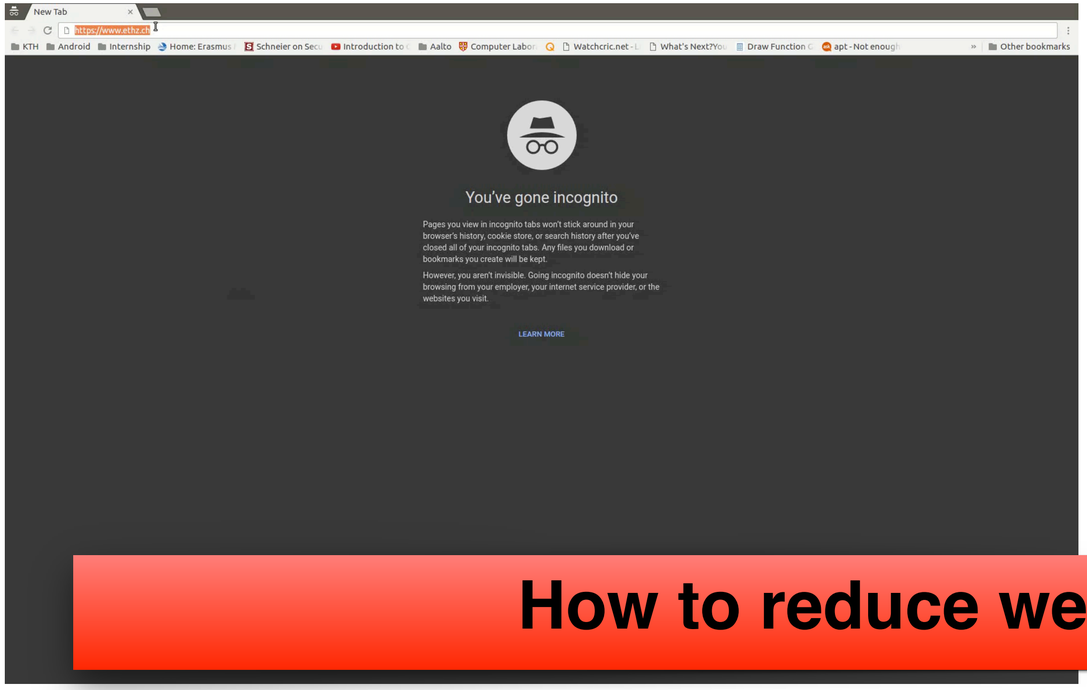
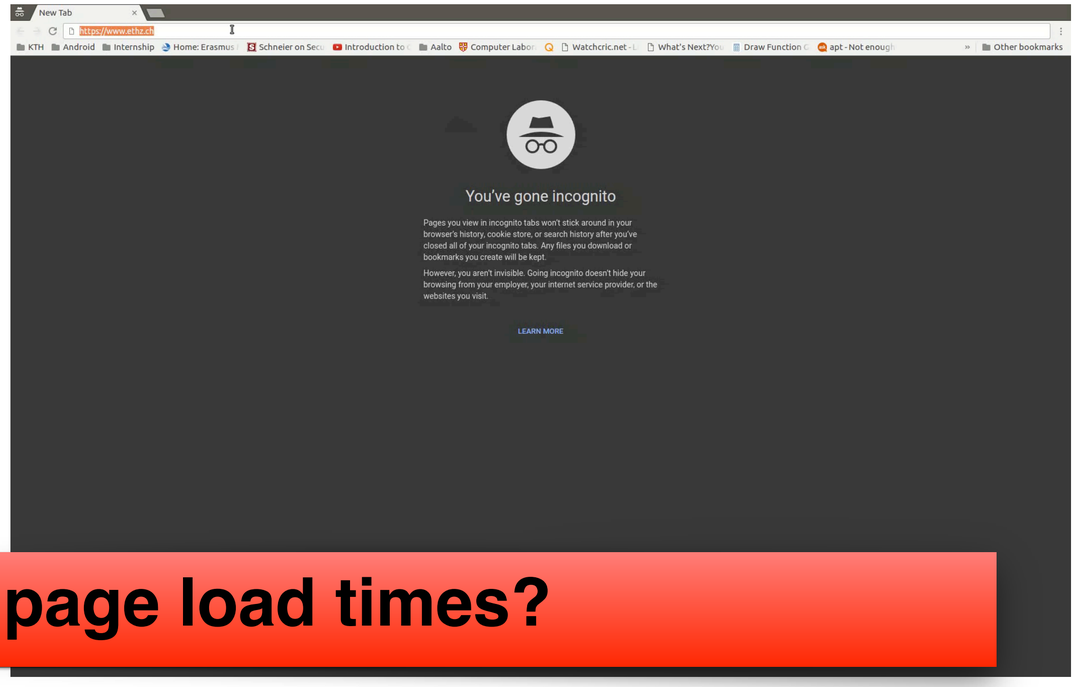
A Cloud-based Content Gathering Network

Debopam Bhattacherjee, Muhammad Tirmazi, Ankit Singla

ETH zürich





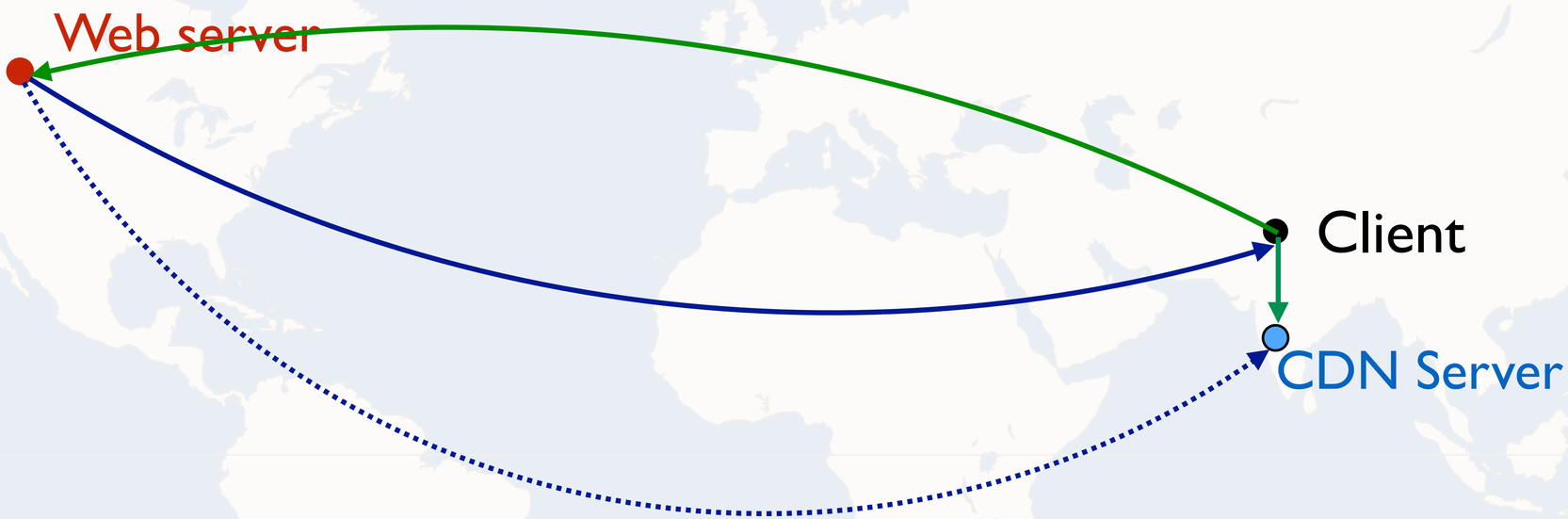
How to reduce web page load times?

: Few 100 millisecs of increase in latency => significant reduction in no. of searches per user

bing

amazon

Content Delivery Networks



Content Delivery Networks

- Predominant model for Web page delivery today
- Globally distributed infrastructure
  - Push content closer to the users
- Key Players: Akamai, Limelight, and Cloudflare
- Problems
  - Poor last mile latency
  - Needs content-origin for dynamic content
  - Infrastructure (Akamai: 233,000+ servers in 1600+ ISP networks/ IXPs)
  - Associated cost

Protocol Enhancements

SPDY: 2012

- Multiplexing and concurrency
- Compression of Headers
- Server Push
- Stream dependencies

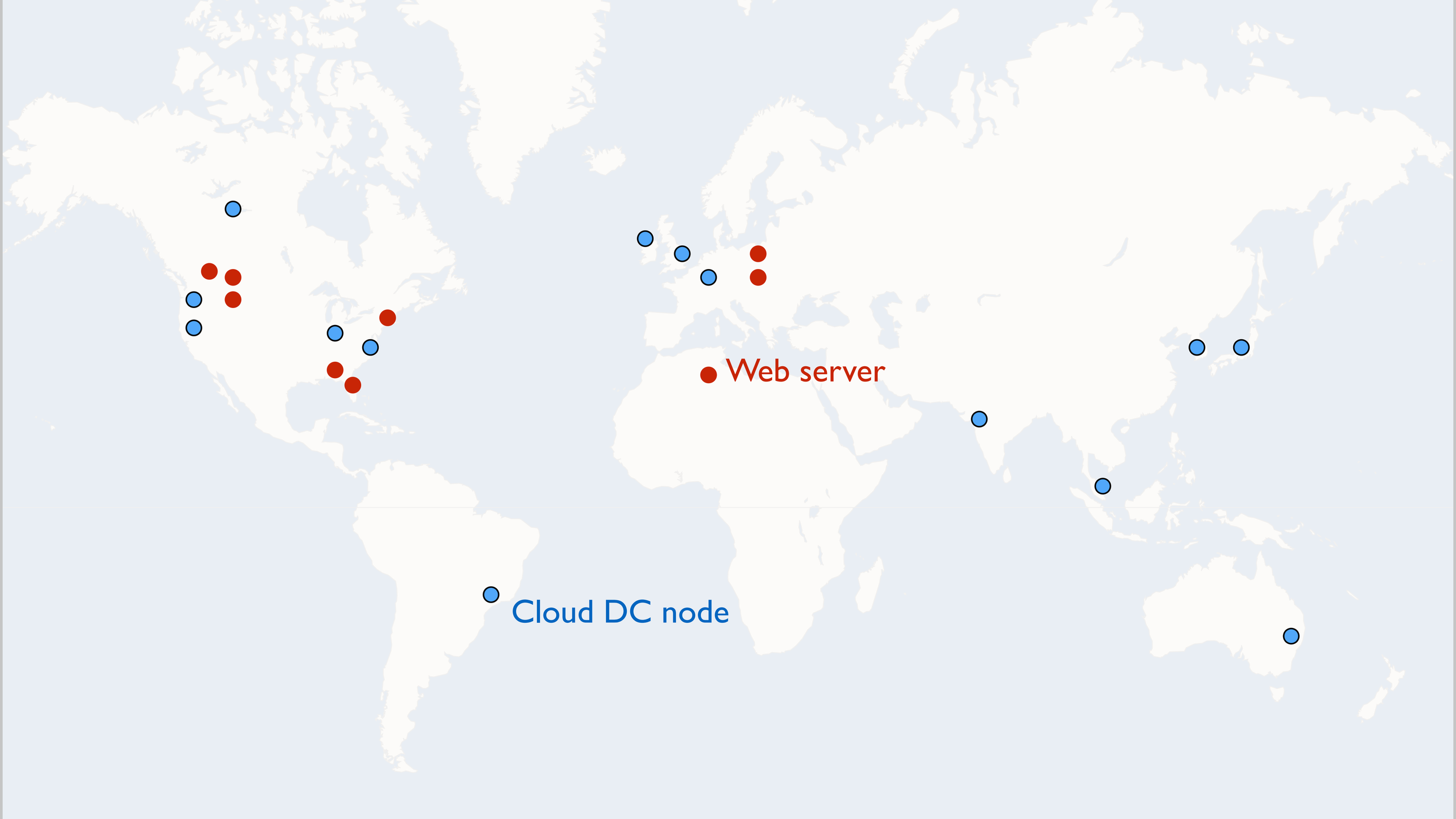
Low adoption rates

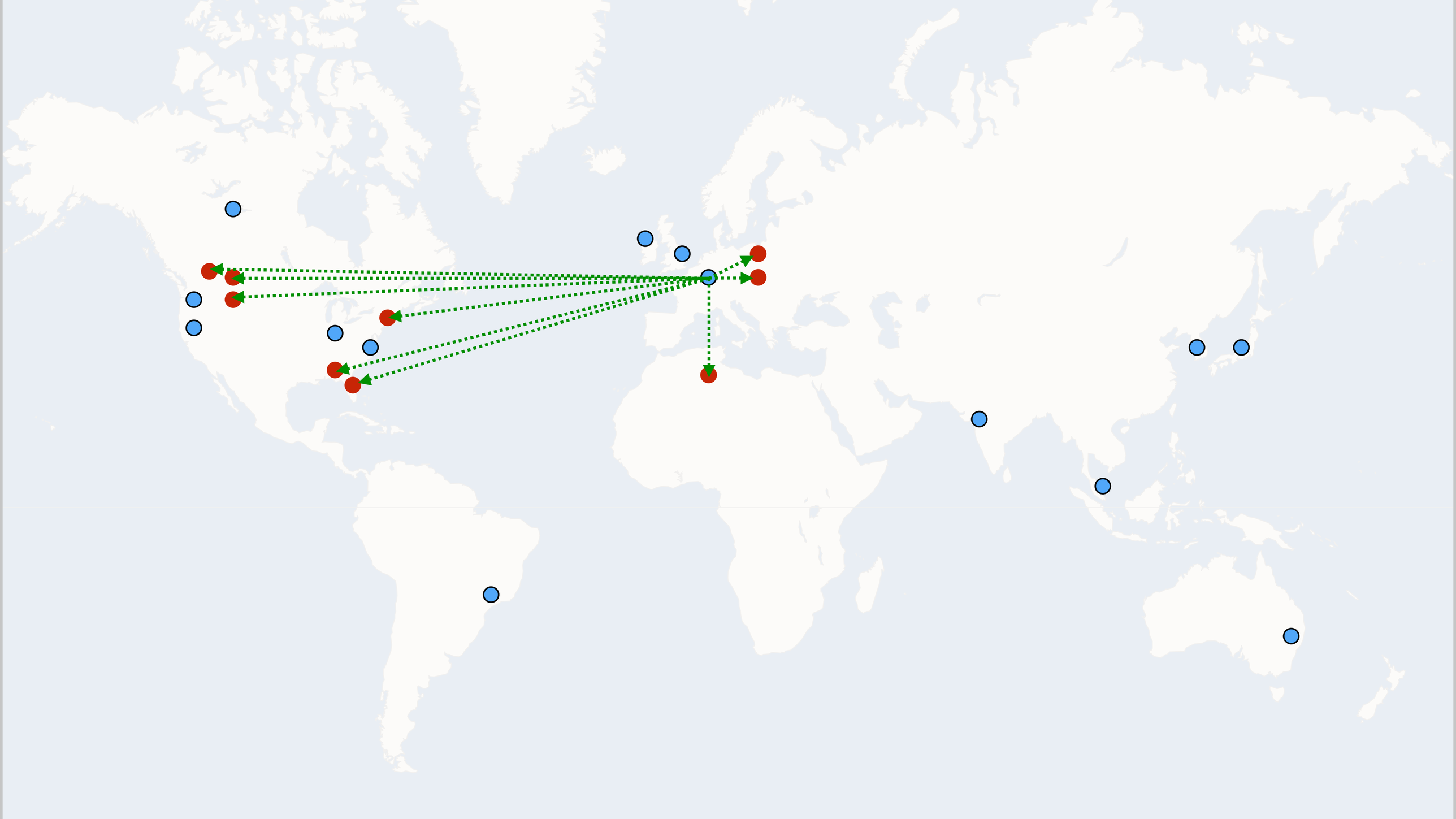
under 1% across Web servers (2015)

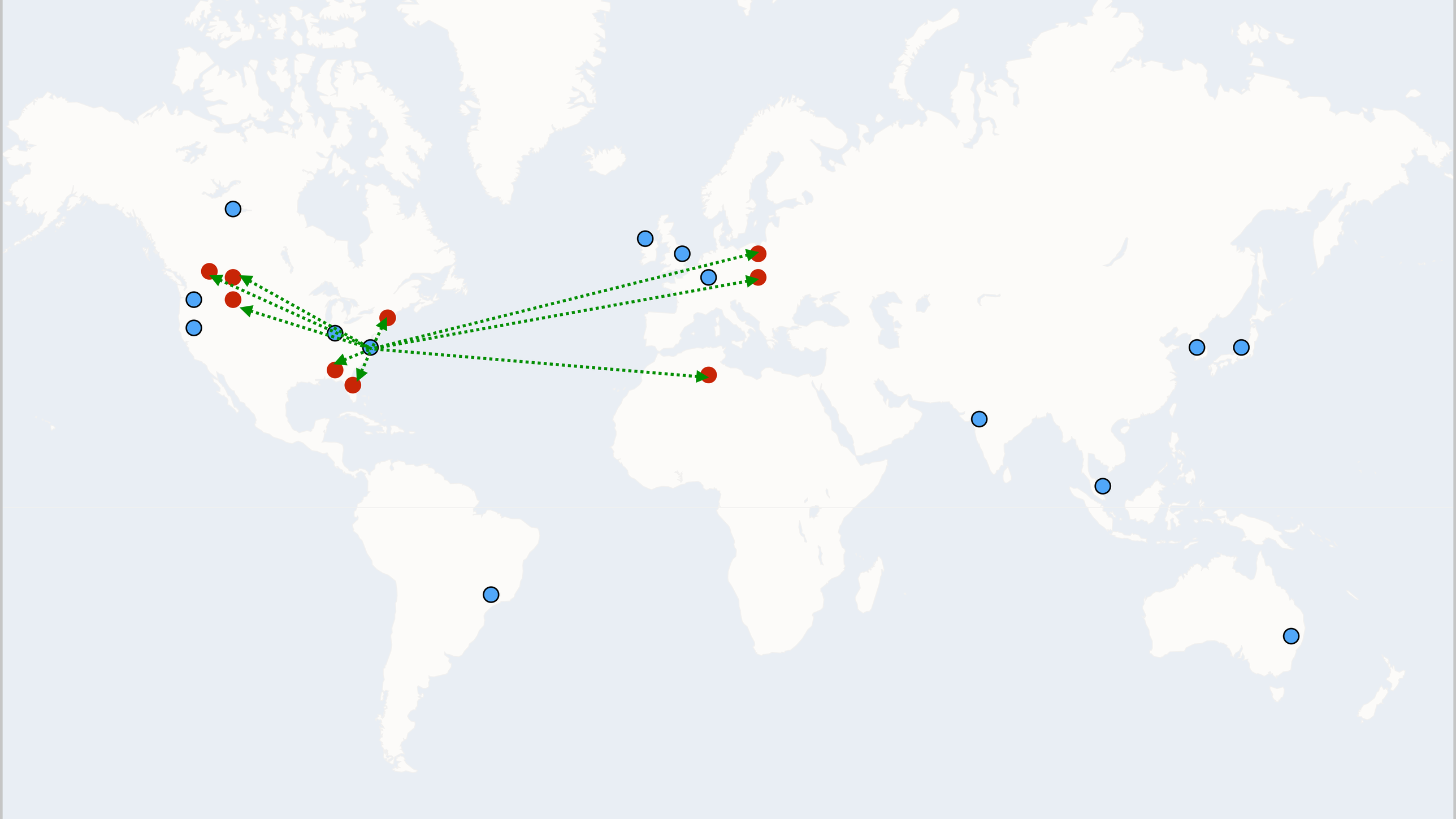
WebP: 2010

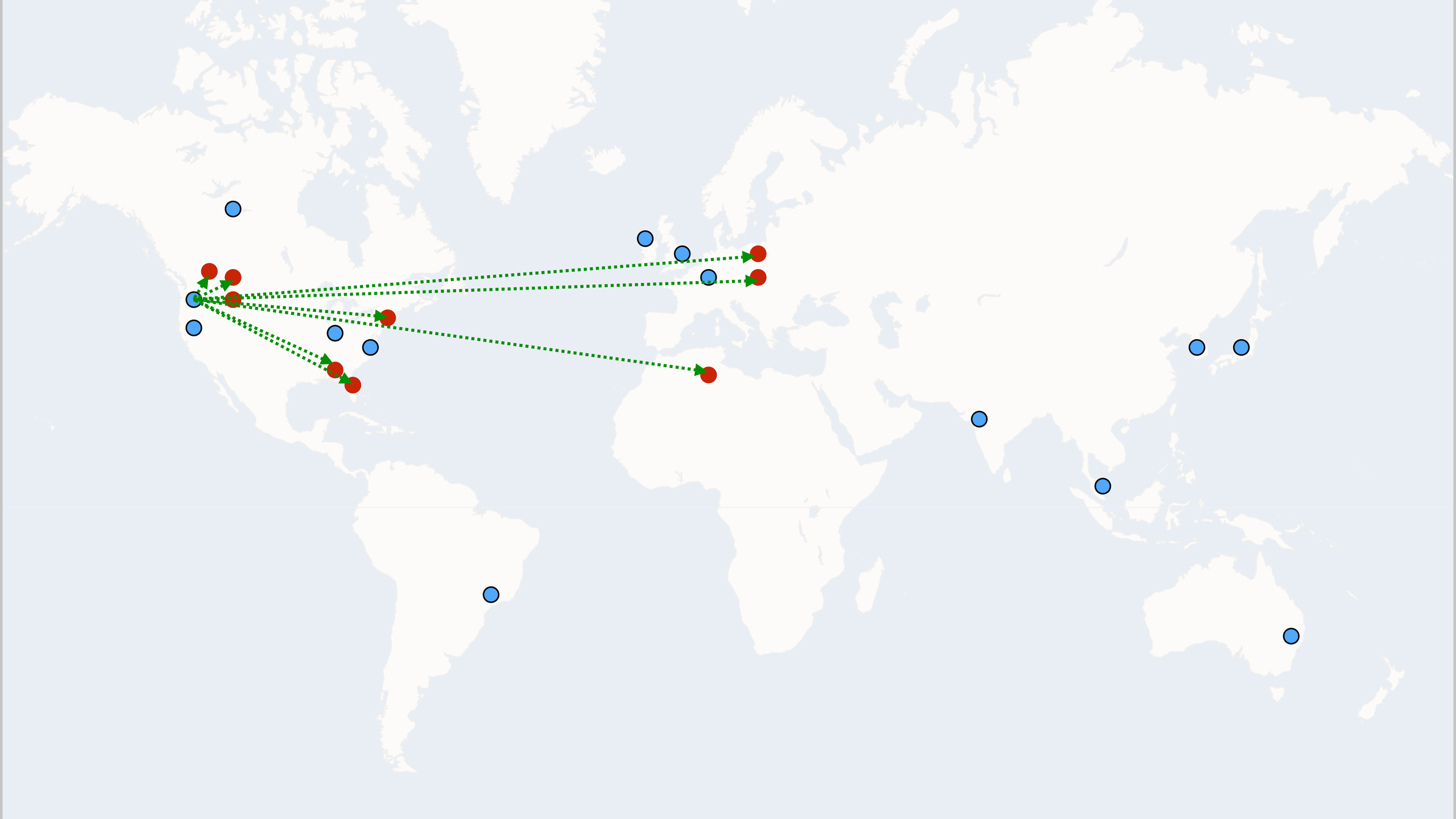
- Supports both lossy and lossless compression
- PNG/JPEG to WebP -> 30% reduction in file size
- Animated GIF to lossy WebP -> 64% reduction in file size

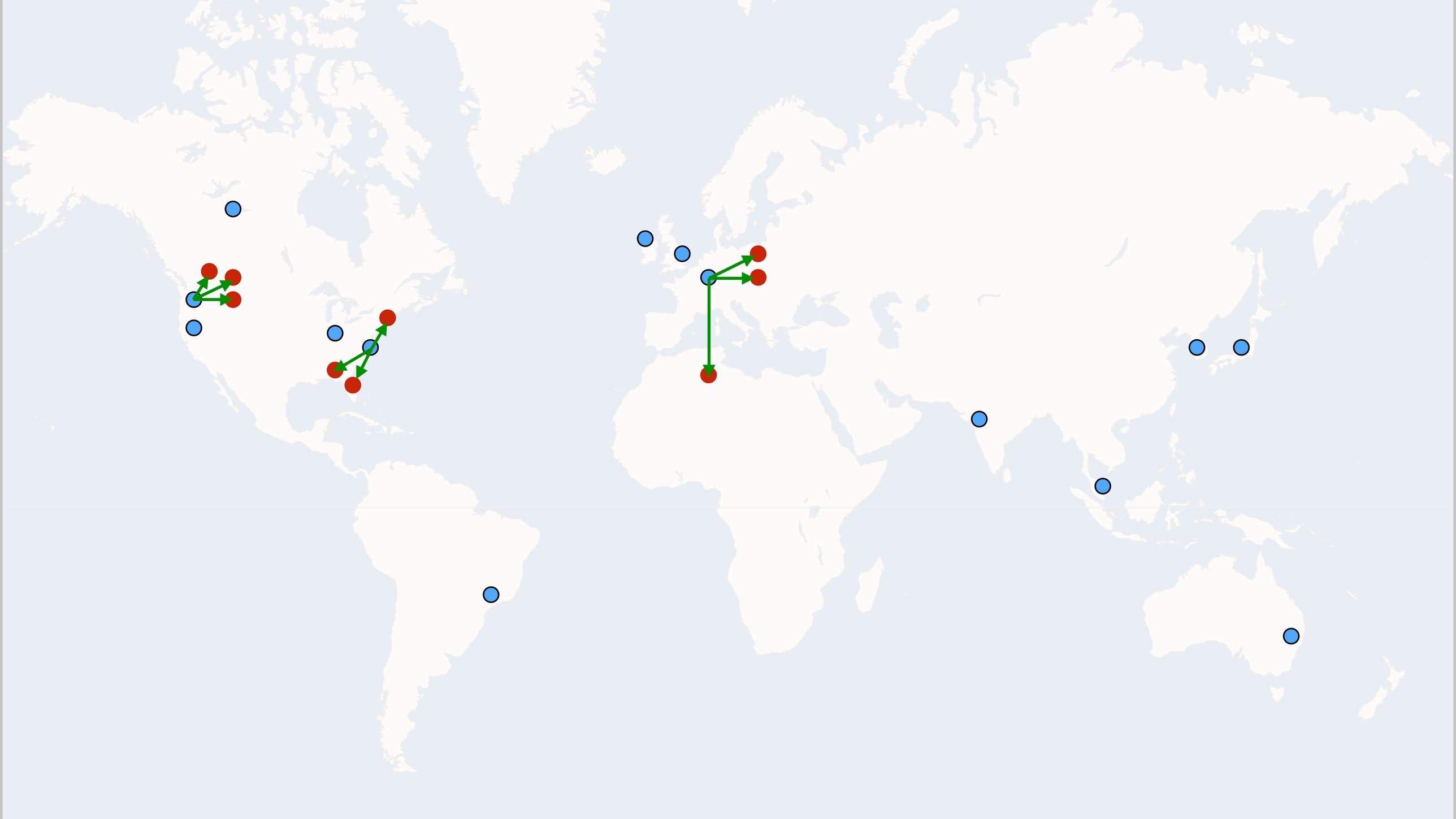

Web server


Web server
Cloud DC node

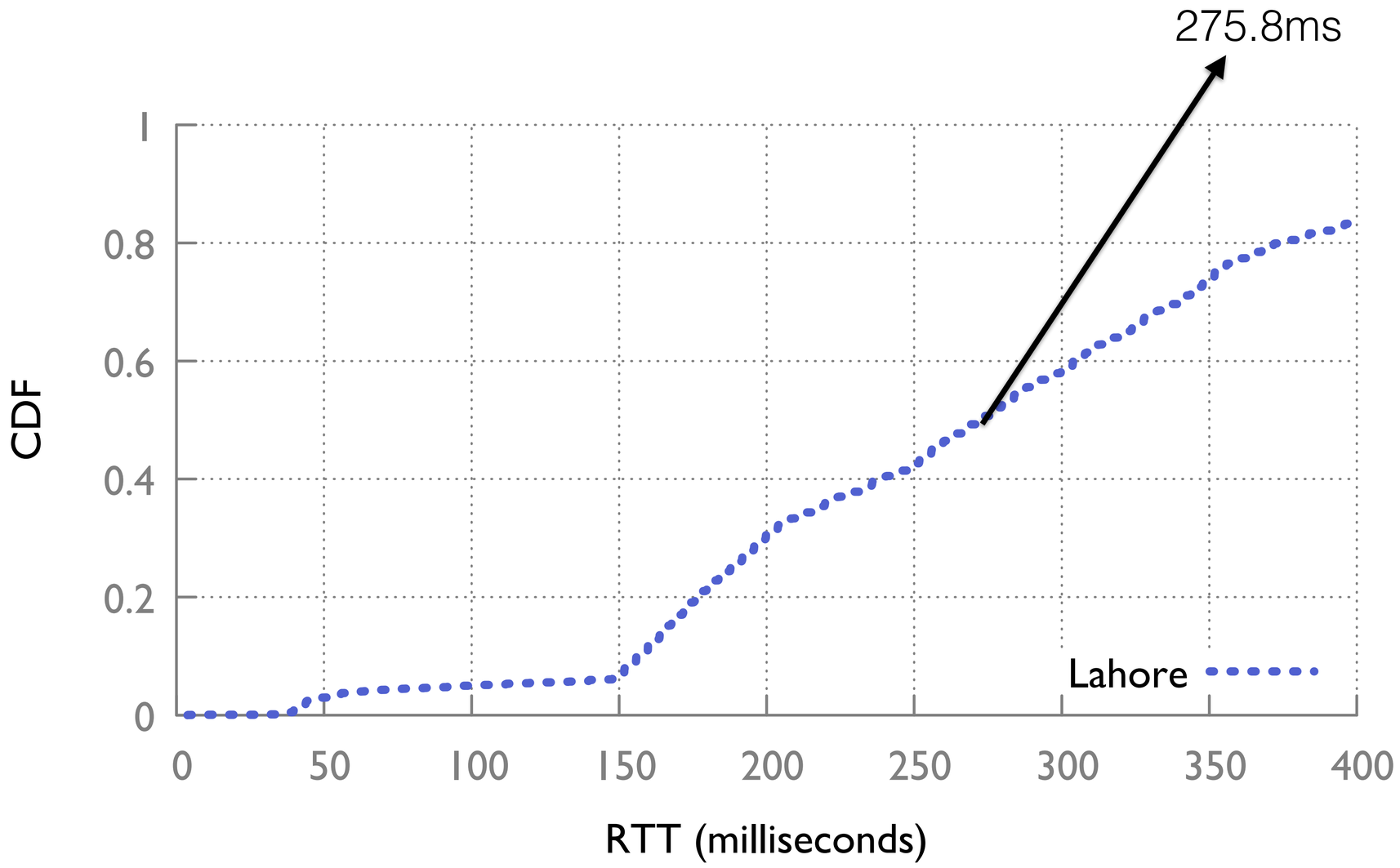




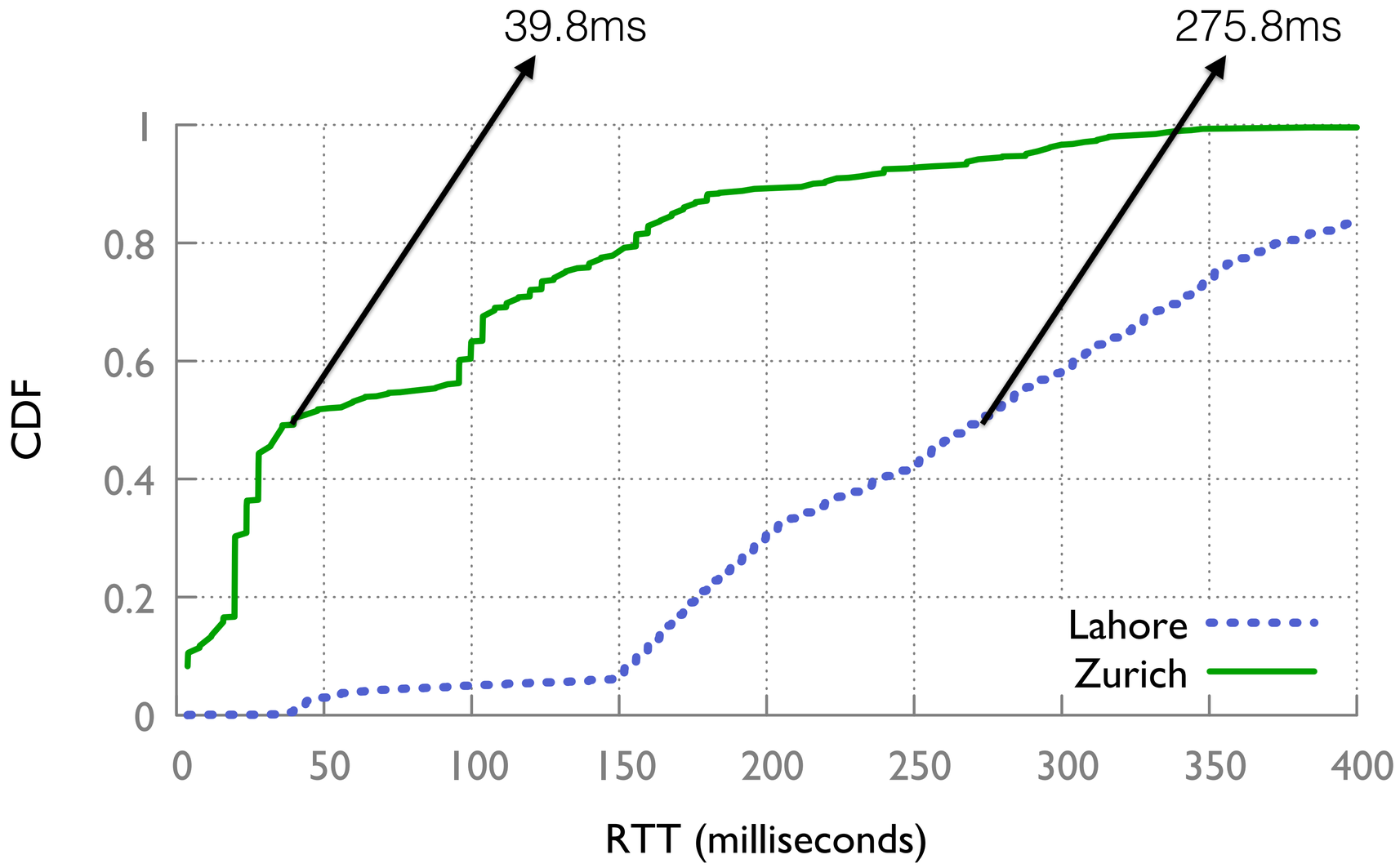




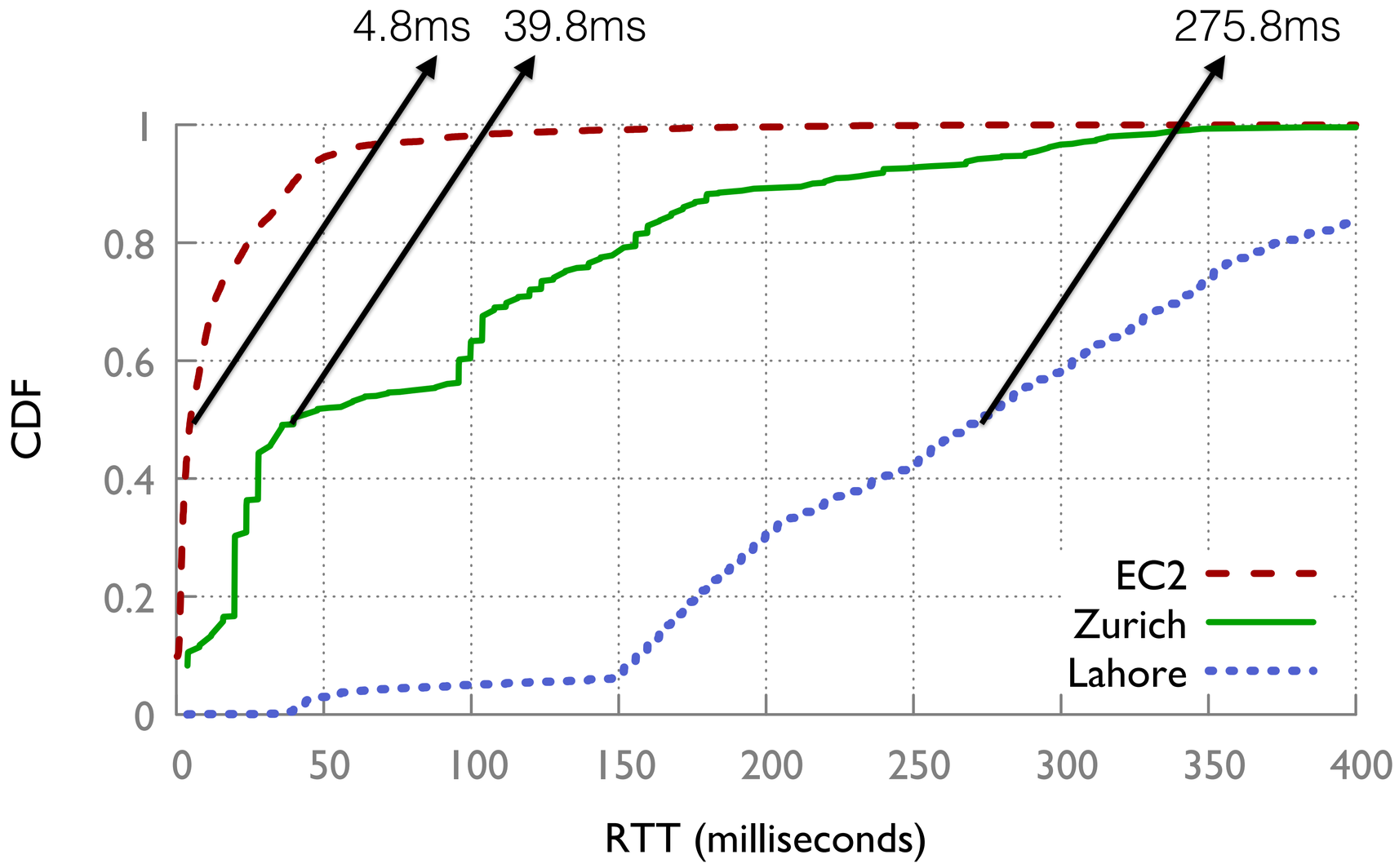
Consolidation of 10,000 popular Web servers



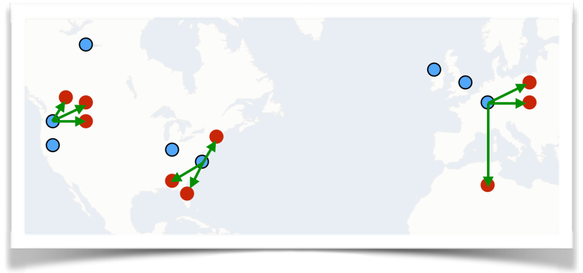
Consolidation of 10,000 popular Web servers

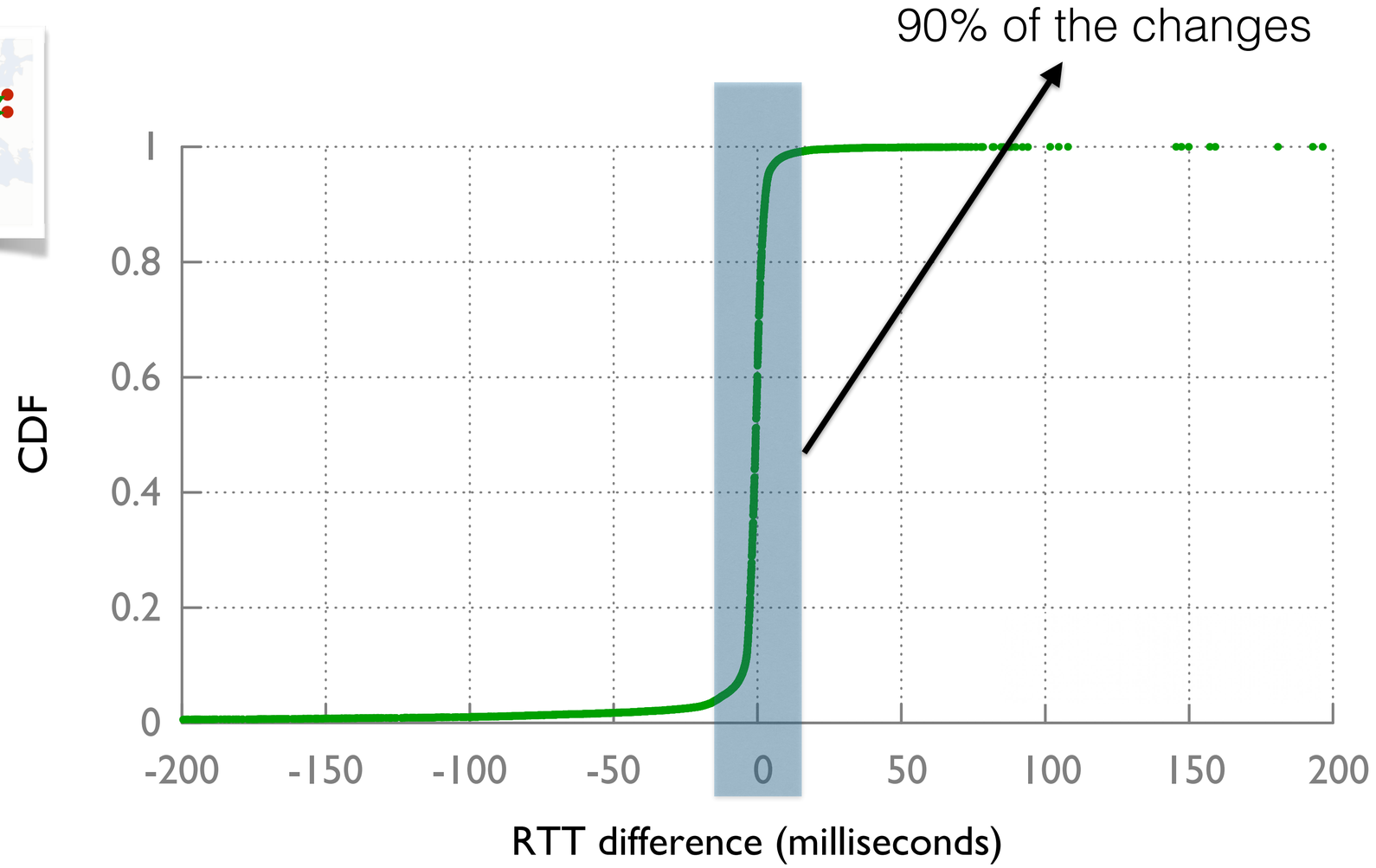


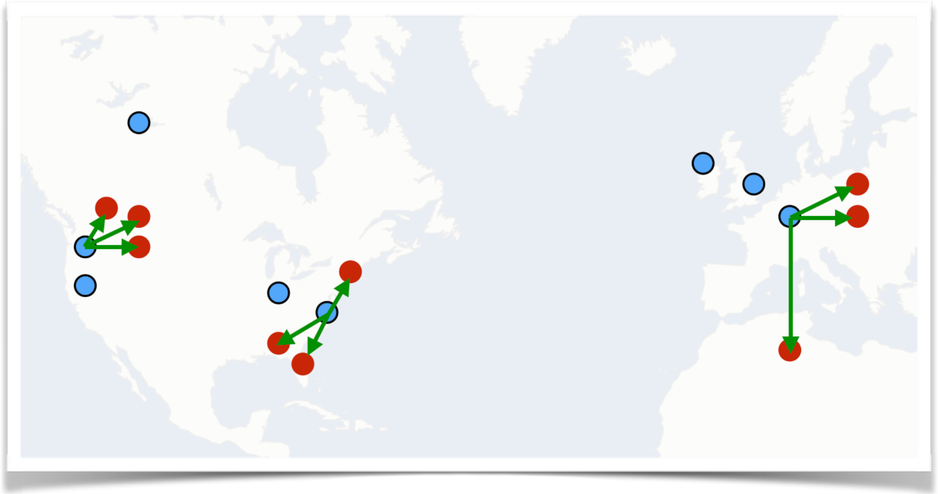
Consolidation of 10,000 popular Web servers



Stability: Mappings do not change often



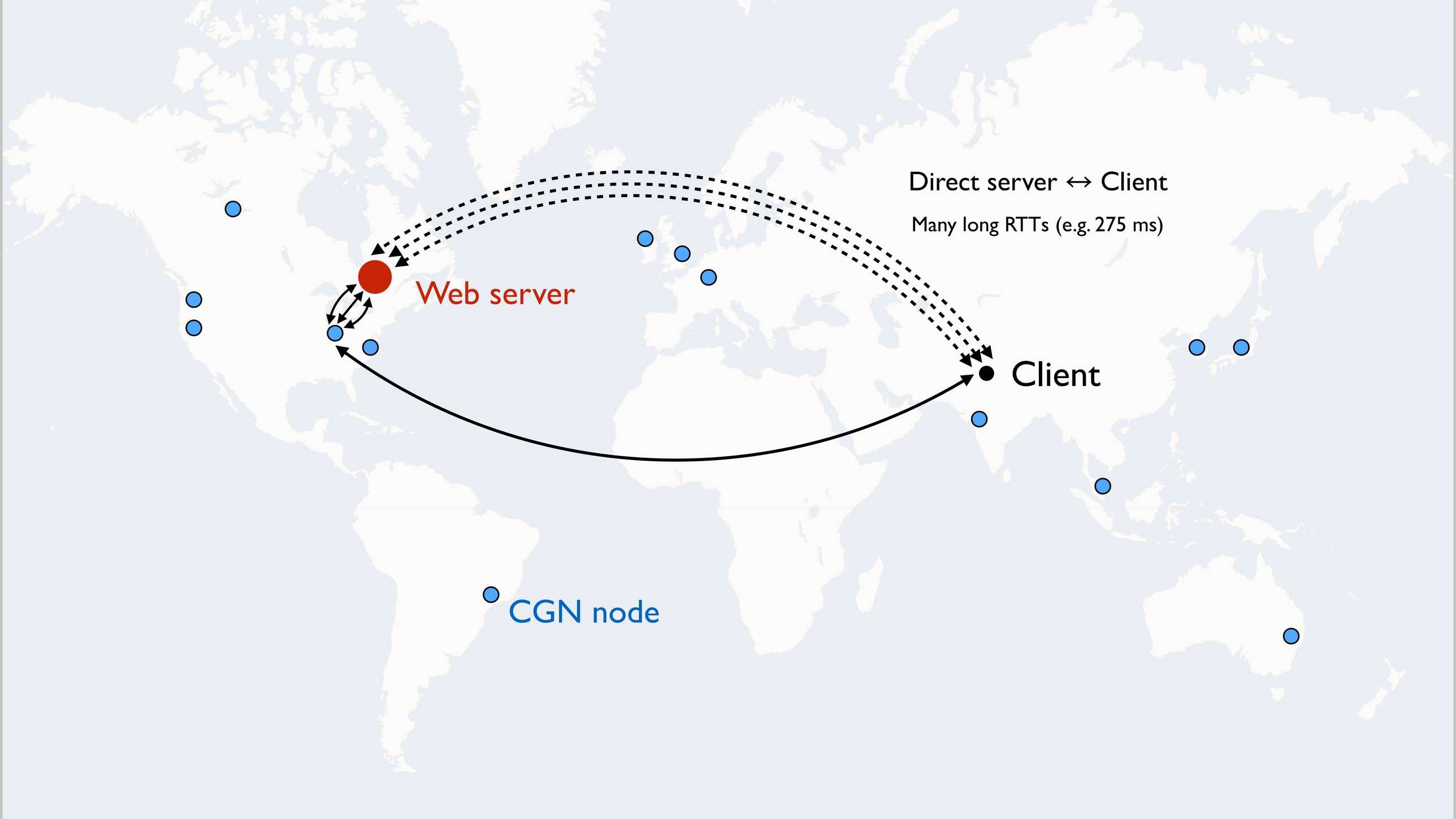


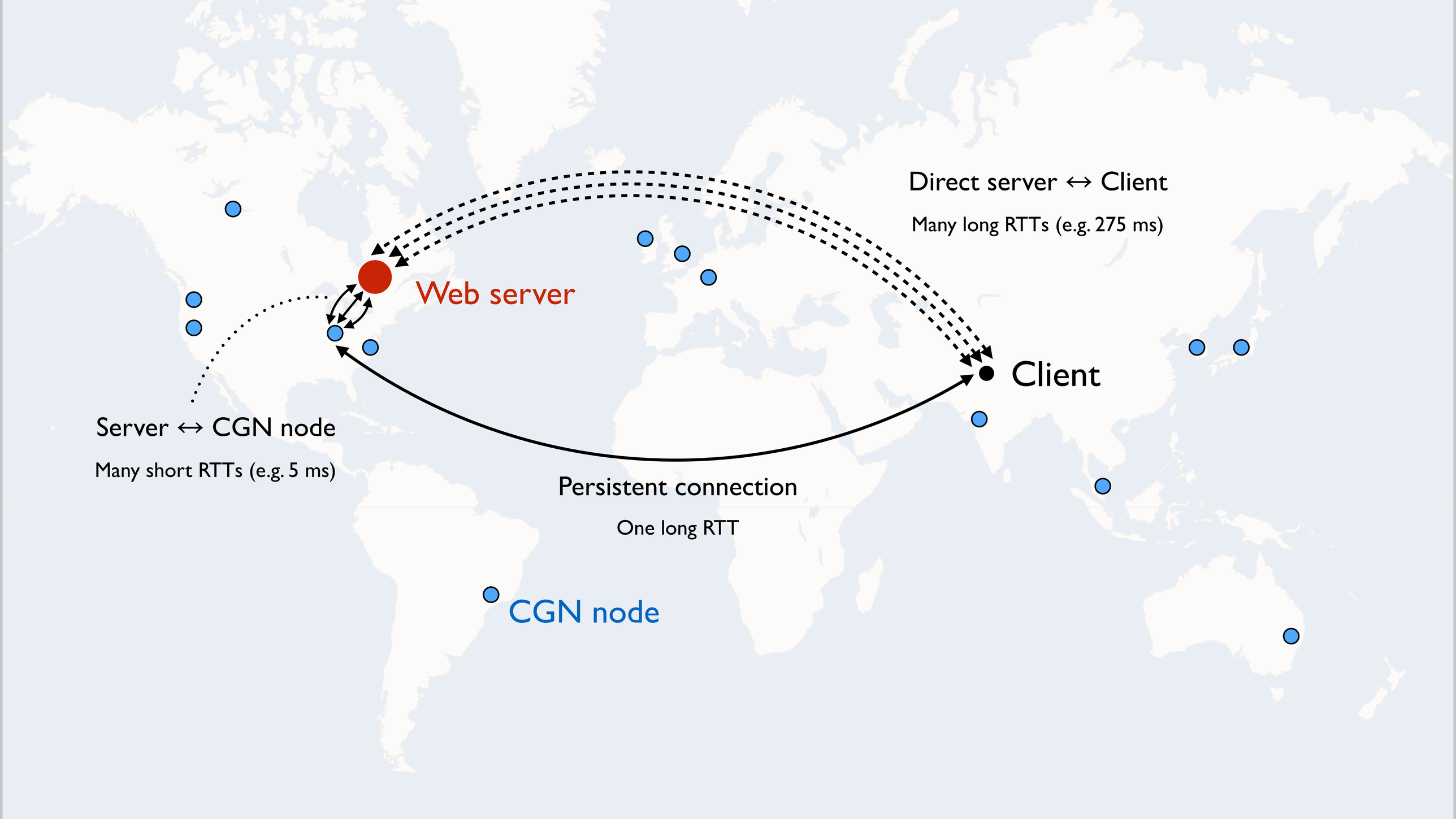


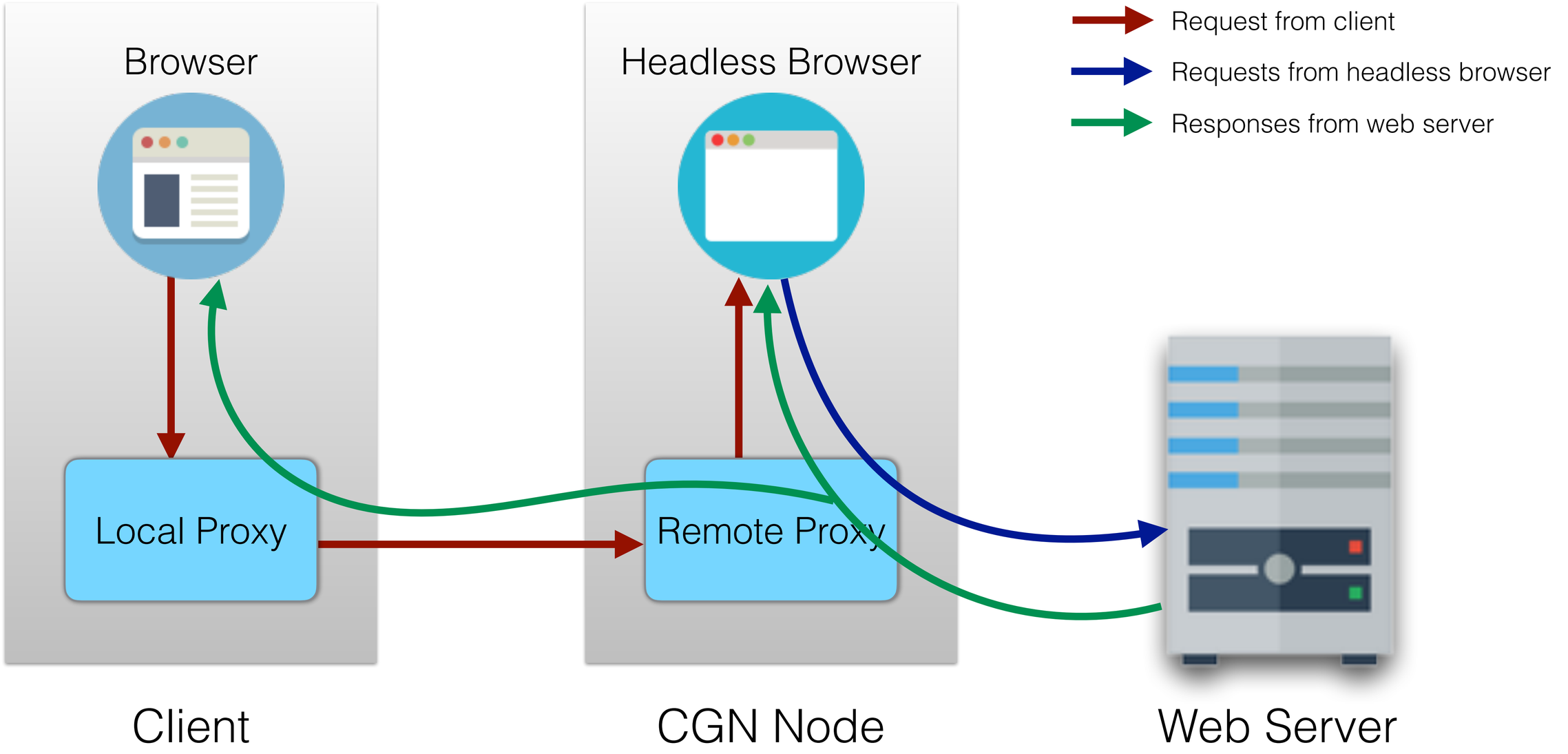
"If the hill will not come to Mahomet, Mahomet will go to the hill."
- Francis Bacon.

Can we reduce Web page load times?


Direct server ↔ Client
Many long RTTs (e.g. 275 ms)
Web server
Client


Direct server ↔ Client
Many long RTTs (e.g. 275 ms)
Web server
Client
CGN node

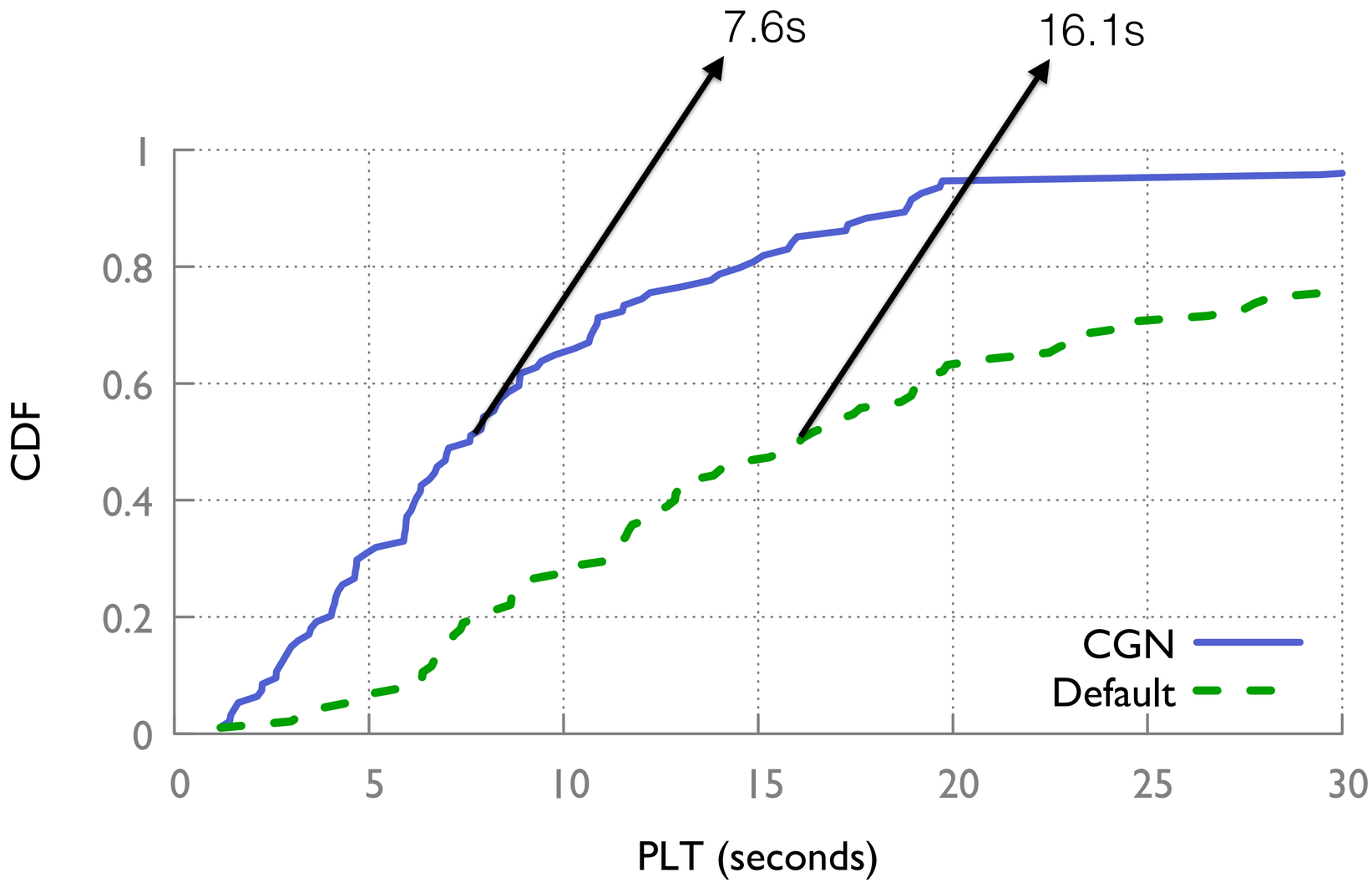

Direct server ↔ Client
Many long RTTs (e.g. 275 ms)
Web server
Client
Server ↔ CGN node
Many short RTTs (e.g. 5 ms)
Persistent connection
One long RTT
CGN node


Request from client
Requests from headless browser
Responses from web server
Browser
Headless Browser
Local Proxy
Remote Proxy
Client
CGN Node
Web Server

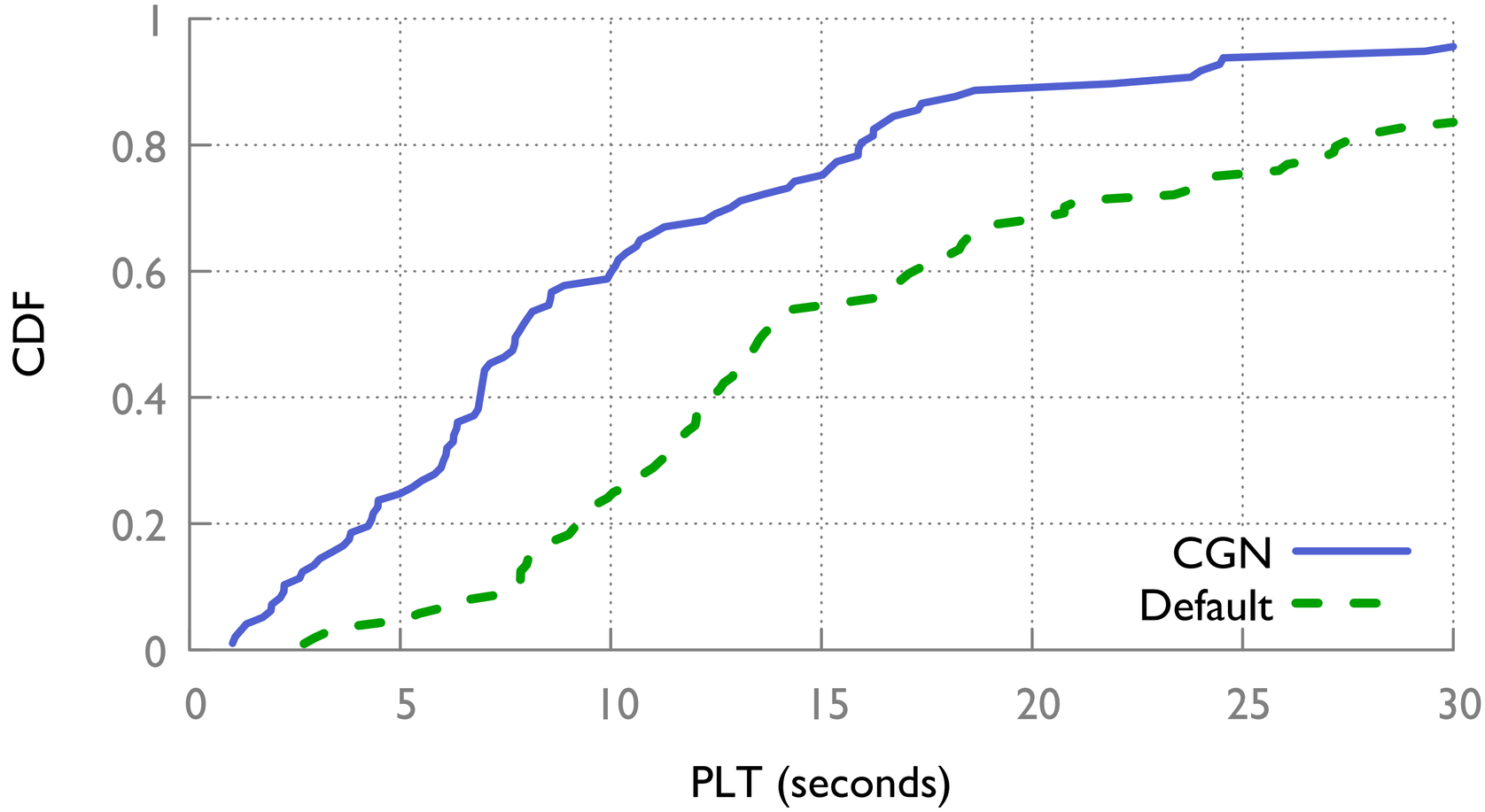
Evaluation

- Headless browser at CGN Node: **PhantomJS**
- Operate only **2 CGN nodes** for the experiments
  - North California, USA and Frankfurt, Germany
- Client browser: Google's **Chrome** browser
- Automating page loads: **sitespeed.io**
- Client location: **Lahore**, Pakistan

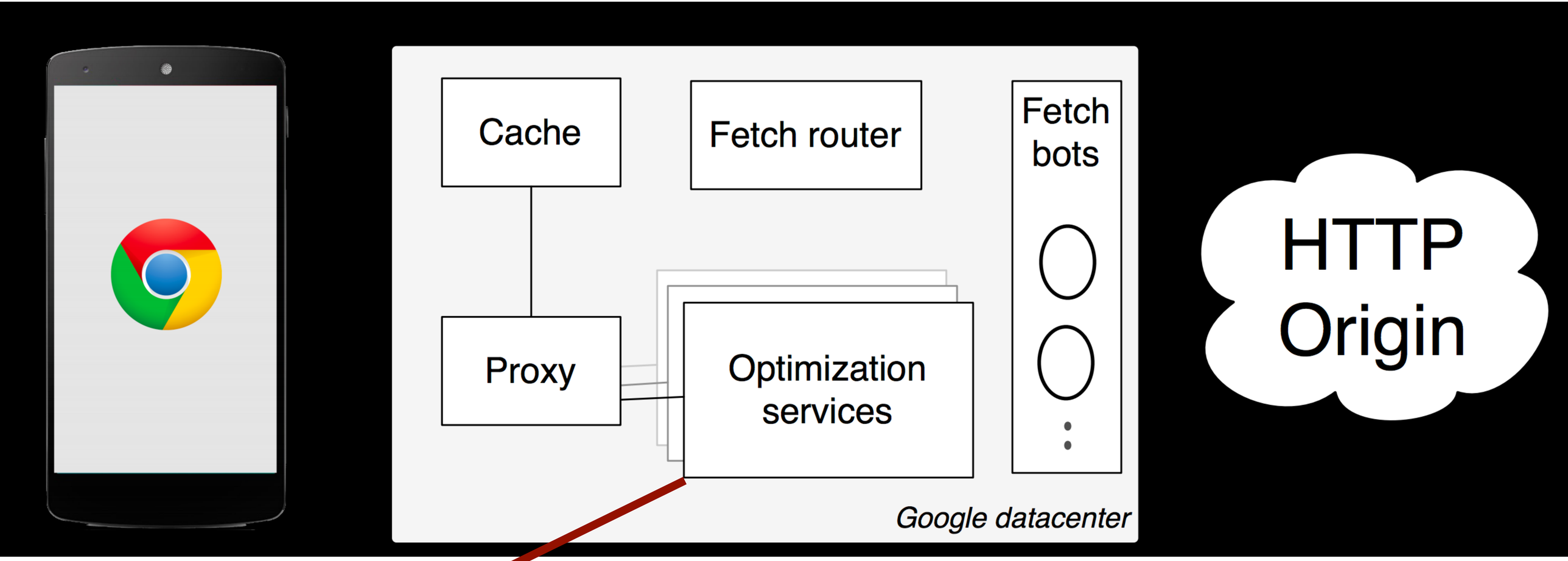
53% reduction in PLTs for the **top** 100 domains



43% reduction in PLTs for 100 **random** domains



CGN vs Google's Flywheel

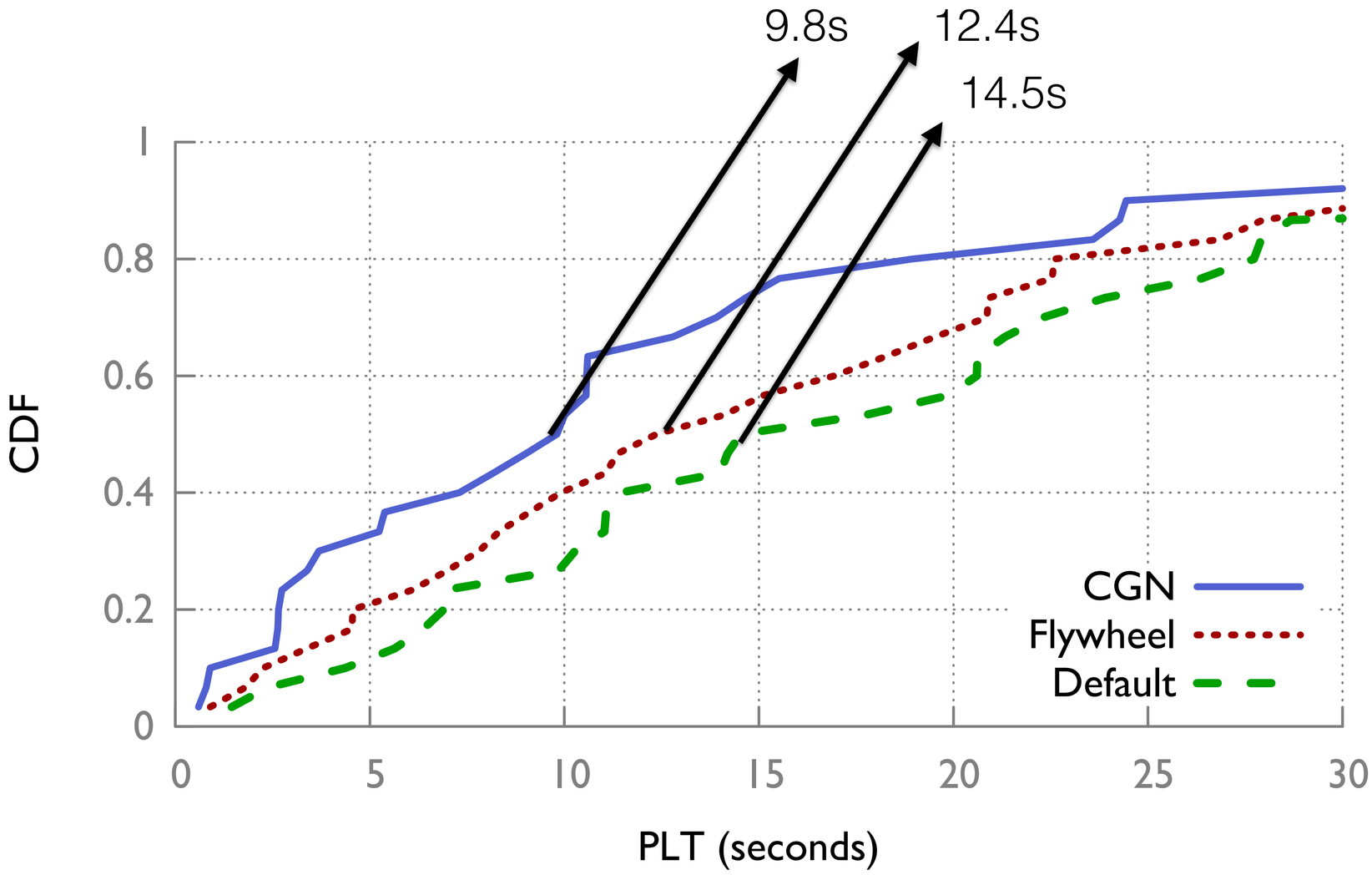


Taken from: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi15_slides_agababov.pdf

Compression

Optimizations are orthogonal

21% faster than Google's Flywheel



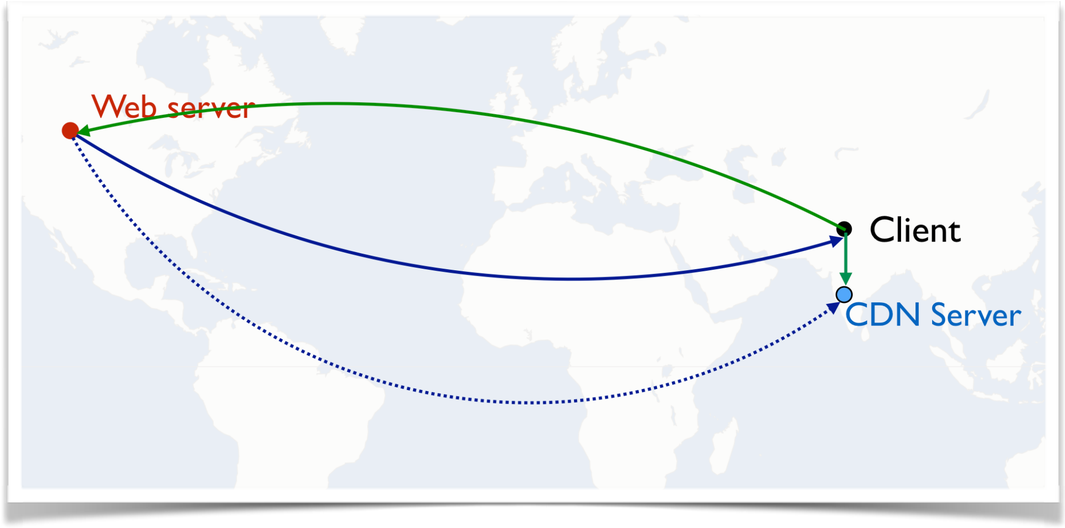
Isn’t this very expensive?

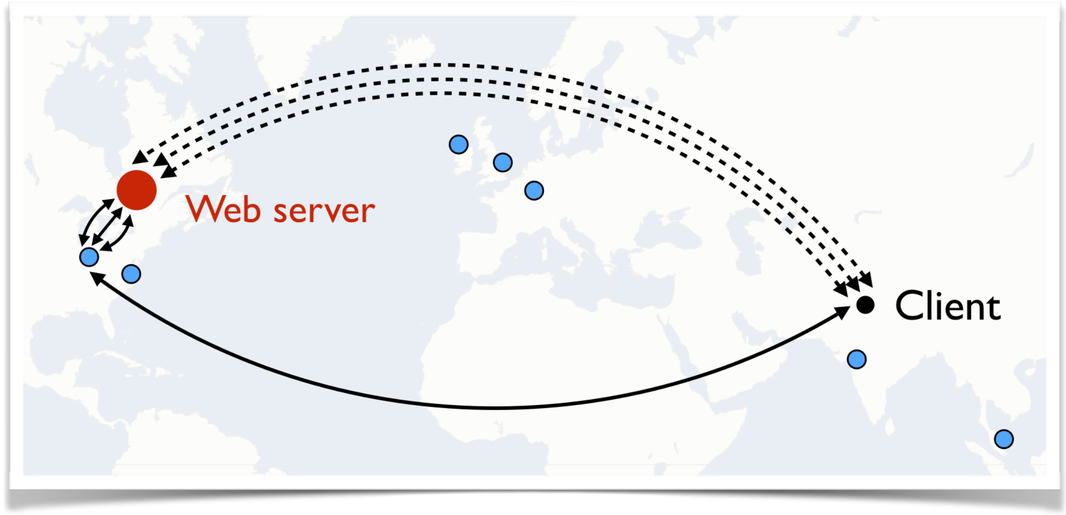
- Reserved m4.10xlarge instances
  - Priciest at Sao Paulo
  - Computation cost: $1.828/hour
  - Network cost: $0.01/GB
- Average request (2MB)
  - ~300 ms CPU time
  - ~2 MB network bandwidth in both directions
- Average usage: 5000 requests/month
  - Cost: **$0.934** per user per month
  - Lahore: ~10% of the cost of typical broadband plans

Ongoing & Future Work

- More measurements - Azure, PlanetLab, etc.
- Speeding up the headless browser
- Visual completion metric instead of PLT
- Reducing cost further
- Incorporating compression, caching, etc. (like Flywheel)
- Security
  - HTTPS: Trust model
  - Hiding content from CGN nodes
  - Sandboxing of user requests from each other
- Management plane

CDN vs CGN





- Do we need 233,000+ servers for Web content delivery?

Summary

- Web server consolidation in or near cloud data centers
- CGN exploits this consolidation for speeding up Web page delivery
- 43-53% faster Web page loads with our preliminary implementation.