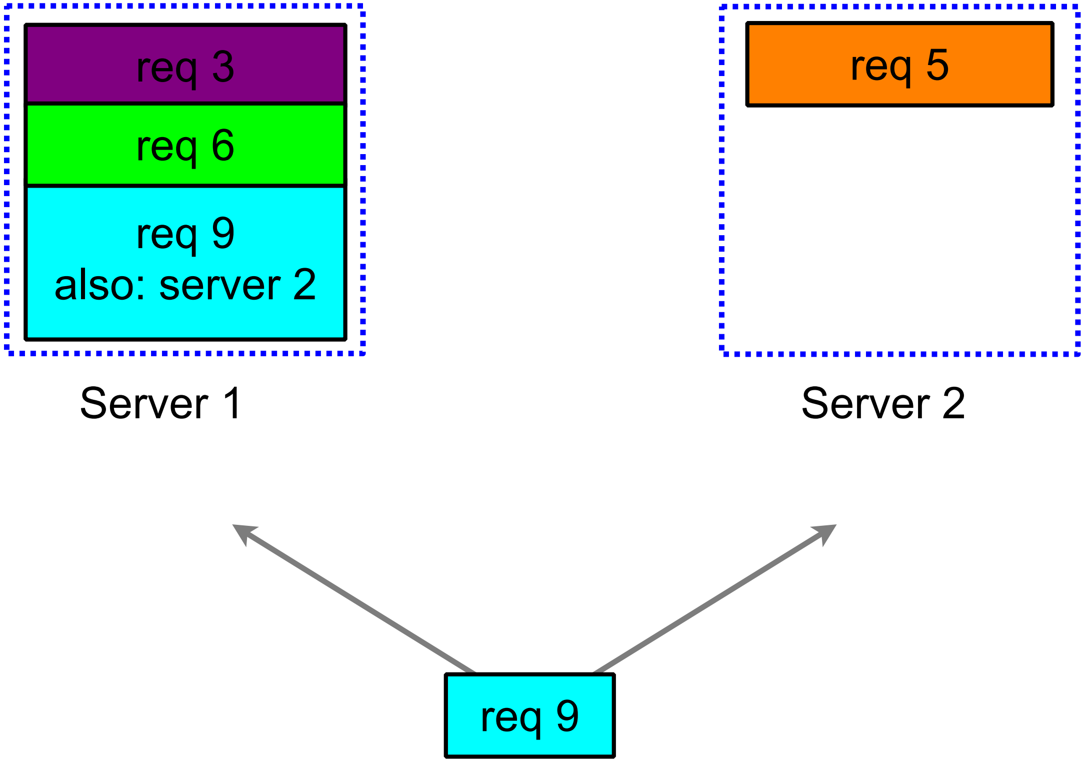
- Modest increase in request load:
  - 10 ms delay: <5% extra requests; 50 ms delay: <1%

Backup Requests w/ Cross-Server Cancellation



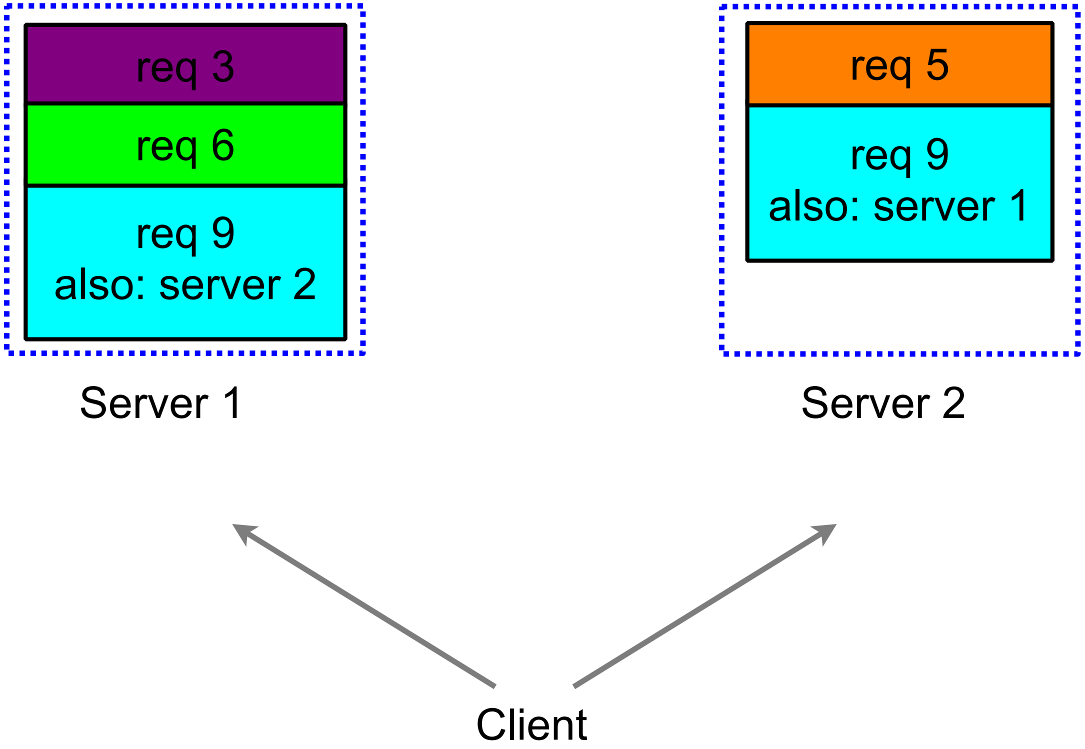
Google

Backup Requests w/ Cross-Server Cancellation



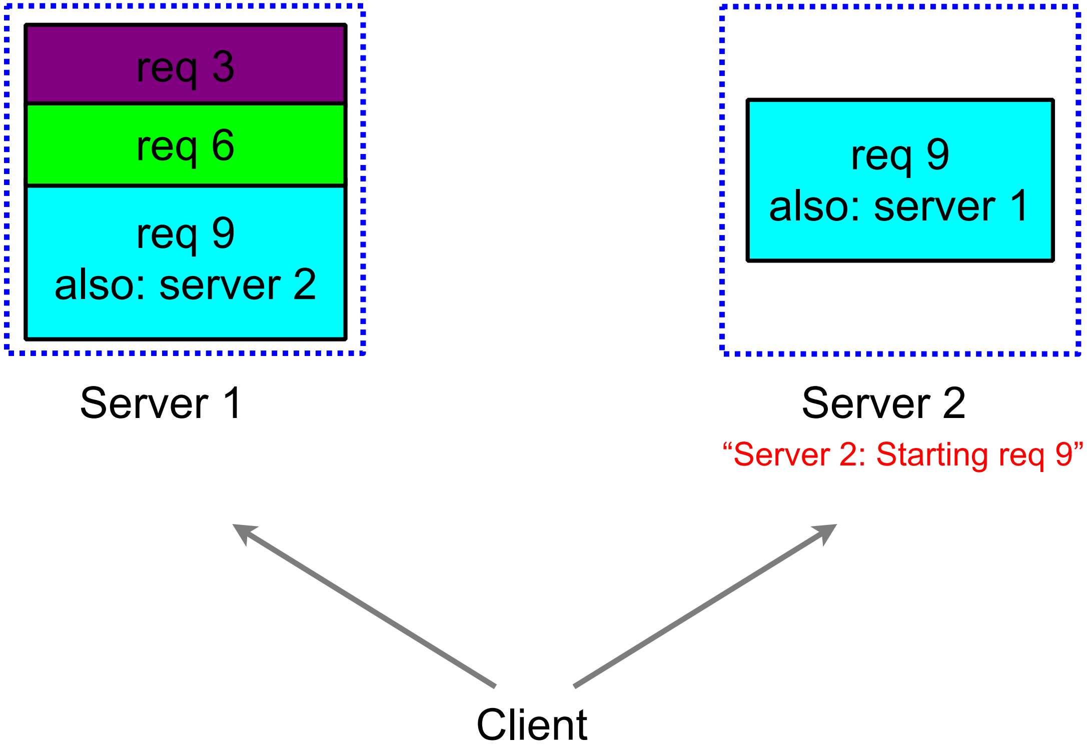
Each request identifies other server(s) to which request might be sent

Backup Requests w/ Cross-Server Cancellation



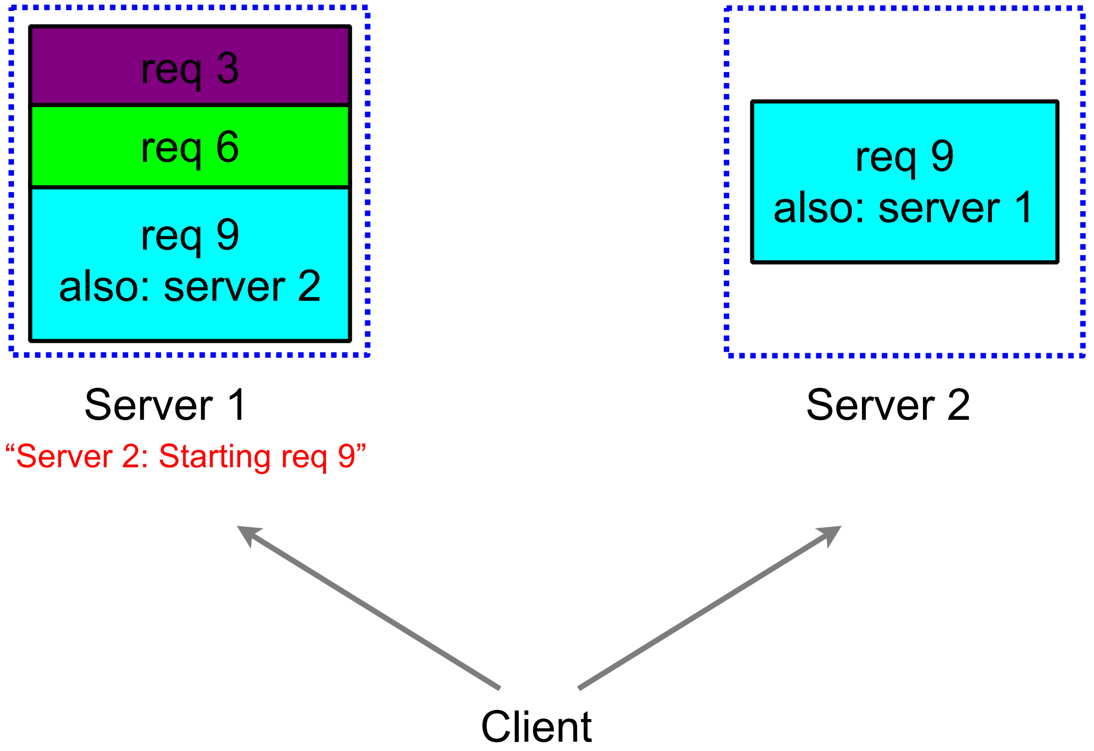
Each request identifies other server(s) to which request might be sent

Backup Requests w/ Cross-Server Cancellation



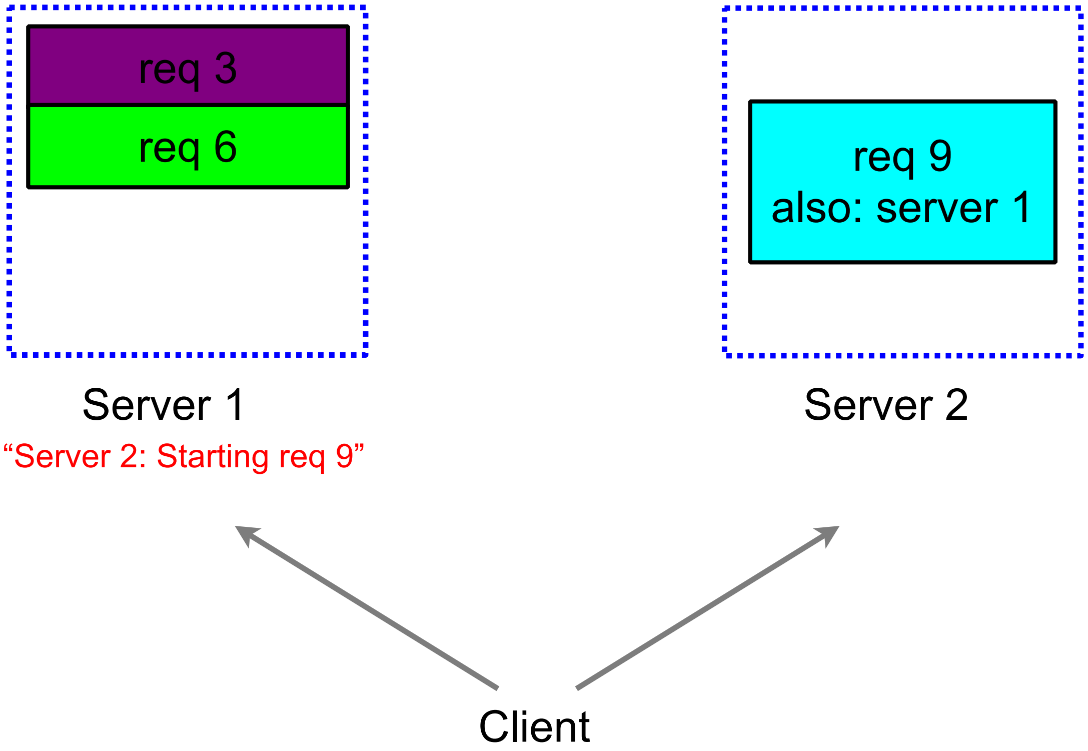
Each request identifies other server(s) to which request might be sent

Backup Requests w/ Cross-Server Cancellation



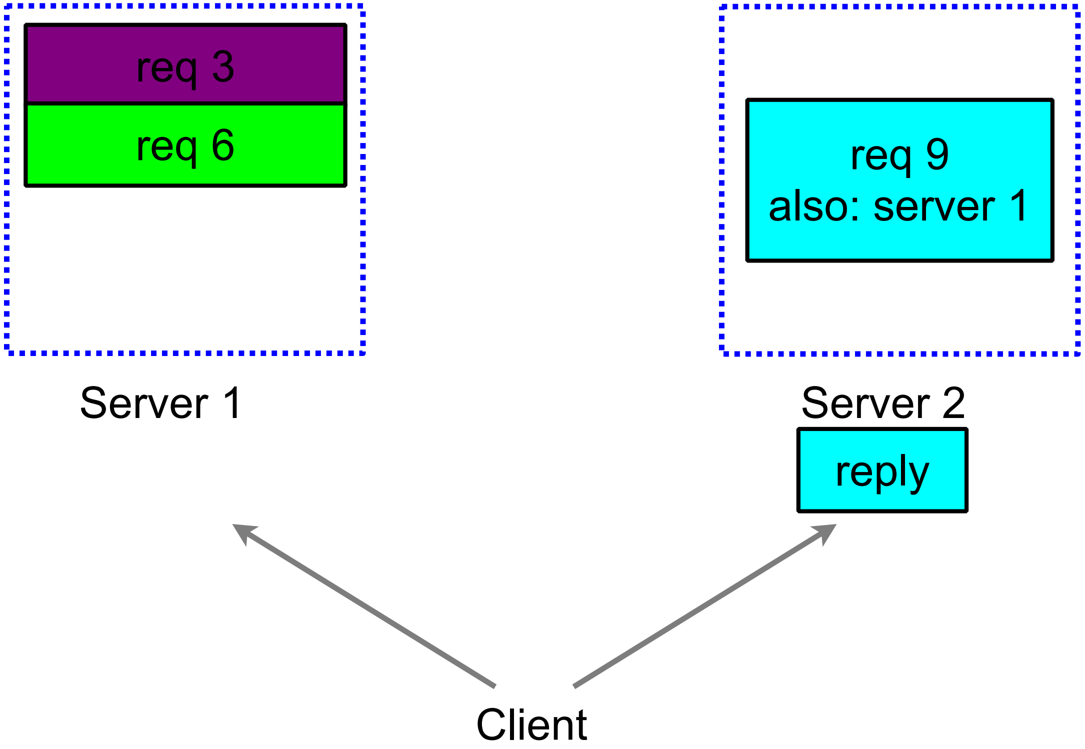
Each request identifies other server(s) to which request might be sent

Backup Requests w/ Cross-Server Cancellation



Each request identifies other server(s) to which request might be sent

Backup Requests w/ Cross-Server Cancellation



Each request identifies other server(s) to which request might be sent

Backup Requests w/ Cross-Server Cancellation

- Read operations in distributed file system client
  - send request to first replica
  - wait 2 ms, and send to second replica
  - servers cancel request on other replica when starting read
- Time for bigtable monitoring ops that touch disk

Cluster state	Policy	50%ile	90%ile	99%ile	99.9%ile
Mostly idle	No backups	19 ms	38 ms	67 ms	98 ms
	Backup after 2 ms	16 ms	28 ms	38 ms	51 ms

-43%

When Can this Go Wrong?

Agenda

- Why is Latency Important?
- Latency in Data Centers
- Reducing Latency through duplications
  - Duplicate Requests
  - Duplicate Storage

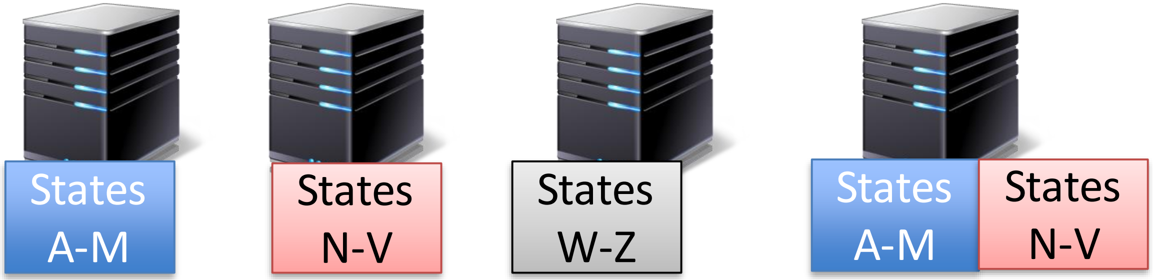
How to Replicate Storage?

- Which data to replicate?
- Where to place the replicated data?
- Replicas waste resources: how to minimize waste?

Next set of slides are from Jeff Dean's Achieving Rapid Response Times in Large Online Services

Storage Issues

- What happens if all queries are for Wisconsin?



States A-M

States N-V

States W-Z

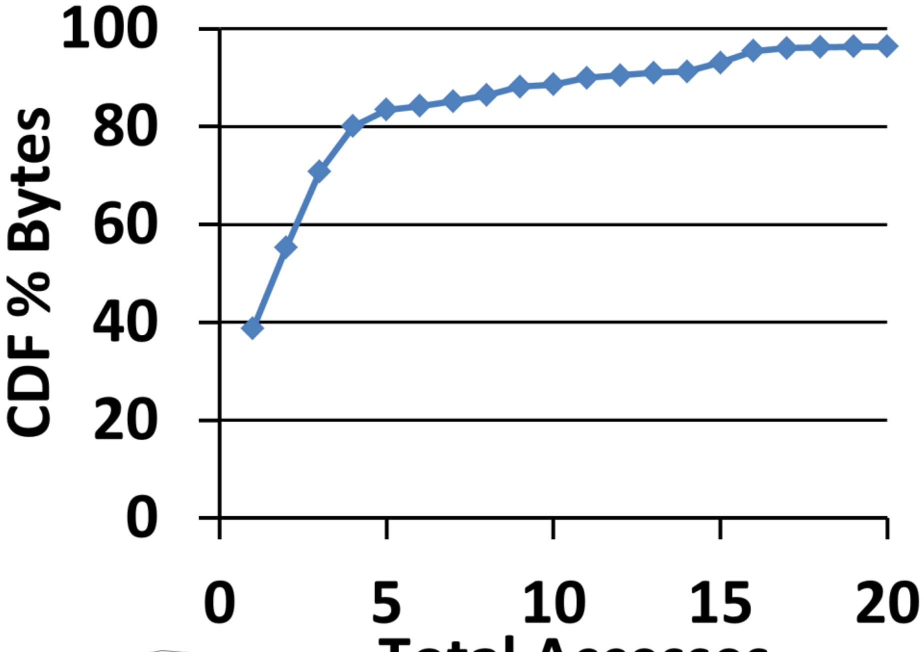
States A-M

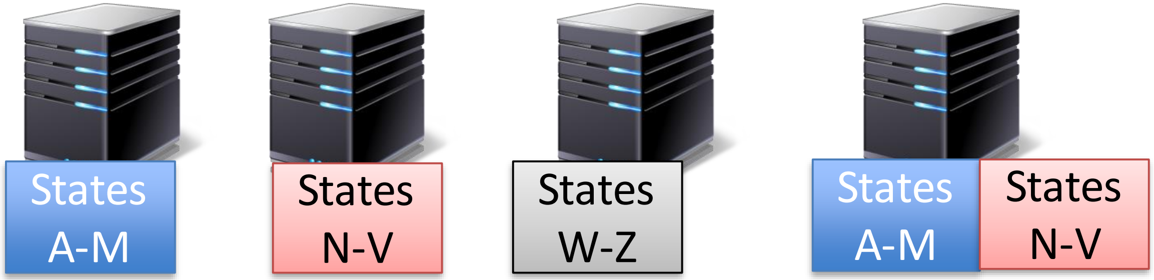
States N-V

Popularity Skew

- According to Microsoft’ data

- Top 12% is 10x more popular than bottom third





States A-M

States N-V

States W-Z

States A-M

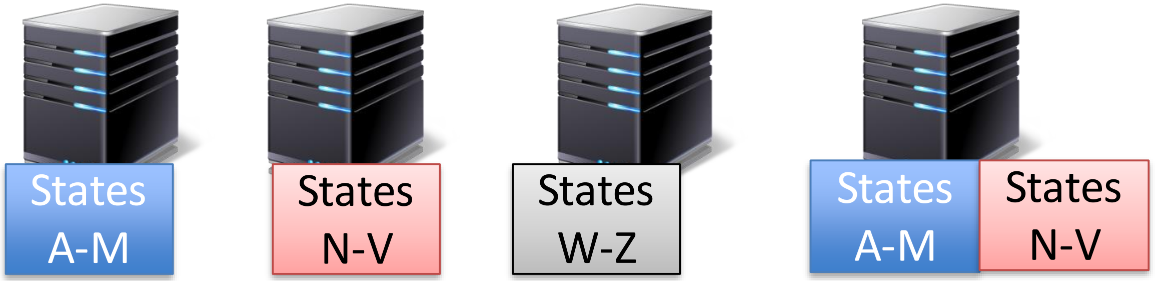
States N-V

Graph from Scarlett **: Coping with Skewed Content Popularity in MapReduce Clusters**

Solution: Make Copies of Popular Content

States X-Z

Where to put
This new chunk?



States A-M

States N-V

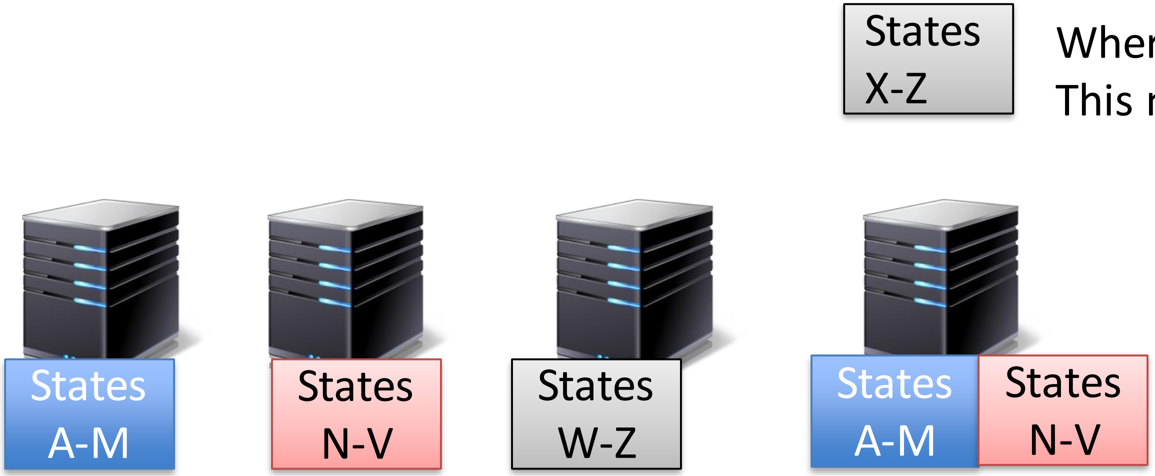
States W-Z

States A-M

States N-V

- If “W” is popular, I make copies of them:

Solution: Make Copies of Popular Content



States X-Z

Where to put
This new chunk?

States A-M

States N-V

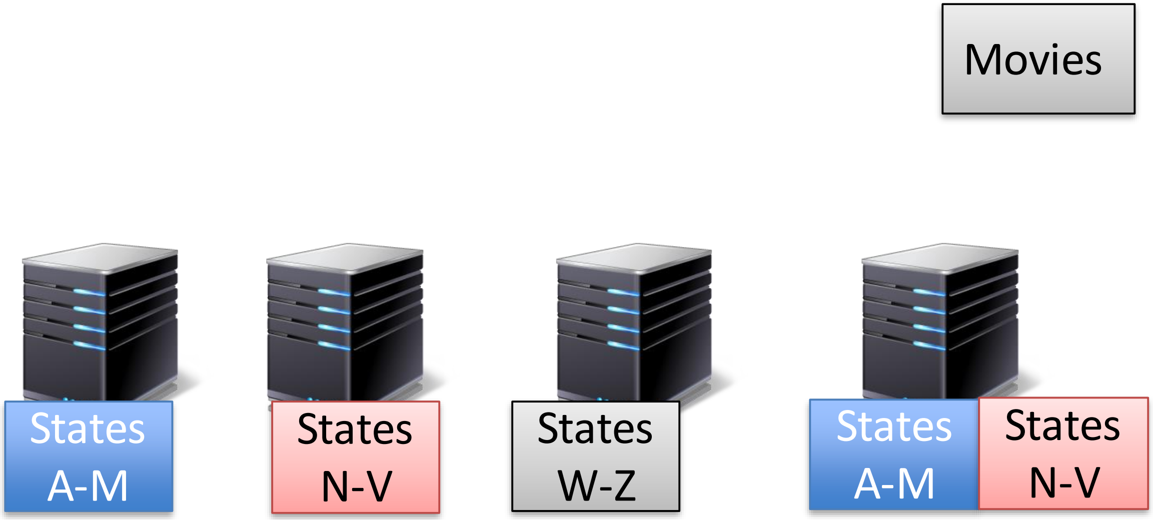
States W-Z

States A-M

States N-V

- If "W" is popular, I make copies of them:
  - – Avoiding putting both copies on the same server
  - – Avoid putting the copy on a server with other popular content (Load Balancing)

Load balance chunk across servers



- Calculate predicted ‘load’: Total Access x Size
  - Place on replica chunks on least ‘loaded’

When to Replicate Storage Chunks?

- Automated:
  - Monitor utilization of chunks
  - Replicate more utilized chunks

- Static:
  - Always replicate chunks of a particular type

Concluding Remarks

- Tail Latency is costly → Users will leave the system.

- Several approaches to improve tail latency leverage replication

- Replicate improves overheads, why are they acceptable?
  - Replication is also used to tackle failures:
  - These same copies can be used to tolerate variability
  - Times scales are very different:
    - Variability: requests with performance issues happen frequently: 1000s of disruptions/sec, scale of **milliseconds**
    - Faults: failure happen infrequently: 10s of failures per day, scale of **tens of seconds**

Reminder

- Project Proposal Due Tomorrow @ Noon!!!