CS 188

Scalable Internet Services

Andrew Mutz

December 3, 2019

Calendar Updates

Final presentations are Thursday and Friday!

Writeups are due Thursday at noon (12pm), emailed to me: andrew.mutz@cs.ucla.edu

Thursday’s presentations are in this room, Friday’s are in the room of your lab session.

Calendar Updates

Team presentation time and order:

https://piazza.com/class/k0wl5njhn3y45c?cid=97

Thursday 12pm-2PM
12:00 PM Poolfolio
12:15 PM Newbie
12:30 PM Wisowls
12:45 PM Pass Me the Aux
1:00 PM GGEZ
1:15 PM WHISPR
1:30 PM password
1:45 PM JOB I/O

Friday 12PM-2PM
12:00 PM Radius on Rails
12:15 PM Arkamatch
12:30 PM spinder
12:45 PM Sails in Jails
1:00 PM Slow Starters
1:15 PM SwapSpace
1:30 PM Incognito
1:45 PM Match

Friday 2PM-4PM
2:00 PM OpenAux
2:15 PM Swiper
2:30 PM DEAZY
2:45 PM PlayNow
3:00 PM LinksIn
3:15 PM dinnerpal
3:30 PM Team Temp
3:45 PM Hot Dev Fall
4:00 PM Team SoftBank

Calendar Updates

If you are still load testing, please terminate after use (don't wait for the auto-termination script).

We are limited in number of concurrent EB stacks we can deploy.

Starting Thursday at noon, max one deploy per team.

Calendar Updates

You should have no need to manually create instances in the EC2 console.

You should only need to use elastic beanstalk.

You should never create instance types other than M, R, C and T

Today's Agenda

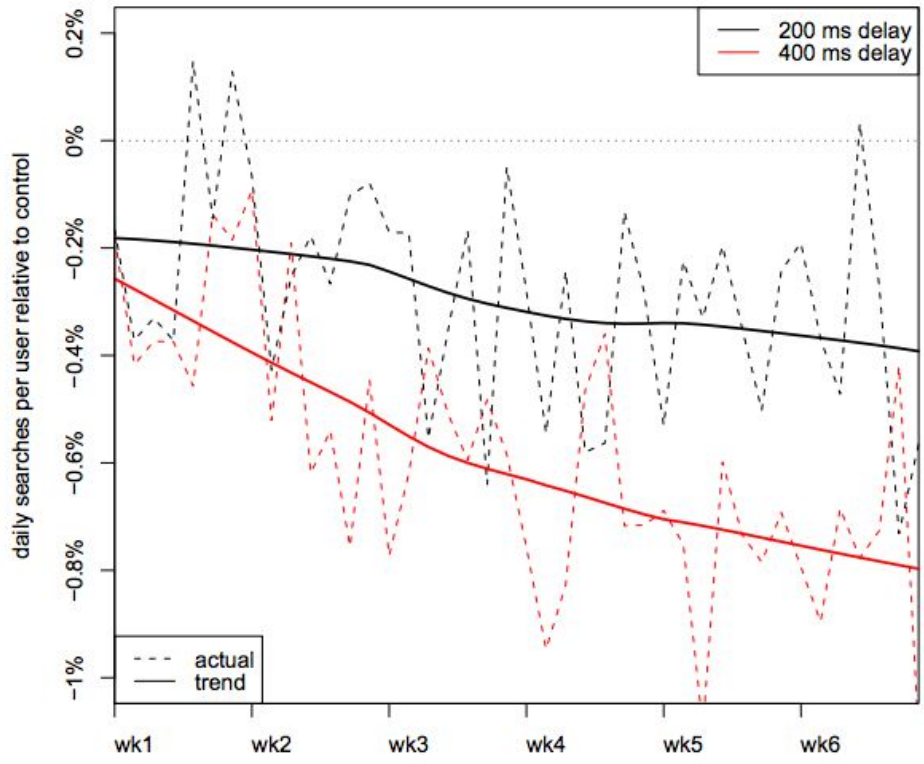
- Content Delivery Networks
  - Motivation
  - Implementation
- Course Conclusion

Motivation

Speed Matters: Google

- Slowing searches down has a measurable effect on user behavior
  - Slowing 200-400ms decreased the rate of future searches per user by 0.2-0.6%
- Users remember slow performance
  - 200ms delay -> 0.22% trending to 0.36% fewer
  - 400ms delay -> 0.44% trending to 0.74% fewer

Motivation



- Users remember poor performance
    - 400ms delay resulted in 0.21% fewer searches for five weeks following the experiment with delays removed
- Effect on pagerank

Motivation

Speed matters: Walmart

- Walmart chose a real user monitoring (RUM) approach to measure page load times on walmart.com
  - Used Boomerang.js to measure time between page head and window.onload (all content loaded)
  - Found low page load times were positively correlated with conversion rate

Motivation

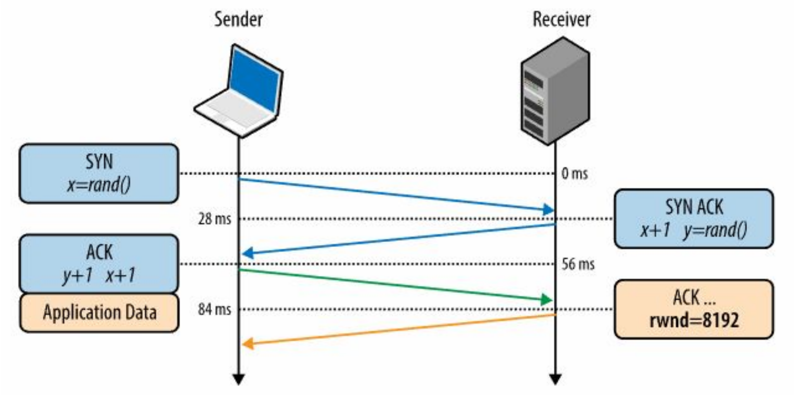

Impact of site performance on overall site conversion rate….
Baseline – 1 in 2 site visits had response time > 4 seconds
* Sharp decline in conversion rate as average site load time increases from 1 to 4 seconds
* Overall average site load time is lower for the converted population (3.22 Seconds) than the non-converted population (6.03 Seconds)
Conversion Rate Vs. Load Time
Population (%)
Conversion Rate (%)
0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10
10-11
11-12
12-13
13-14
14-15
>15
Load Time (Seconds)
Note: Load Time here is the time taken from head of the page to page ready (T_Page)
Page Performance & Site Conversion – Feb 2012
Walmart
37

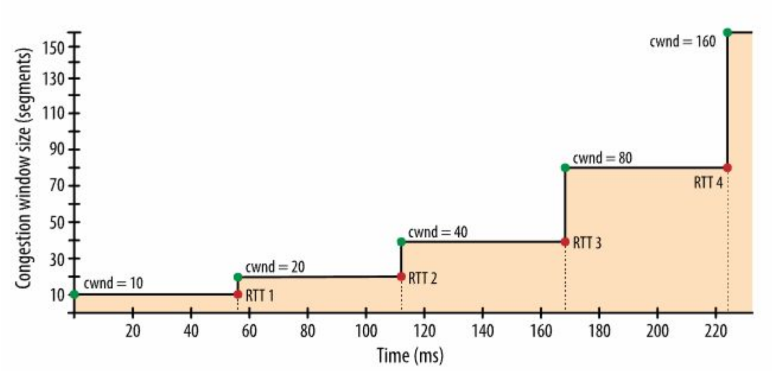
Motivation

Speed matters: Walmart

- Team set performance goals for particular pages based on measurements
- Improving page load times by 1 second resulted in up to a 2% increase in conversion rates
- Improving page load times by 100ms resulted in as much as 1% revenue increase

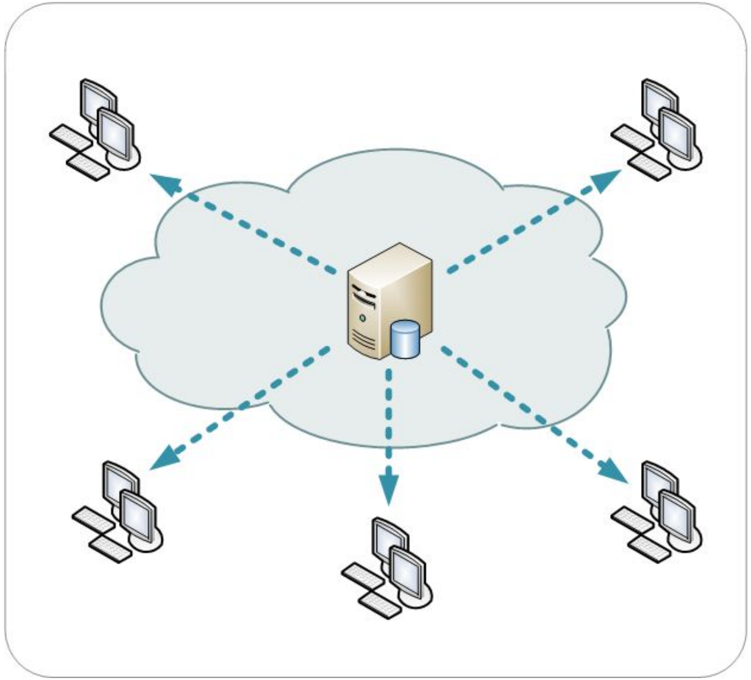
Motivation





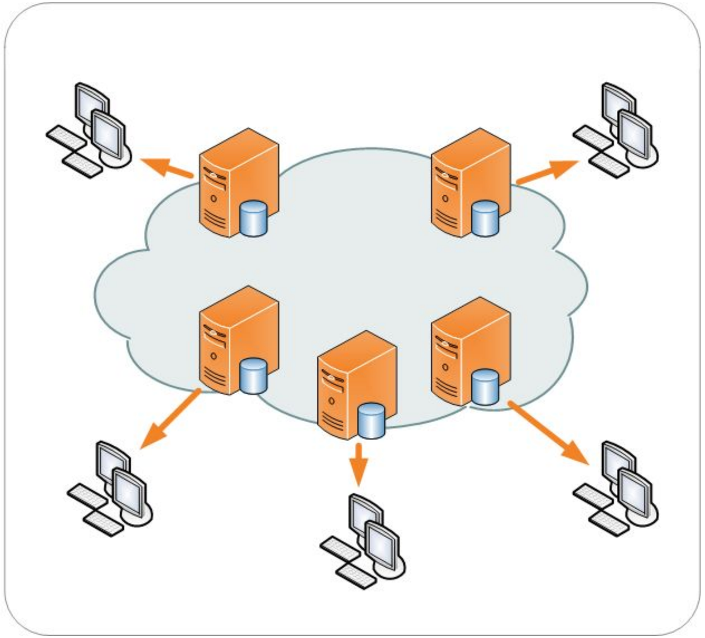
And while page load time is important, network latency continues to hold us back

Motivation



If our users are geographically distributed, how can we reduce latency?

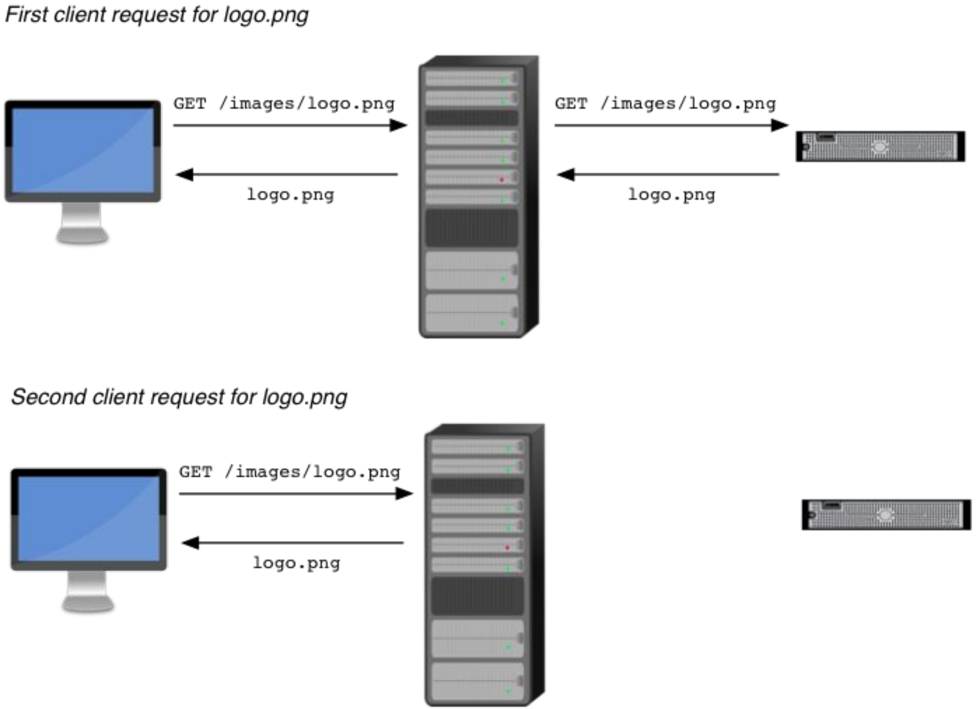
Motivation



If our users are geographically distributed, how can we reduce latency?

- Bring the content closer to the users

Implementation



Caching mechanism acts like an HTTP proxy.

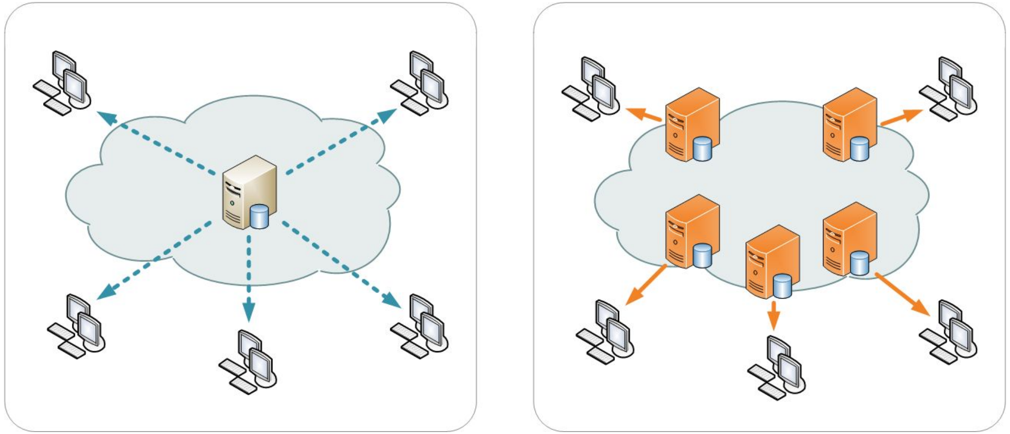
HTTP headers guide caching decisions

Implementation

Web Caching Fundamentals

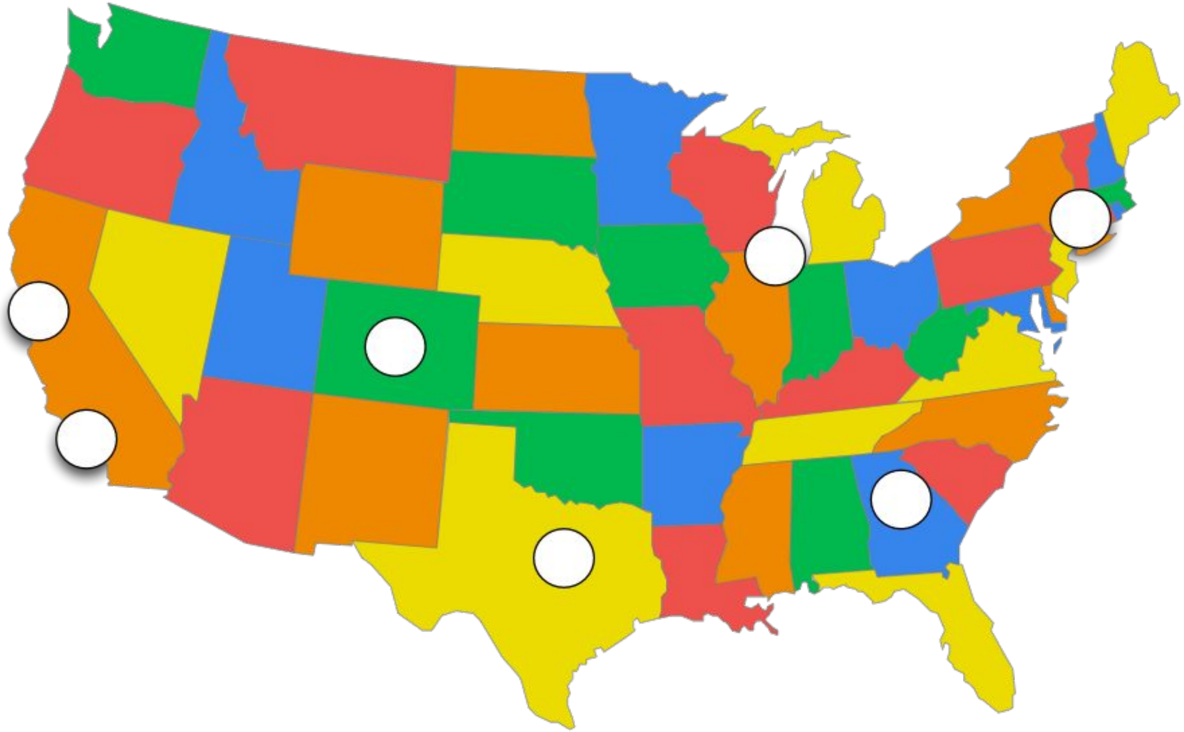
- Caches rely on a subset of popular content to maintain hit rate
  - When misses occur, cache interaction is pure overhead
- If content becomes popular very quickly, caches can introduce delay
  - Not in cache implies origin fetch, which ties up resources
  - Many requests for a missing object have to be coalesced to avoid transferring unnecessary load onto the origin
- HTTP Headers
  - Cache-Control, If-Modified-Since, etags

Motivation



So instead of all users contacting the origin app server for content, they will contact servers that are closer to them.

Implementation



Two big questions:

- Which endpoint ("Point of Presence", PoP) should we send them to?
- How should we send them there?

Implementation: How

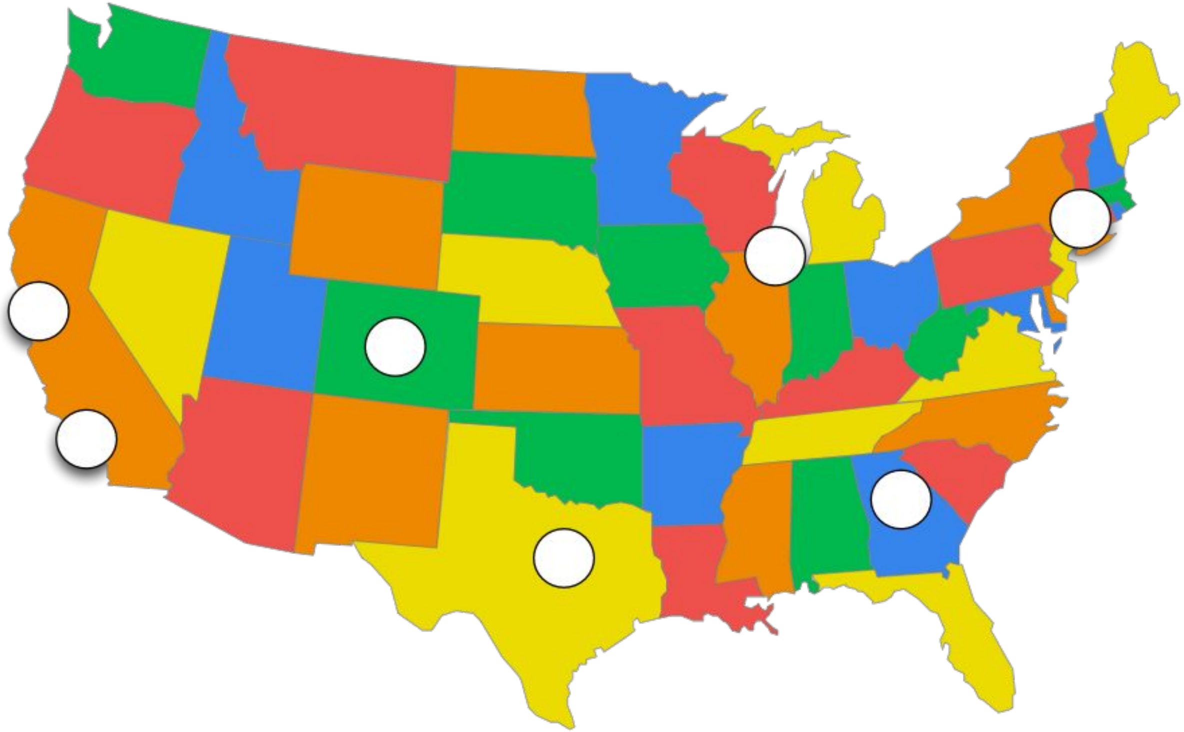
Modify hostname of URLs that you want to cache in order to route to CDN provider:

http://www.example.com/images/logo.png

Becomes:

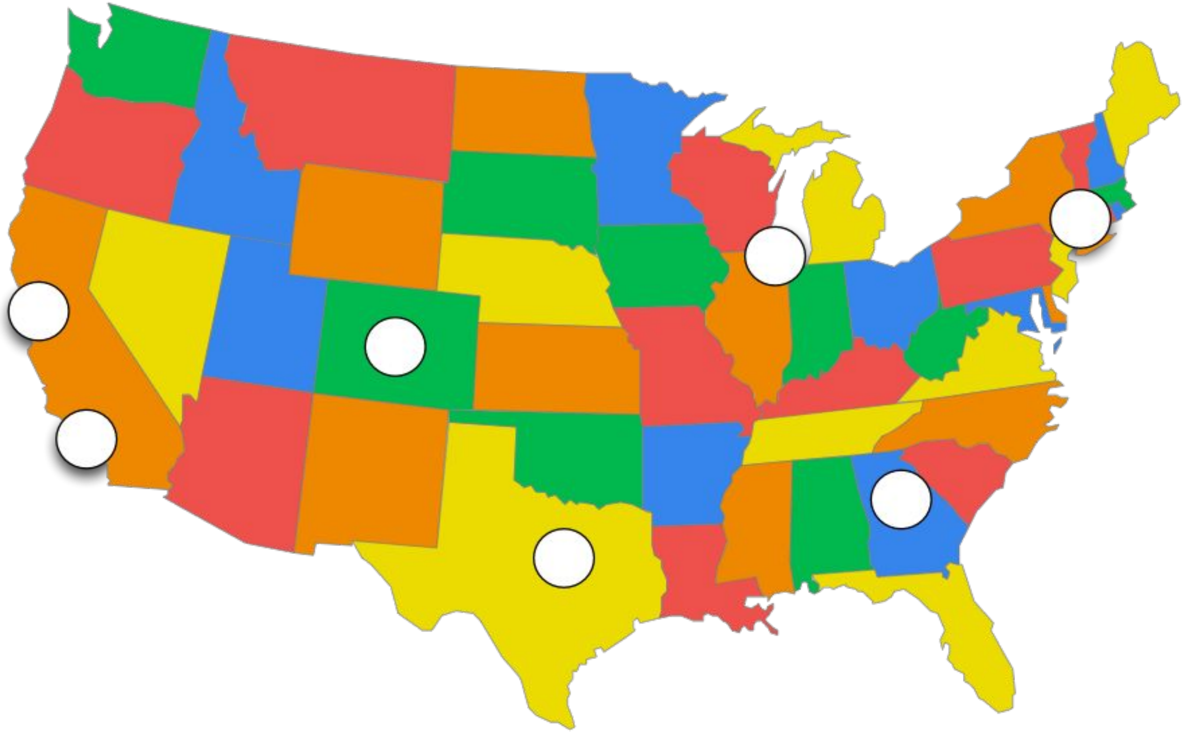
http://www.example.com.akamai.net/images/logo.png

Implementation: How



CDN provider configures DNS to route these requests to its "Points of Presence (POP)"

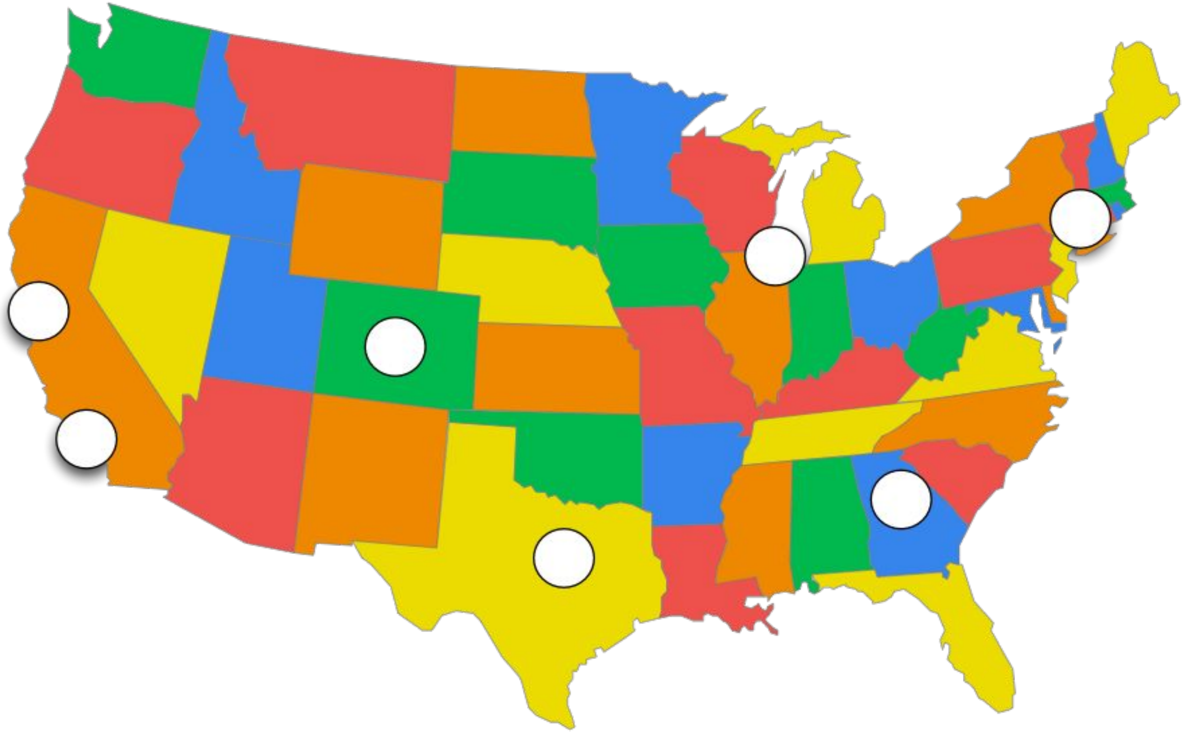
Implementation



www.example.com.akamai.net

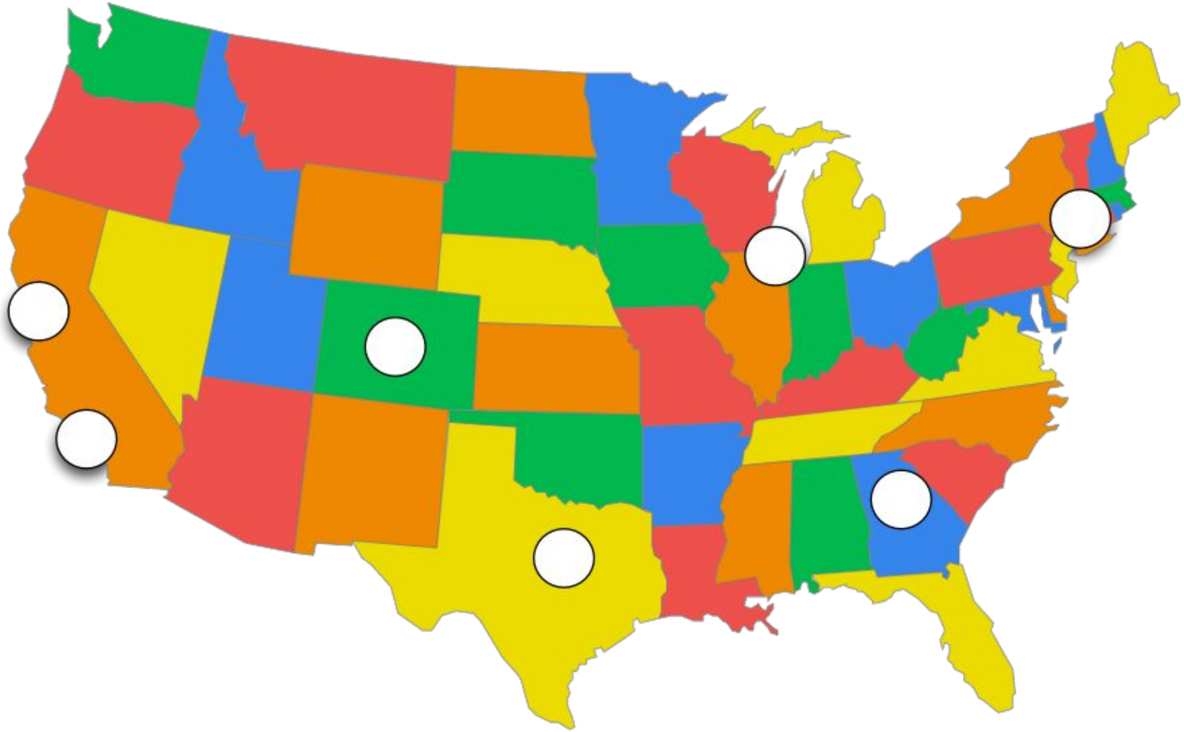
DNS provider can choose how to resolve this. Can Resolve to an IP address in Los Angeles for a west coast client, and New York for an east coast client.

Implementation



- DNS has time to live field. How should we configure this for use with a CDN?

Implementation



- Low TTL on DNS means you can be dynamic
  - Resolve load issues on pops or cache servers
  - Take cache servers in and out of service
  - Take POPs in and out of service

Implementation: Which

How could we choose which PoP to serve the content from?

- We know the IP address of the resolver, but not the client.

Implementation: Which

How do we choose which PoP to serve the content from?

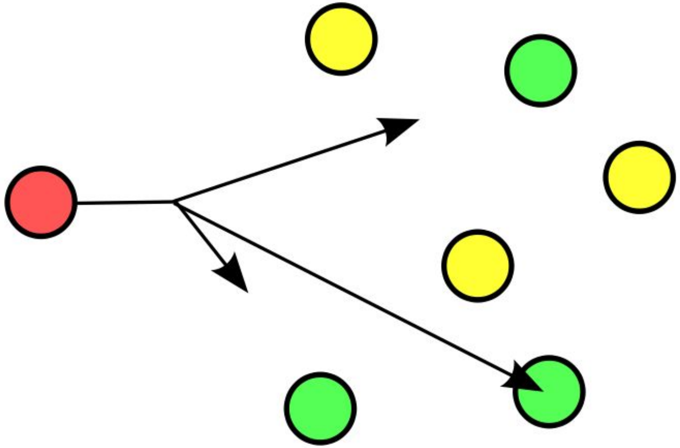
- Anycast
- Geolocation
- Client performance

Implementation: Which

Anycast IP allows multiple hosts on the internet to have the same IP address

BGP rules are set to route to nearest host

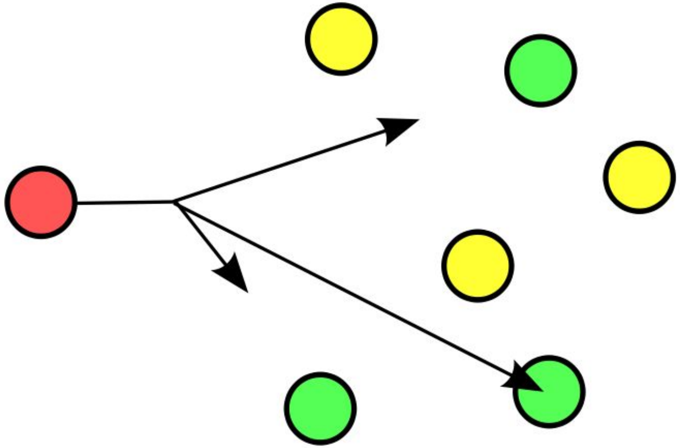
We can have all PoPs have the same IP, and use BGP and anycast to route to nearest PoP



Implementation: How

Challenges with anycast:

- A sequence of IP packets may end up at different hosts (BGP route changes)
    - Because HTTP is short-lived, not a big problem
- This will route to closest PoP, but closest might not be best
    - Packet loss, bandwidth could be issues



Implementation: Which

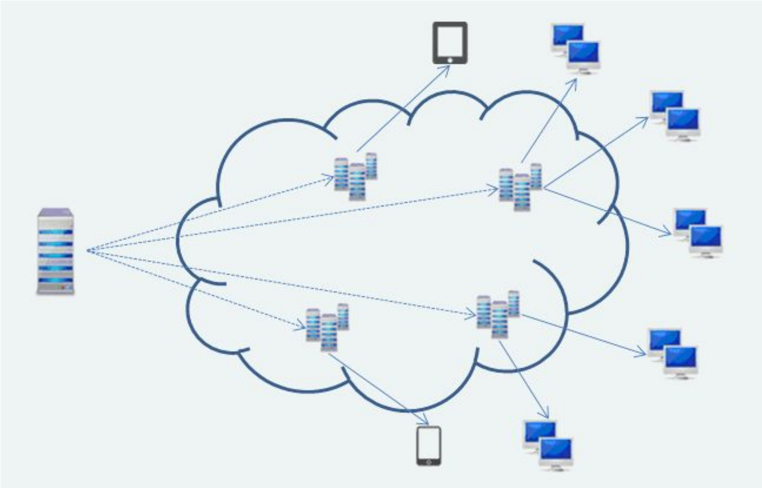
- Geolocate the DNS resolver
- Send to PoP that is geographically closest to the DNS resolver

- Works okay, but clients don't always use nearby DNS servers
    - Example?

Implementation: How

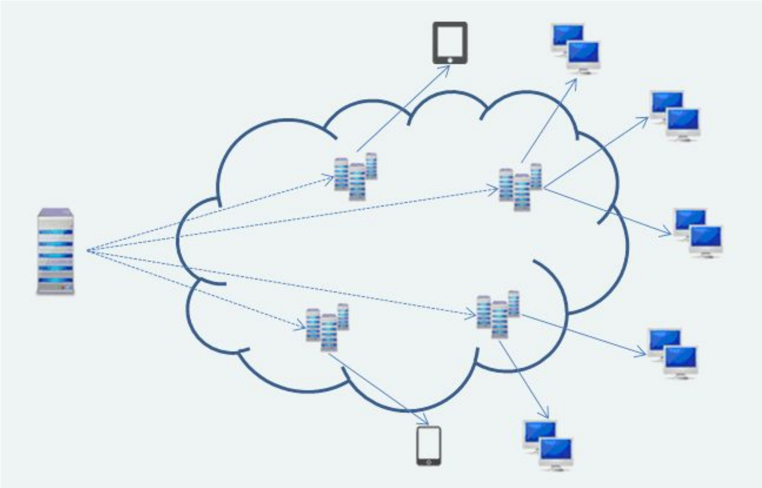
Client Performance Measurements

- Find a high-traffic web property that you can serve images and js from
- When clients arrive, request an image from a random PoP and record performance
- Maintain a mapping of dns resolver to average client performance
- Use this average client performance to make PoP decision at request time



Implementation: How

How to use measurements?



- Over time this becomes a huge dataset
- Want to remember history, but be adaptable
- Simple moving average

$$SMA = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

- Exponential moving average

$$S_1 = Y_1$$

$$\text{for } t > 1, \quad S_t = \alpha \cdot Y_t + (1 - \alpha) \cdot S_{t-1}$$

Implementation: How

Implementation

In summary

A CDN is a relatively simple way to speed up the serving of static assets in your web application.

Common providers:

- Akamai
- Amazon (CloudFront)
- CloudFlare

Course Conclusion

Let’s say...

...I want to find a home to live in.

...I am lost in a foreign city.

...I want to go on a date.

... what do I do?

Course Conclusion

Every day, billions of people use the same suite of technologies to solve these problems: internet services.

As these services get increasingly popular, they need to continue to function.

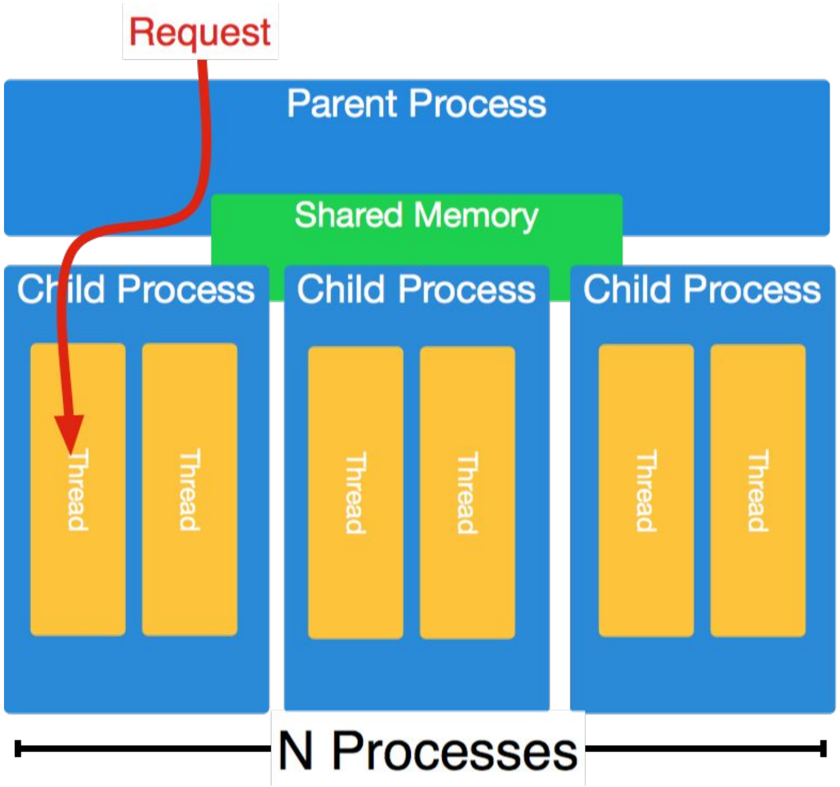
Scaling even relatively simple web applications (Twitter) can be very complex.

Course Conclusion

You've got a web application that is becoming increasingly popular and performance is degrading.

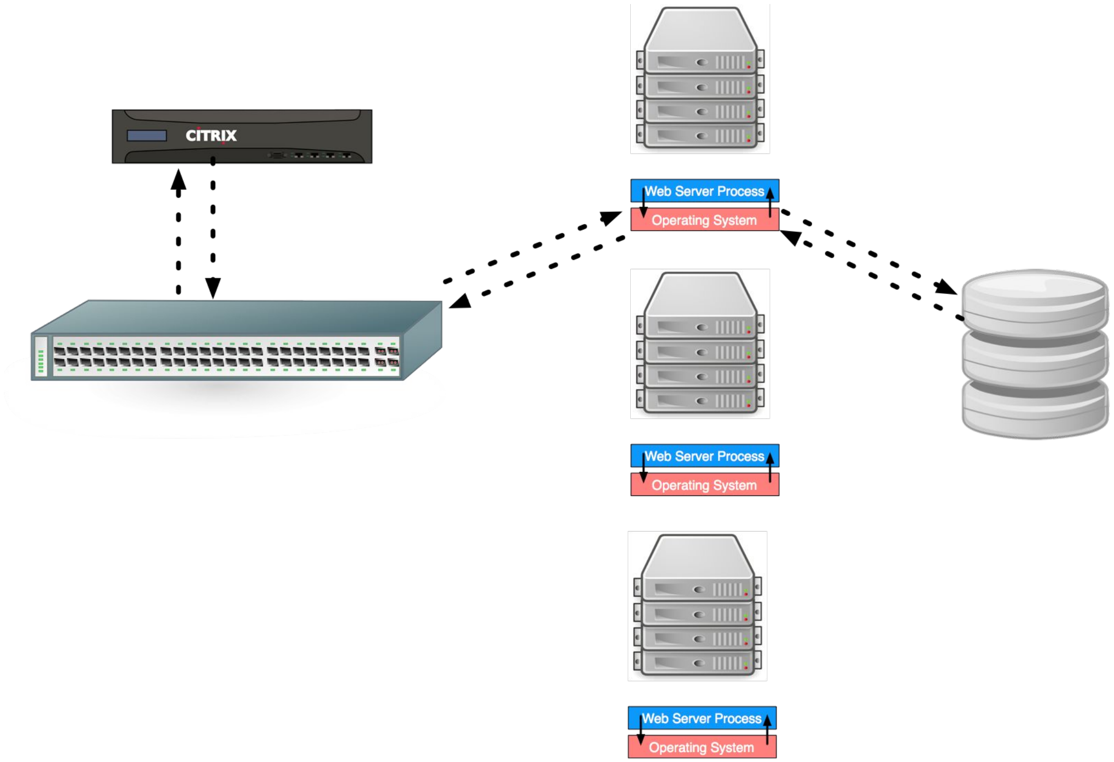
What do you do?

We’ve covered a lot of ground



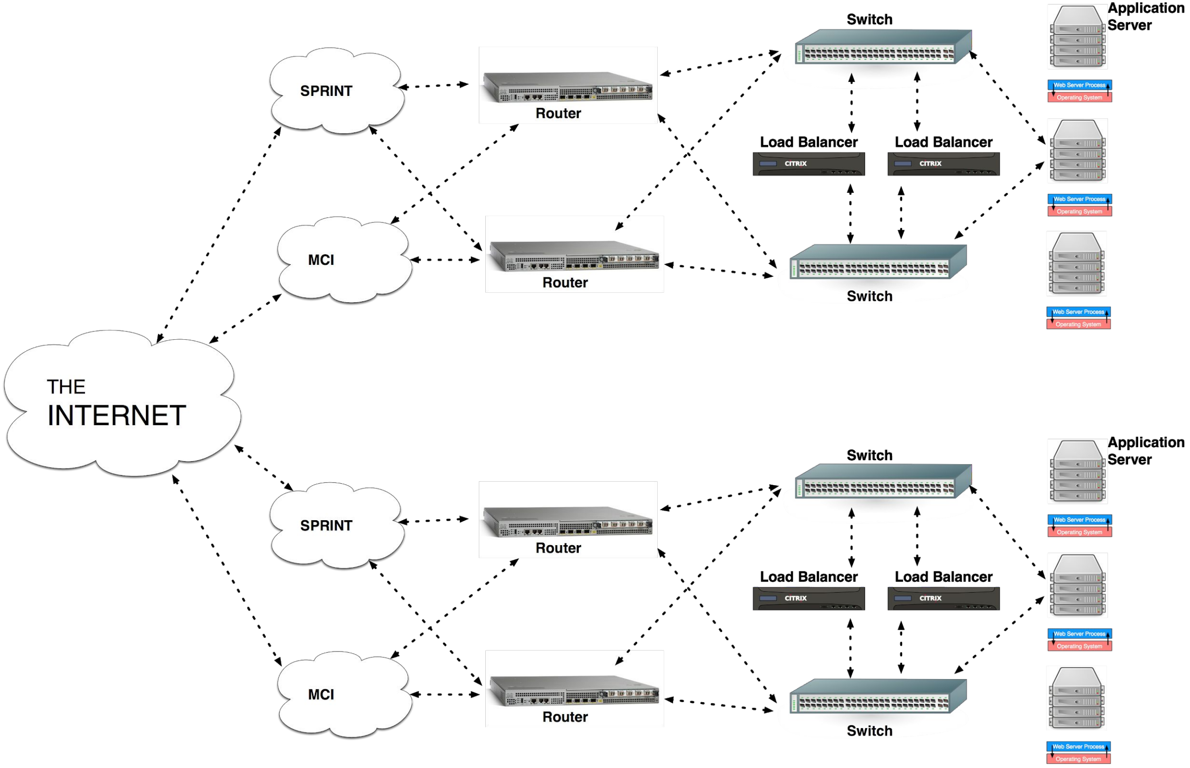
HTTP Servers, Application servers and their design

We’ve covered a lot of ground



The use of load balancing in achieving horizontal scaling

We've covered a lot of ground



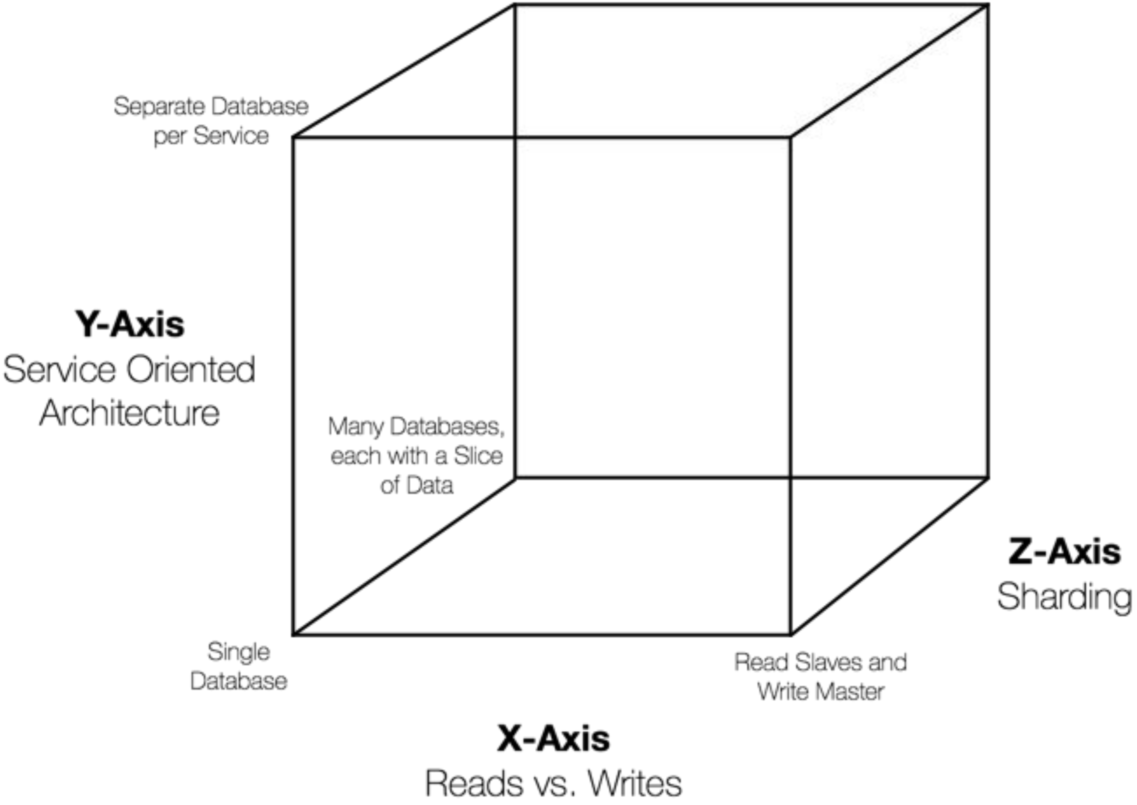
The architecture of a high availability, share nothing web application

We’ve covered a lot of ground

Name Path	Method	Status Text	Type	Size Content	Time Latency	Timeline 100 ms 150 ms 200 ms
www.cs290…	GET	304 Not Modified	text/h…	354 B 6.8 KB	18 ms 17 ms	
page.css /_stylesheets	GET	304 Not Modified	text/css	354 B 9.1 KB	19 ms 19 ms	
home.css /_stylesheets	GET	304 Not Modified	text/css	355 B 508 B	77 ms 76 ms	
jquery-1.11… /_javascript	GET	304 Not Modified	applic…	355 B 94.1 KB	76 ms 75 ms	
page.js /_javascript	GET	304 Not Modified	applic…	356 B 191 B	140 ms 139 ms	
octicons.css /_stylesheets	GET	304 Not Modified	text/css	354 B 12.0 KB	16 ms 16 ms	
normalize.css /_stylesheets	GET	304 Not Modified	text/css	355 B 8.8 KB	75 ms 74 ms	
grid.css /_stylesheets	GET	304 Not Modified	text/css	355 B 1.7 KB	72 ms 72 ms	
header.css /_stylesheets	GET	304 Not Modified	text/css	354 B 2.0 KB	30 ms 29 ms	
hero.css http://www.cs290.com/_stylesheets/hero.css	GET	304 …ied	text/css	354 B 1.9 KB	55 ms 55 ms	
ga.js www.googl…	GET	304 Not Modified	text/j…	170 B 40.0 KB	36 ms 35 ms	
ucsbcs-2x.… /images	GET	304 Not Modified	image…	355 B 36.4 KB	71 ms 71 ms	
Archimedes… upload.wiki…	GET	200 OK	image…	(from cache)	0 ms 0 ms	
screen-sho… /images	GET	200 OK	image…	(from cache)	0 ms 0 ms	
project_log…	GET	200	image…	(from cache)	0 ms	

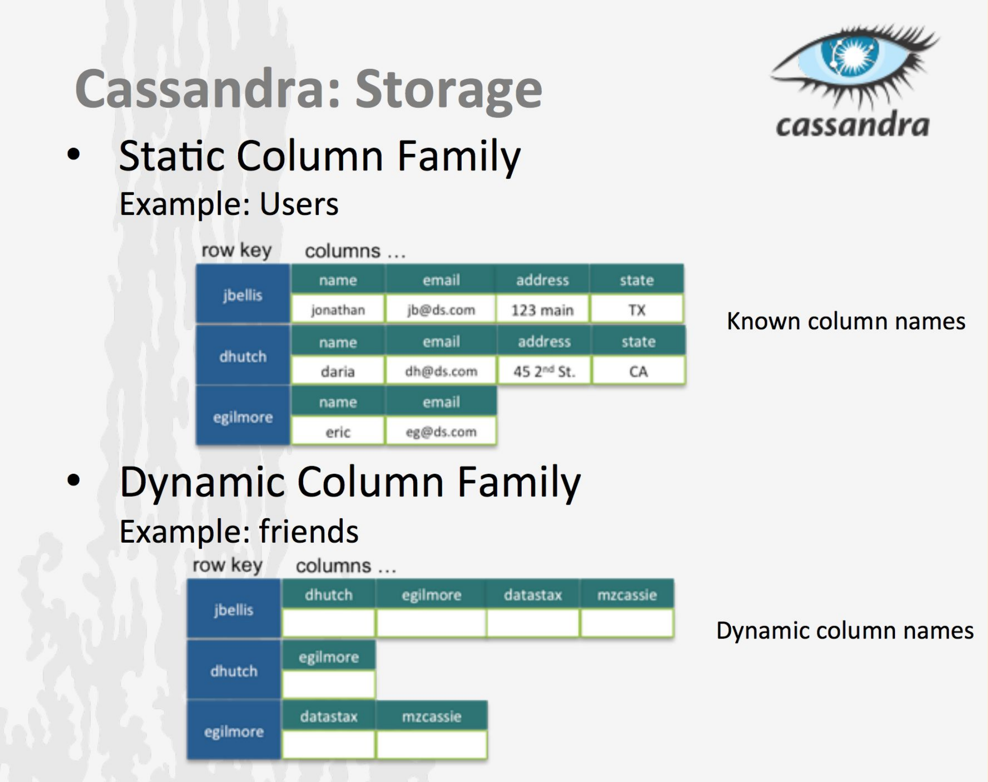
All about caching, both on the client and the server

We've covered a lot of ground



Relational databases and how to scale them

We've covered a lot of ground



Scaling beyond relational databases: NoSQL stores

We’ve covered a lot of ground

In addition to the question of scaling we have looked at:

- Basics of web security
- Client side MVC
- Asm.js and Emscripten
- HTTP 2.0 & QUIC
- GraphQL
- Serverless Computing
- How ML fits in to Scalable Internet Systems
- CDNs

We’ve covered a lot of ground

We’ve gained a lot of experience with modern web application technology

What I hope...

If you are headed for academia, I hope this window into industry helps your future research

If you are headed for industry, I hope the skills you've gained can help you get a job

If you ever want to start a company, I hope this course has given you all the tools you need to build a scalable internet service.

What I hope...

And I hope you've had fun!

- Please remember to fill out EIP on my.ucla.edu
- See you all on Thursday for presentations (here, usual time).
- Please send writeups by 12pm noon Thursday to andrew.mutz@cs.ucla.edu