

# Lecture 25 – Content Delivery Networks (CDN) and the Edge

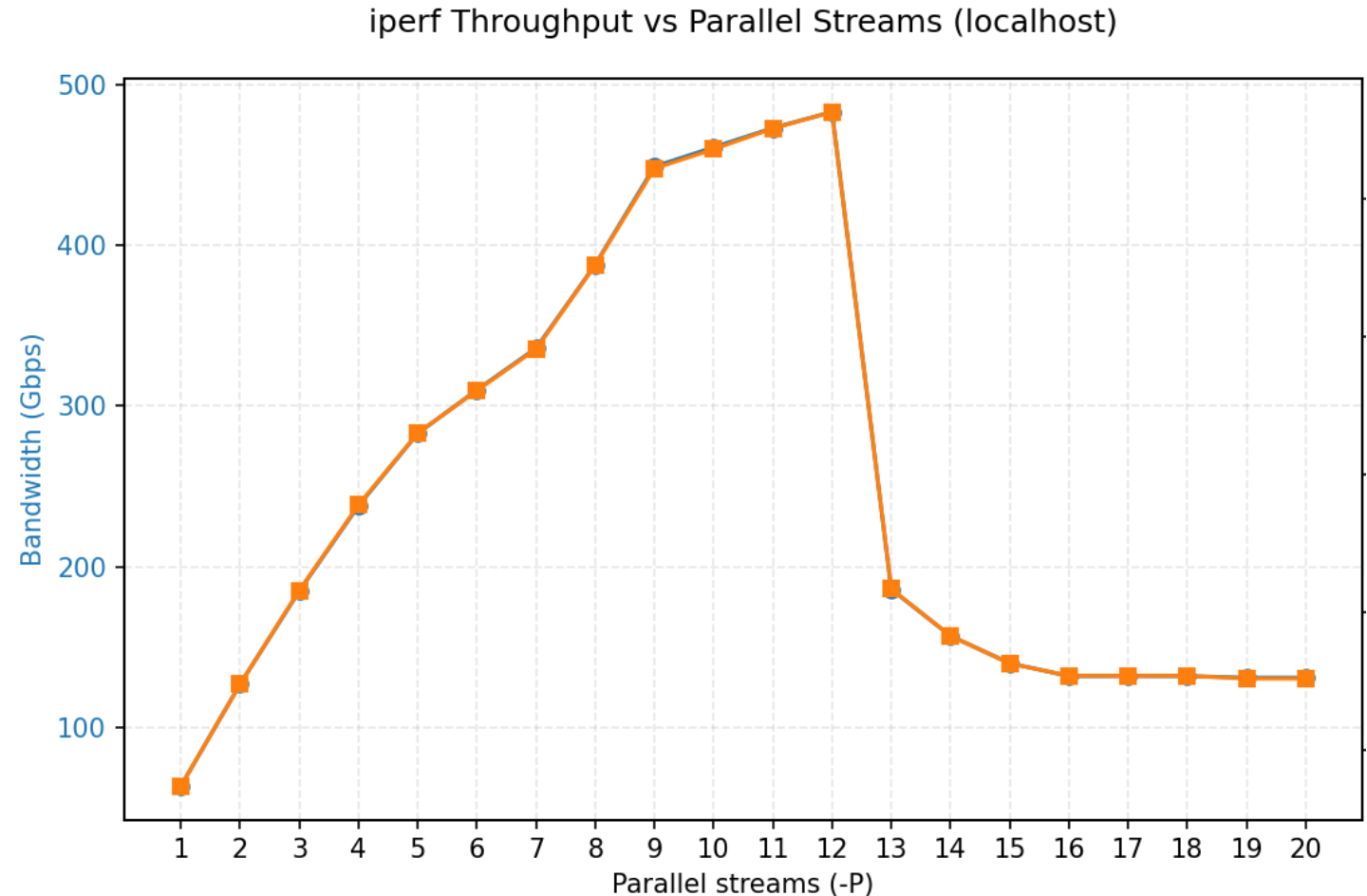
Lecturer: Venkat Arun

# Datacenters continued briefly: Datacenter Tax

- Datacenter networks are extremely fast
  - In the upper layers, 100-400 Gbit/s links are common
  - Individual servers can have links with capacity up to 100 Gbit/s
- If you try to open a TCP socket (say, using code in assignment 1), you will have a hard time sending at these speeds even over localhost!
  - **Live demo using iperf** – could only reach ~140 Gbit/s. It uses 100% of a CPU just doing TCP/IP! And this reduced since last year!
- This is called datacenter tax. It is estimated that 30% of CPU cycles are spent on just the network stack (TCP/IP, serialization/deserialization (e.g. JSON, protobufs), HTTP, and TLS)
- SmartNICs are a way to offload some of this workload to dedicated hardware, since you do not need the full power of a modern CPU

# Fun experiment running iperf to understand datacenter tax

- On my Mac I only got 15 Gbit/s when iperf started a TCP connection over localhost (that is, from my machine to itself). Last year, I got 140 Gbit/s. I do not know what changed
- On Linux, I got more. I tried starting multiple TCP streams and the *total* throughput looked like this. What do you think is going on?



# Datacenters are in a lot of places, but that still isn't enough



Amazon's locations seen in our class on datacenters



Akamai's locations. Source: Akamai

# What are CDNs/Edge computing?

- TL;DR - Have more, smaller datacenters
- Advantages:
  - Lower latency to client
  - Cheaper bandwidth to client, since it costs less to transmit data over shorter distances
    - Q: We don't see this effect in our internet bill. Why?
- Disadvantages:
  - Less data storage capacity per location. Desired data might not be available
  - Less compute and memory capacity per location. Might have to redirect to another nearby location
  - Harder to manage.
    - E.g. some claim that, in theory, all the text content of Twitter can be managed from a single large server with a super-fast in-memory database. This is much easier to manage

Discussion: When should you use the edge vs the datacenter?

# Example use case: delivering static content

- Websites have a lot of static files
  - Examples: Javascript code, static images like logos, CSS
  - The commonly used ones can be pre-distributed to the edge locations
  - HTML is loaded from a server in the main datacenter. HTML contains URLs for the other resources to be loaded. These can point to the CDN locations



Source:

<https://www.cloudflare.com/learning/cdn/what-is-a-cdn/>

# Live Demo: Course website

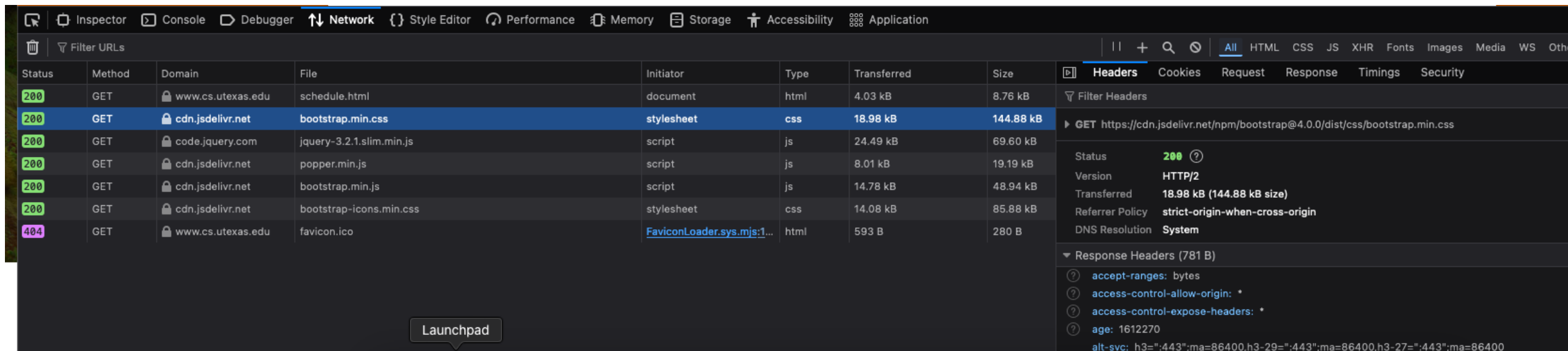


Image: “Network” tab of “Inspect element” in firefox for some the course website

- I used some standard libraries used by a lot of websites. Rather than serving those files to the user directly, I used a link to CDNs other people build and maintain.
- An additional advantage is that the web browser is likely to have these files already cached, in which case an entire request can be avoided



# How does the browser know which CDN replica to access?

- We want the browser to go to the nearest replica
- However, the URL looks like:  
<https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css>
- This contains no information about where to go
- Instead, people configure the DNS to give different IP addresses for the same domain depending on where the user is connected
- This is called **DNS load balancing**
  - It is also used to route clients to the closest origin server (i.e. large datacenter)

# Live demo: Do a DNS lookup from various locations

- I will use the command “nslookup cdn.jsdelivr.net 8.8.8.8” after connecting to various locations using a VPN
- The VPN makes all packets from laptop come from a different (configurable) location. This means that a DNS query made over a VPN will cause the Domain Name System (DNS) to return the result that it would have returned to a laptop that was actually in that other location
- Note: 8.8.8.8 is the IP address of a globally accessible DNS server hosted by google.com. If I had not specified it, my laptop would have use a cached value. Another option was to use the command: “nslookup -querytype=NS cdn.jsdelivr.net”

# Live demo: useful website - builtwith.com

- For example, this page lists the websites that use the Akamai CDN in the US

<https://trends.builtwith.com/websitelist/Akamai/United%20States>

- It also guesses what services other websites are using, revealing the true complexity of today's internet services

# Live demo: Walmart.com

The screenshot shows the Chrome DevTools Network tab. The left pane lists various requests, including GET requests to i5.walmartimages.com for various assets like BogleWeb\_subset-Bold.woff2, BogleWeb\_subset-Regular.woff2, and a large CSS file 74bd49d4cdcaba19.css. The right pane shows the details for the selected CSS file request, including its status (200), version (HTTP/2), and a list of response headers such as accept, accept-ranges, access-control-allow-headers, and server-timing.

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	www.walmart.com	/	document	html	73 kB	222.74 kB
	POST	analytics.www-release...	v2?OrgId=o-1W0T4G-na1&UserId=0da7369a-c5fe-438d-99bc-bd5f	fs.js:4 (beacon)			
	POST	www.walmart.com	beacon	init.js:2 (beacon)			
	POST	b.www.walmart.com	rum.gif	rum.js:1 (beacon)			
	POST	drfdiscv.walmart.com	AKh0uiOifc_gC1o?ae523a4230272dda5=a1GcVILI2YyhGmd-m8vj0ht	pPcuBcdJNgjrzNRC:83...			
	GET	www.walmart.com	680eb494	script	CSP		
200	GET	i5.walmartimages.com	BogleWeb_subset-Bold.woff2	font	font-woff2	18.76 kB	17.93 kB
200	GET	i5.walmartimages.com	BogleWeb_subset-Regular.woff2	font	font-woff2	18.32 kB	17.49 kB
200	GET	i5.walmartimages.com	beacon.js?bd=b.www.walmart.com&bh=beacon.lighttest.walmart.com	script	js	2.36 kB	2.50 kB
200	GET	i5.walmartimages.com	k2-_417cf614-4372-4909-a282-ba26f79ae115.v1.jpg?odnHeight=4	img	jpeg	36.40 kB	35.40 kB
200	GET	www.walmart.com	init.js	script	js	112.17 kB	242.95 kB
200	GET	i5.walmartimages.com	74bd49d4cdcaba19.css	stylesheet	css	62.33 kB	295.90 kB
200	GET	i5.walmartimages.com	ui_item-queue_banner-queue-banner-59bdf1304fe9fa16.js	script	js	6.92 kB	16.78 kB
200	GET	i5.walmartimages.com	ui_event-timer_queue-timer.28817945a0128759.js	script	js	5.21 kB	13.03 kB
200	GET	i5.walmartimages.com	node_modules_pnpm_walmart+use-safeiframe@0.4.0_react@18.2	script	js	5.11 kB	13.81 kB
200	GET	i5.walmartimages.com	node_modules_pnpm_react-device-detect@2.2.3_react-dom@18.2	script	js	11.55 kB	24.59 kB
200	GET	i5.walmartimages.com	ads_core_hooks_use-is-tracking-enabled-43d2722f6dd7df.js	script	js	10.18 kB	34.87 kB
200	GET	i5.walmartimages.com	intl-ads_sponsored-products-tracking_tracking-constants-04d7fe0	script	js	7.42 kB	20.54 kB
200	GET	i5.walmartimages.com	intl-ads_video_vast-video-player-ffa502c2341d1723.js	script	js	14.84 kB	37.93 kB
200	GET	i5.walmartimages.com	ads_core_utils_get-viewport-677581cca2d57df2.js	script	js	9.88 kB	27.58 kB
200	GET	i5.walmartimages.com	intl-ads_ui_btf-1ed20addf2a83a60.js	script	js	29.10 kB	98.64 kB
200	GET	i5.walmartimages.com	ads_core_hooks_use-load-ivt.401bf59551ea00a4.js	script	js	5.32 kB	10.73 kB
200	GET	i5.walmartimages.com	ads_data-access_hooks_use-get-ad-query.8abd73ae657256d4.js	script	js	7.64 kB	28.18 kB
200	GET	i5.walmartimages.com	intl-ads_ad-routing_ini...	script	js	8.36 kB	20.80 kB
200	GET	i5.walmartimages.com	tempo-phased-module_adaptive-card_adaptive-card-wrapper.d4f5	script	js	14.73 kB	45.28 kB

Details for GET https://i5.walmartimages.com/dfw/63fd9f59-b71d/dd5c0c53-8691-4b68-a818-5ebfa024b784/v2/en-US/\_next/static/css/74bd49d4cdcaba19.css:

- Status: 200
- Version: HTTP/2
- Transferred: 62.33 kB (295.90 kB size)
- Referrer Policy: strict-origin-when-cross-origin
- Request Priority: Highest
- DNS Resolution: System

Response Headers (956 B):

- accept: text/css
- accept-ranges: bytes
- access-control-allow-headers: \*
- access-control-allow-methods: GET, PUT, POST, OPTIONS
- access-control-allow-origin: \*
- access-control-expose-headers: \*
- cache-control: public, max-age=32912, s-maxage=86400
- content-encoding: gzip
- content-length: 61372
- content-type: text/css
- date: Thu, 21 Nov 2024 17:51:38 GMT
- expires: Fri, 22 Nov 2024 03:00:10 GMT
- server-timing: cdn-cache; desc=HIT
- server-timing: edge; dur=1
- server-timing: product; desc="slb", host; desc="c07fc60617f1", dc; desc="644e3f5", fetch-ms; dur=234, req-proc-ms; dur=0, resp-proc-ms; dur=0, product; desc="edge", host; desc="09ec5da0b8fb", dc; desc="7f8e6dc", fetch-ms; dur=264, req-proc-ms; dur=80, resp-proc-ms; dur=2
- server-timing: Ak-ipv; desc="ipv4"
- server-timing: Ak-cont-type; desc="text/css"
- server-timing: ak\_p; desc="1732211498945\_388970742\_921997245\_19\_910\_14\_0\_41", dur=1

- The complex URL tells the browser which domain to download an image from
- Based on nslookup, it seems to use akamai and Fastly as its CDN depending on the location

# Instapoll question

- Suppose, I think that DNS load balancing is too complex.
  - That is, I do not want to make the DNS return different IPs depending on the client's location
- Instead, I want the origin server to look at the client's IP address and send a URL that will work without DNS load balancing
- I have two options in mind. Which one works?
  - <edge-location-id>.mydomain.com/<image-id>
  - mydomain.com/<edge-location-id>/<image-id>
- Why?

# Instapoll answer

- <edge-location-id>.mydomain.com/<image-id>
- Why? In this case, the DNS receives <edge-location-id>.mydomain.com, which means even if DNS returns the same IP address independent of the client's location, the origin server can send different <edge-location-id>s to different clients based on their location
- In the other case, the DNS only receives “mydomain.com”, to which it will be forced to return the same IP address to all the requests.

# Other uses for the edge

- Edge storage:
  - In addition to what we discussed, a common “static file” served from CDNs is stored video (E.g. youtube, Netflix). They are large and frequently accessed(thus, easy to cache)
- Edge compute is an emerging trend (we won't discuss this in detail in this course) :
  - Cloud gaming puts a lot of the compute on a remote server. Latency requirements force the compute to be close to the user
  - RANs in 5G networks