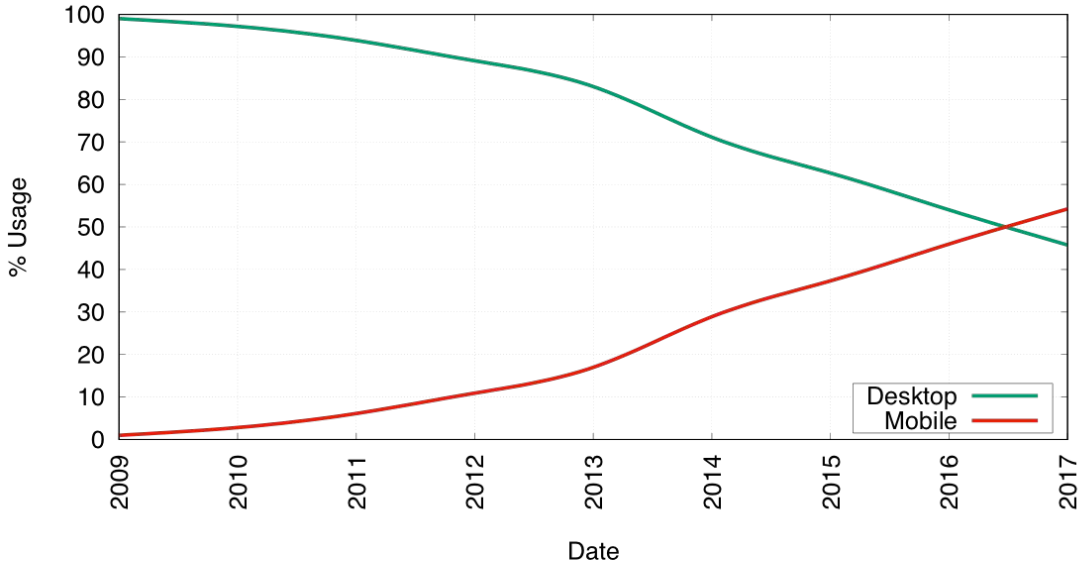
# Vroom: Accelerating the Mobile Web with Server-Aided Dependency Resolution

Vaspol Ruamviboonsuk[1], Ravi Netravali[2],
Muhammed Uluyol[1], Harsha V. Madhyastha[1]

[1]University of Michigan, [2]MIT


# Mobile Web Dominant … but Slow…

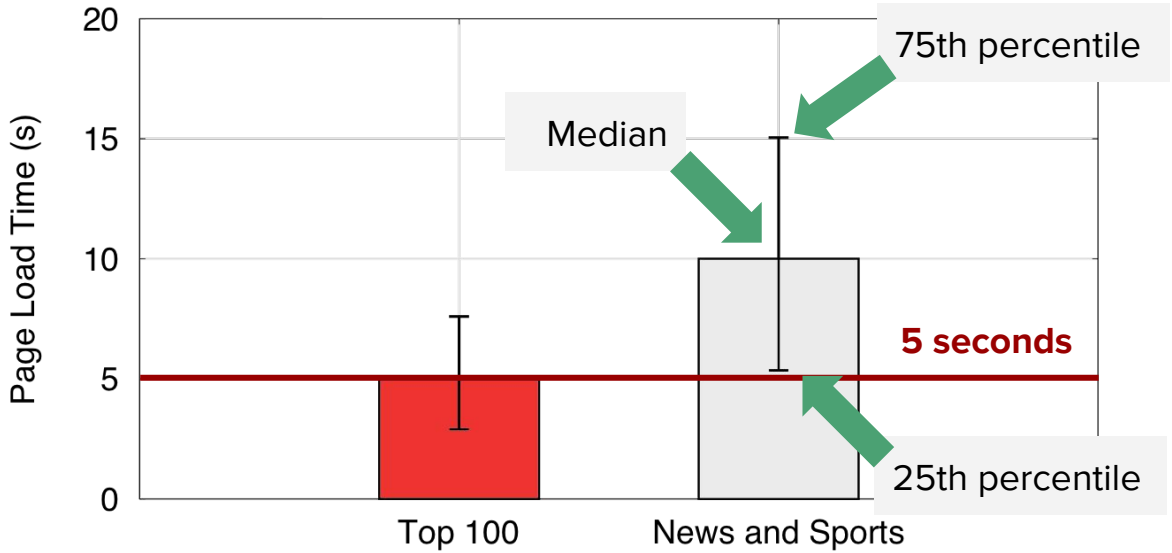



**“9.85s to load median mobile retail sites” - Keynote Systems**

**“Average load time 14s on 4G” - DoubleClick**

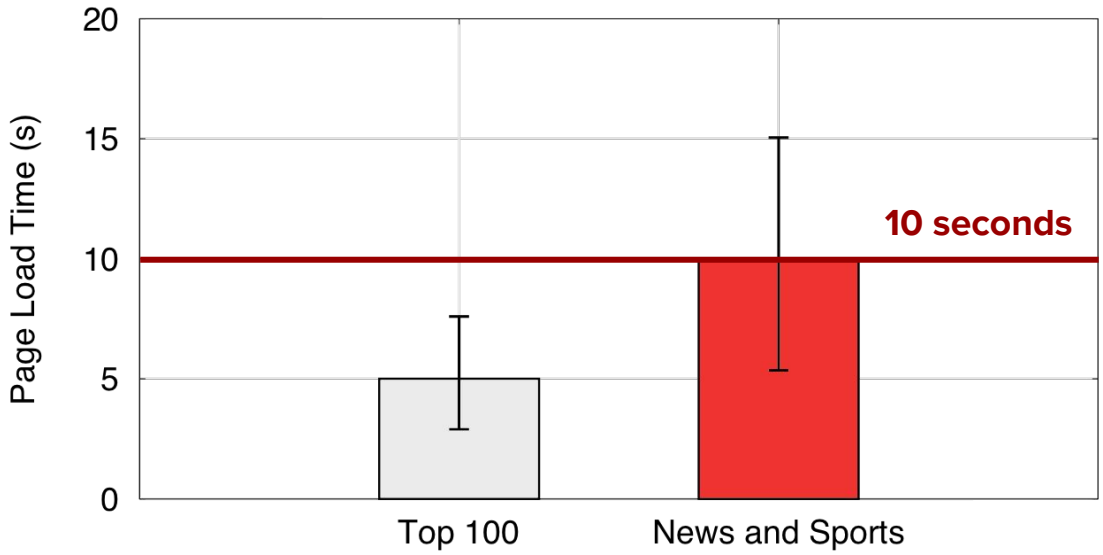
Ref: http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide

# Problem: Slow web page loads




**Mobile Optimized Popular Pages,**
**Nexus 6 Phone, Good LTE network**

# Problem: Slow web page loads




**Mobile Optimized Popular Pages,**
**Nexus 6 Phone, Good LTE network**

# Outline

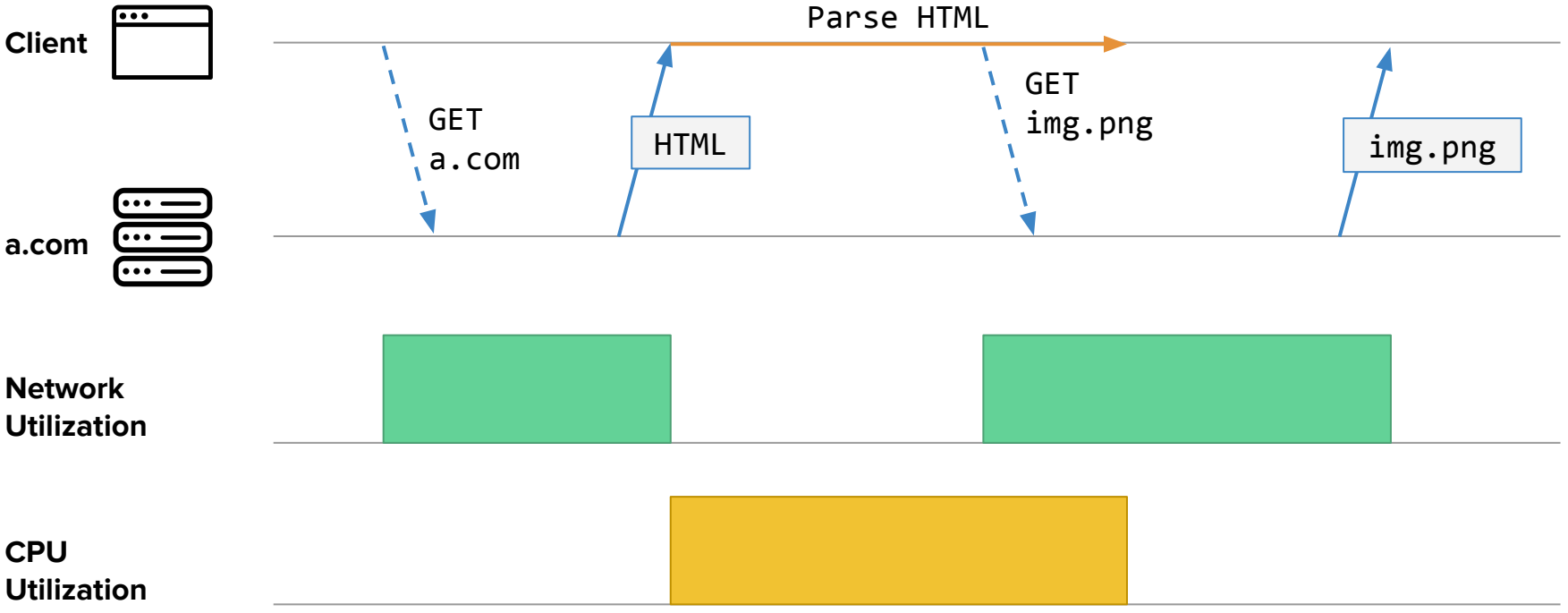
1. Why are page loads slow?
2. Our solution: Vroom
3. Implications of Vroom

# Outline

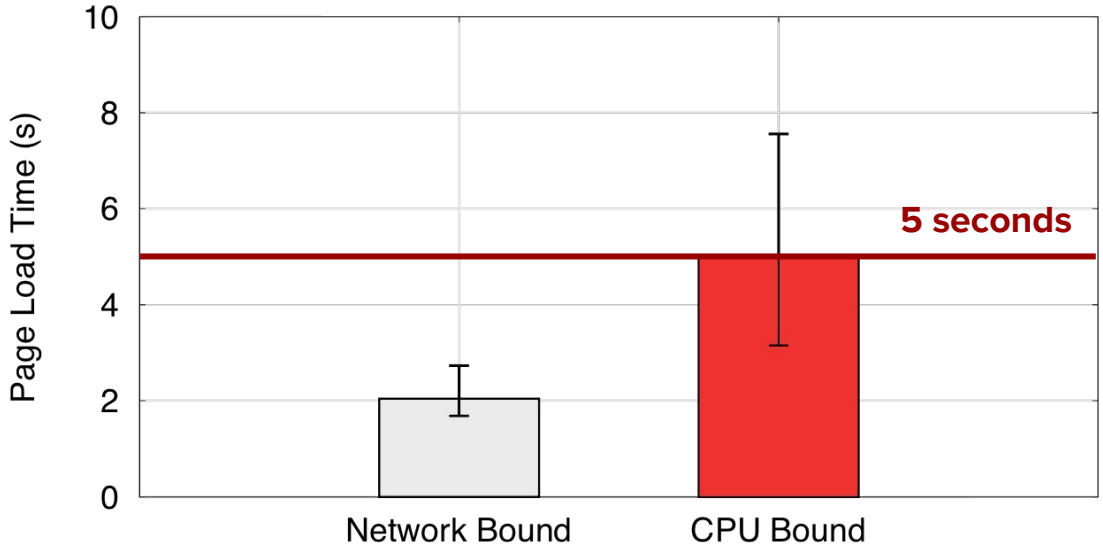
1. **Why are page loads slow?**
2. Our solution: Vroom
3. Implications of Vroom

Loa

**Neither CPU nor network is fully utilized**



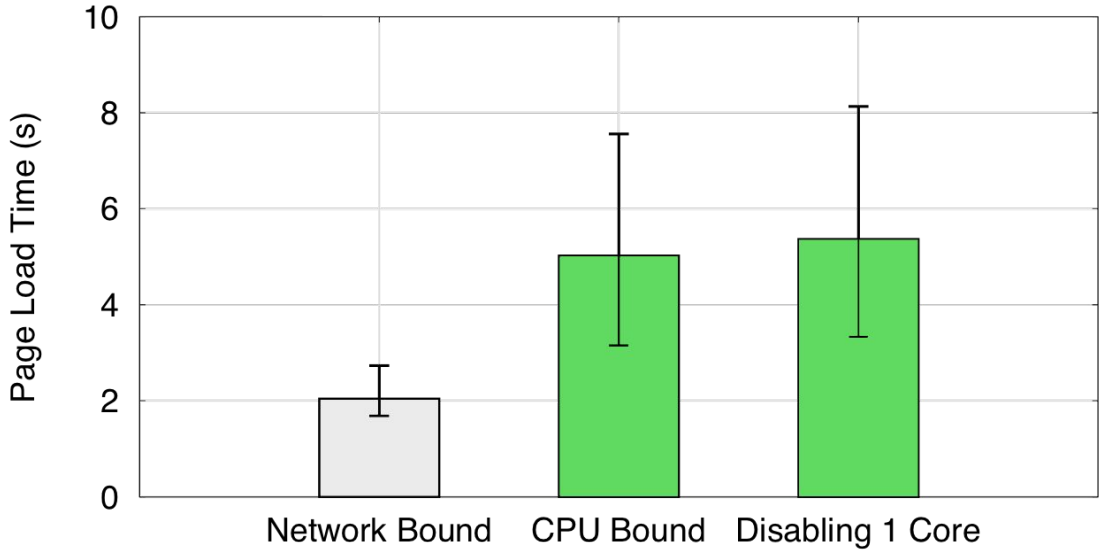
# CPU is the bottleneck



# Is the CPU bottleneck always?

- **Network may be the bottleneck in other settings**
- Trends:
  - *Network: Higher bandwidth and lower latency*
  - But, *CPU only increases in no. of cores*

# More CPU cores do not help much



# Is the CPU bottleneck always?

- **Network may be the bottleneck in other settings**
- Trends:
  - *Network: Higher bandwidth and lower latency*
  - But, *CPU only increases in no. of cores*
- Browser's processing on a page largely serial
- Implication: CPU will be bottleneck in the long-term

# Rethinking how web pages are loaded

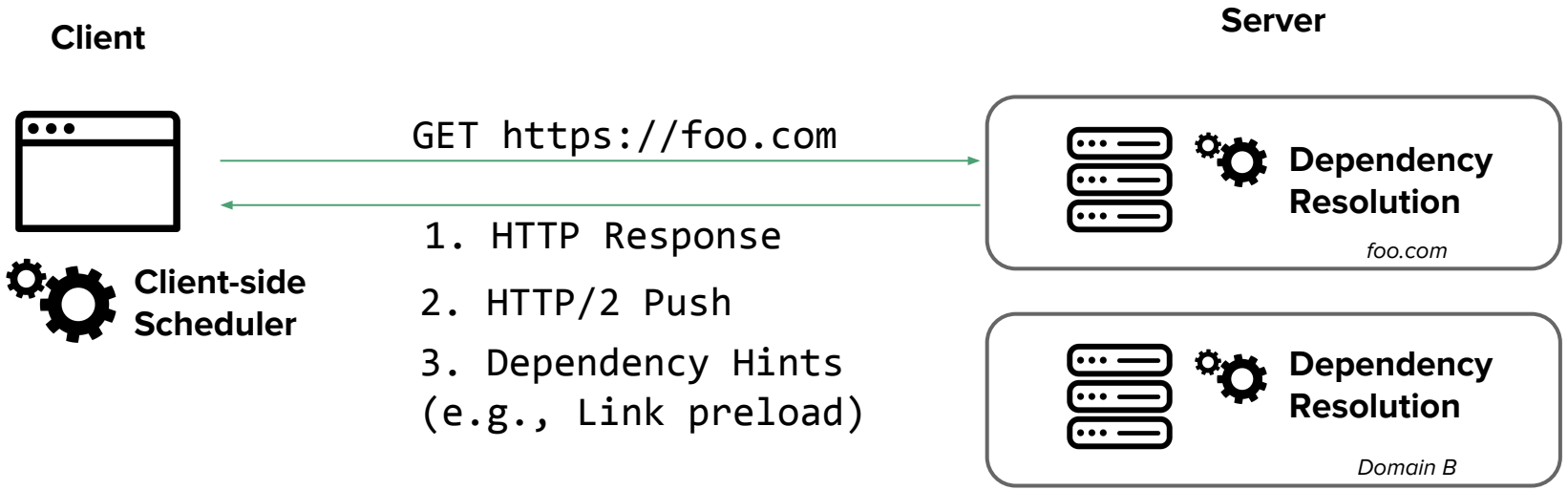
- ***Browsers discover resources from parsing and execution***

- Rethink page load:
  - ***Have servers aid clients in resource discovery***

# Outline

1. Why are page loads slow?
2. **Our solution: Vroom**
3. Implications of Vroom

# Vroom


Client
Server
GET https://foo.com
1. HTTP Response
2. HTTP/2 Push
3. Dependency Hints
(e.g., Link preload)
Client-side
Scheduler
Dependency
Resolution
foo.com
Dependency
Resolution
Domain B

# Challenges to approach

1. How can web servers discover dependencies?
2. How should clients use input from servers?

# Challenges to approach

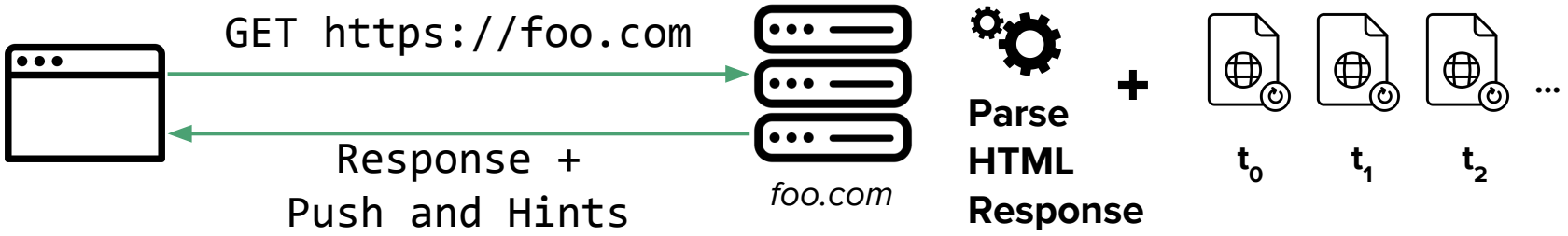
1. **How can web servers discover dependencies?**
2. How should clients use input from servers?

# Strawman Dependency Resolution


**Drawbacks**

- Back-to-back loads differ
- Server cannot account for personalization

# Combined Offline-Online Discovery



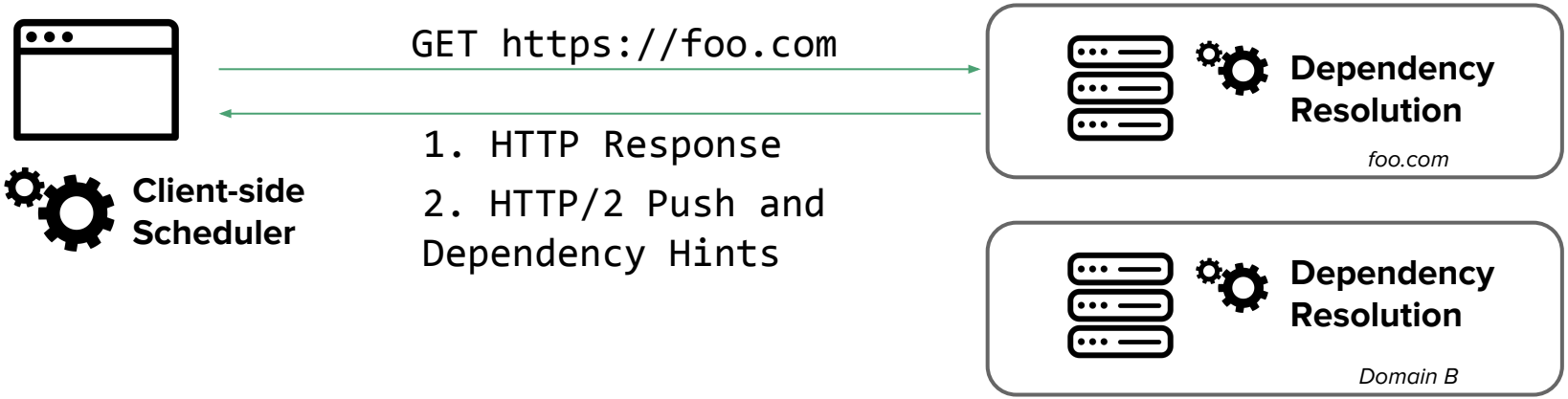

- **Stable dependencies:** Intersection of offline loads
- **Dynamic content:** Online parsing of HTML

# Challenges to approach

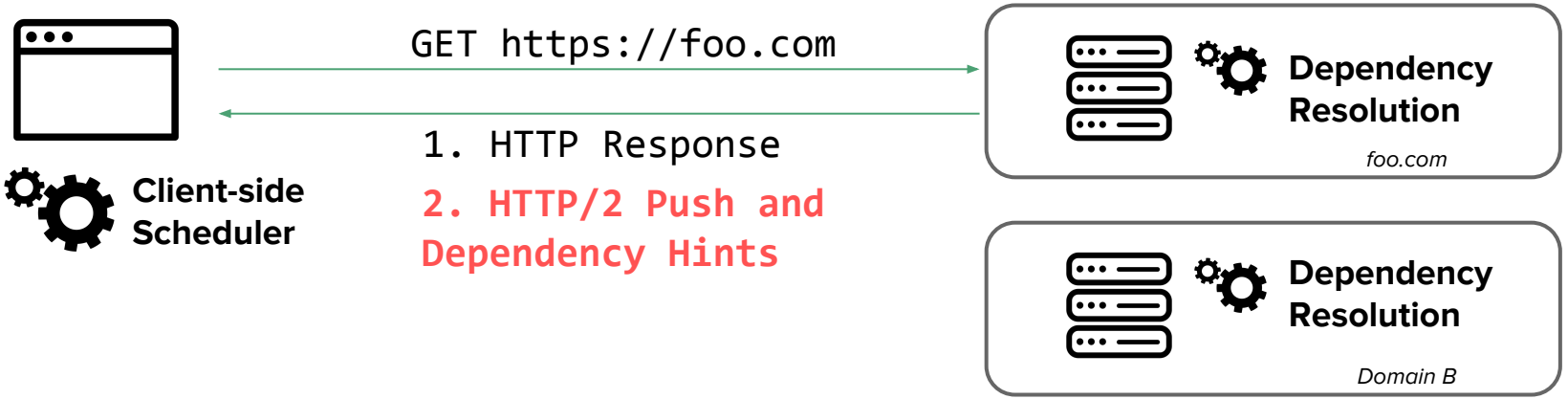
1. How do web servers discover dependencies?
   - *Combine offline and online resource discovery*

2. **How do clients use input from servers?**

# Vroom


GET https://foo.com
1. HTTP Response
2. HTTP/2 Push and Dependency Hints
Client-side Scheduler
Dependency Resolution
foo.com
Dependency Resolution
Domain B

# Push All + Fetch ASAP Approach



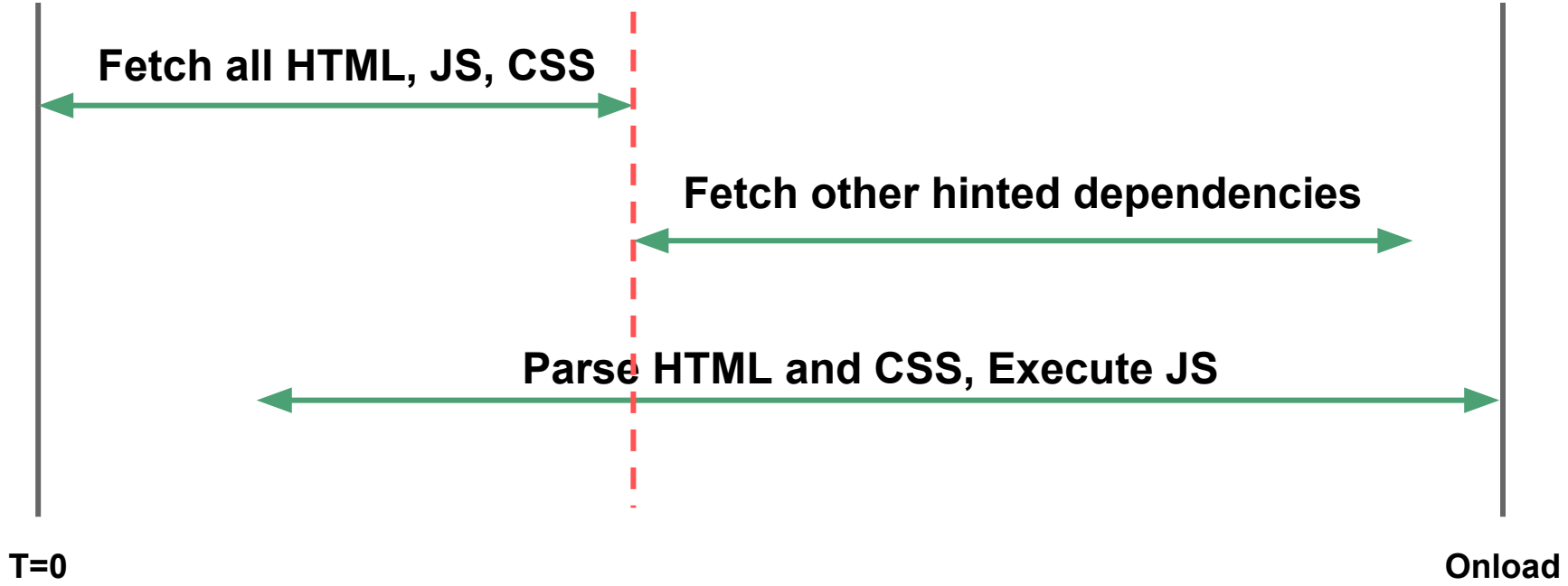

**Every server pushes all resources it could**
**Client fetches immediately upon receiving hint**

# Need for Scheduling

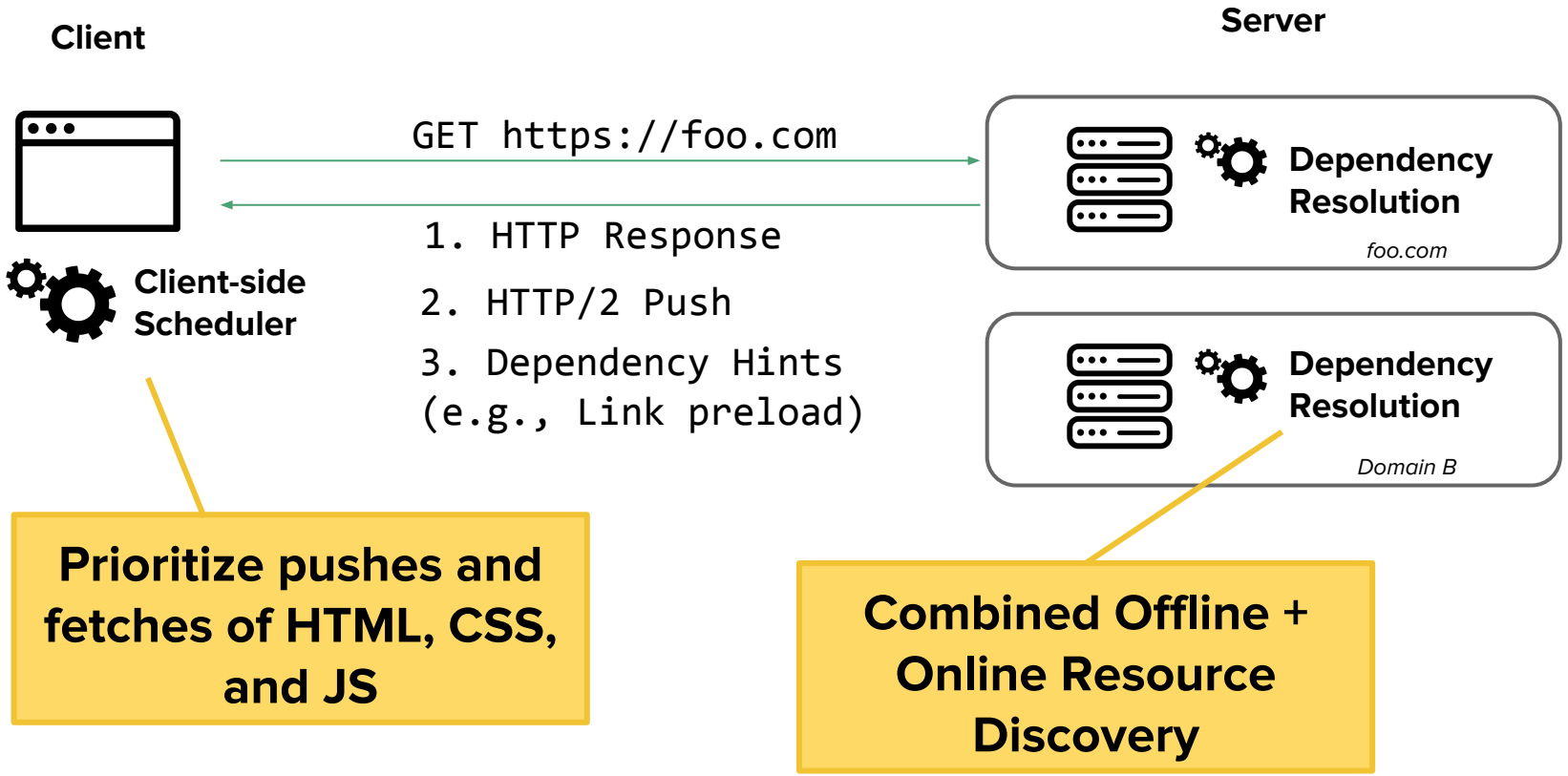
- No speedup with “Push All + Fetch ASAP”
  - Contention for access link bandwidth stalls processing

- **Prioritize pushes and fetches of HTML, CSS, and JS**
  - Schedule in order of processing

# Vroom scheduler in action



# Vroom


Client
Server
GET https://foo.com
1. HTTP Response
2. HTTP/2 Push
3. Dependency Hints
(e.g., Link preload)
Client-side Scheduler
Dependency Resolution
foo.com
Dependency Resolution
Domain B
Prioritize pushes and fetches of HTML, CSS, and JS
Combined Offline + Online Resource Discovery

# Results overview

- **Vroom’s dependency resolution is accurate**
  - Median: 0% false positives and < 5% false negatives

- **Vroom speeds up page loads**
  - Speedup over status quo
  - Simple strawmans don't suffice
  - Speedup even with warm caches

# Results overview

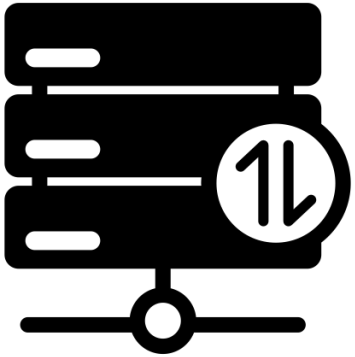
- **Vroom's dependency resolution is accurate**
  - Median: 0% false positives and < 5% false negatives

- **Vroom speeds up page loads**
  - *Speedup over status quo*
  - Simple strawmans don't suffice
  - Speedup even with warm caches

# Evaluation Setup


**Nexus 6**

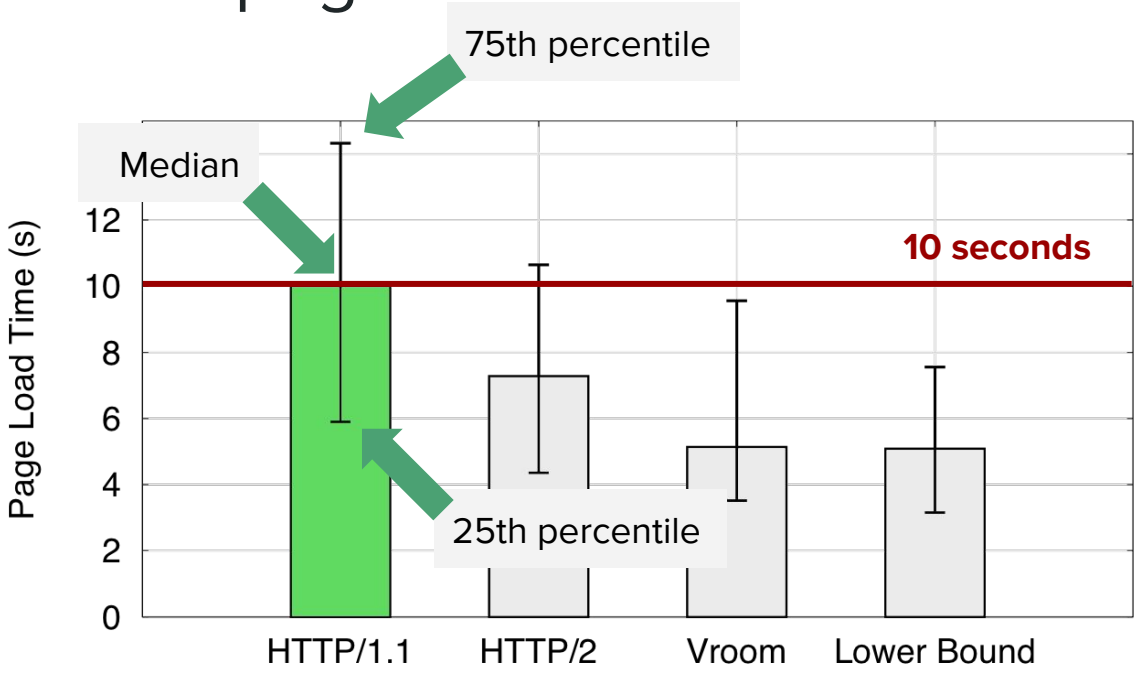

**4G Network**



**Web Record & Replay Environment**

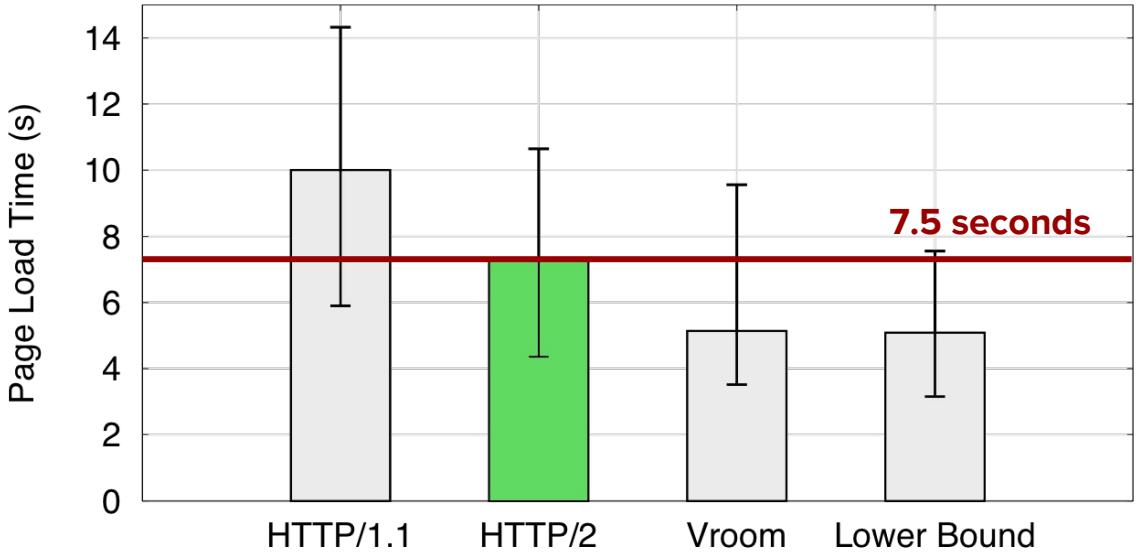
**Attribution**
Server Icon: mungang kim

# Vroom makes page loads much faster



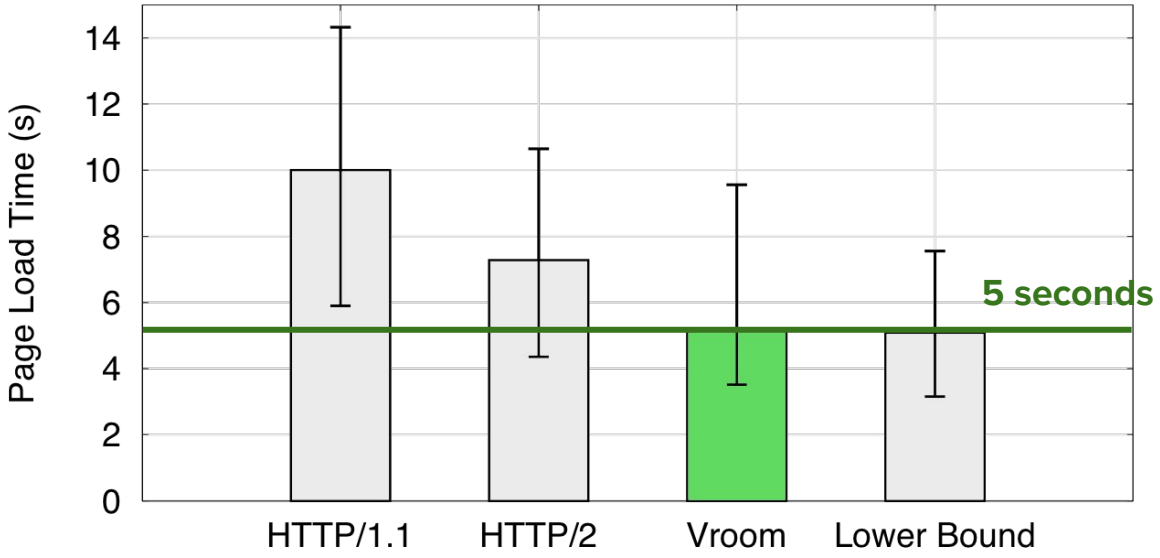

**Alexa top 50 news and 50 sports sites**

# Vroom makes page loads much faster



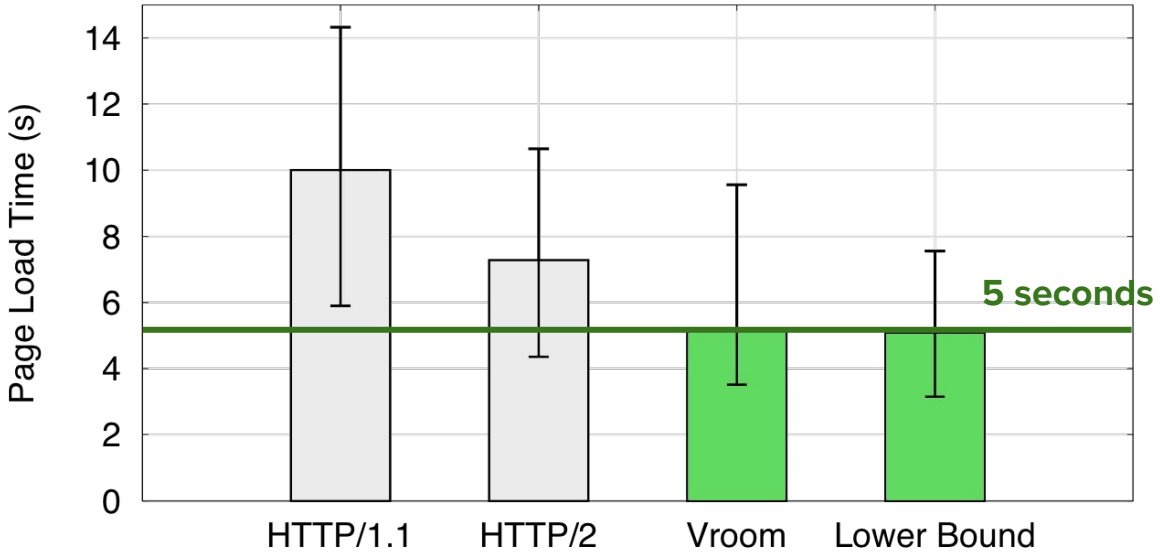

**Alexa top 50 news and 50 sports sites**

# Vroom makes page loads much faster



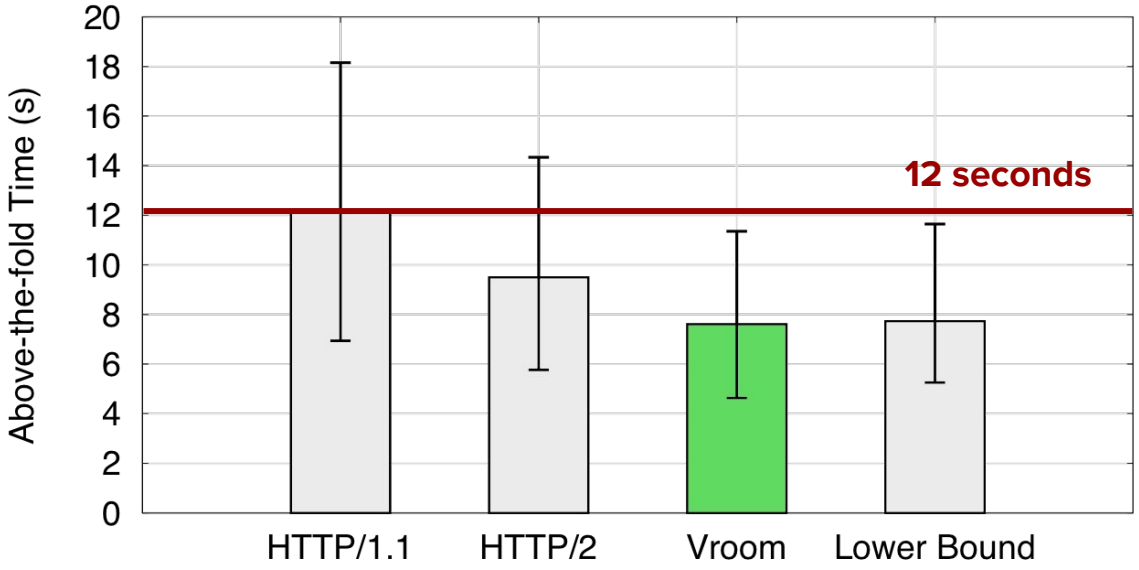

**Alexa top 50 news and 50 sports sites**

# Vroom makes page loads much faster



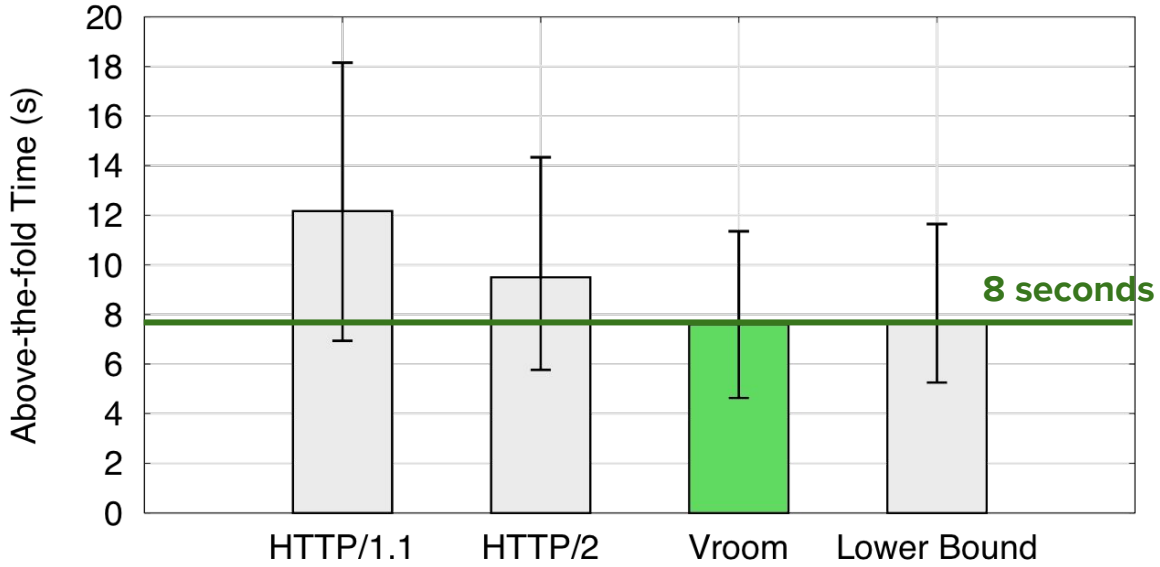

**Alexa top 50 news and 50 sports sites**

# Vroom also improves page loads visually



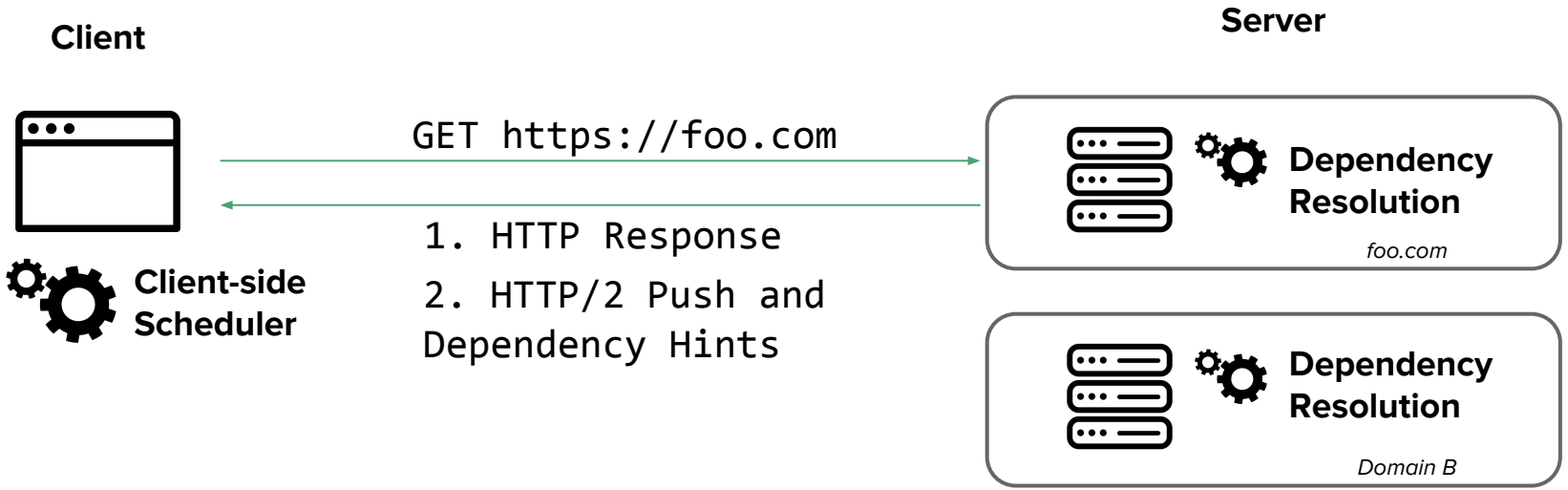

**Alexa top 50 news and 50 sports sites**

# Vroom also improves page loads visually

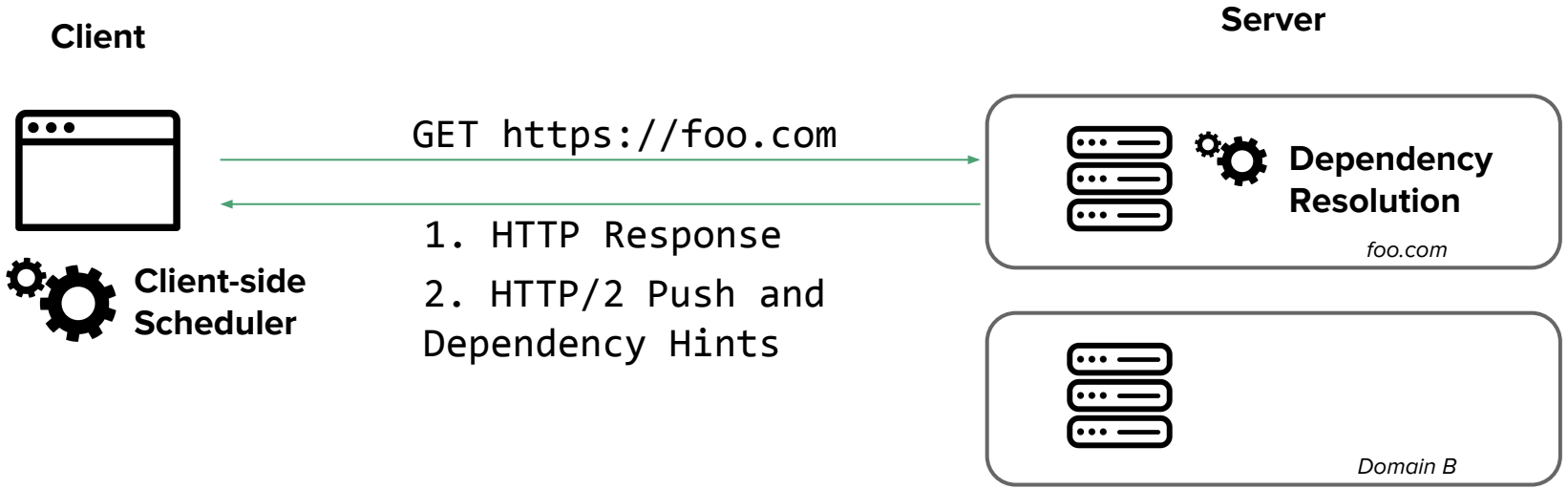



**Alexa top 50 news and 50 sports sites**

# Incrementally Deploying Vroom



# Incrementally Deploying Vroom




***Most benefits is still achievable from incremental deployment***

# Outline

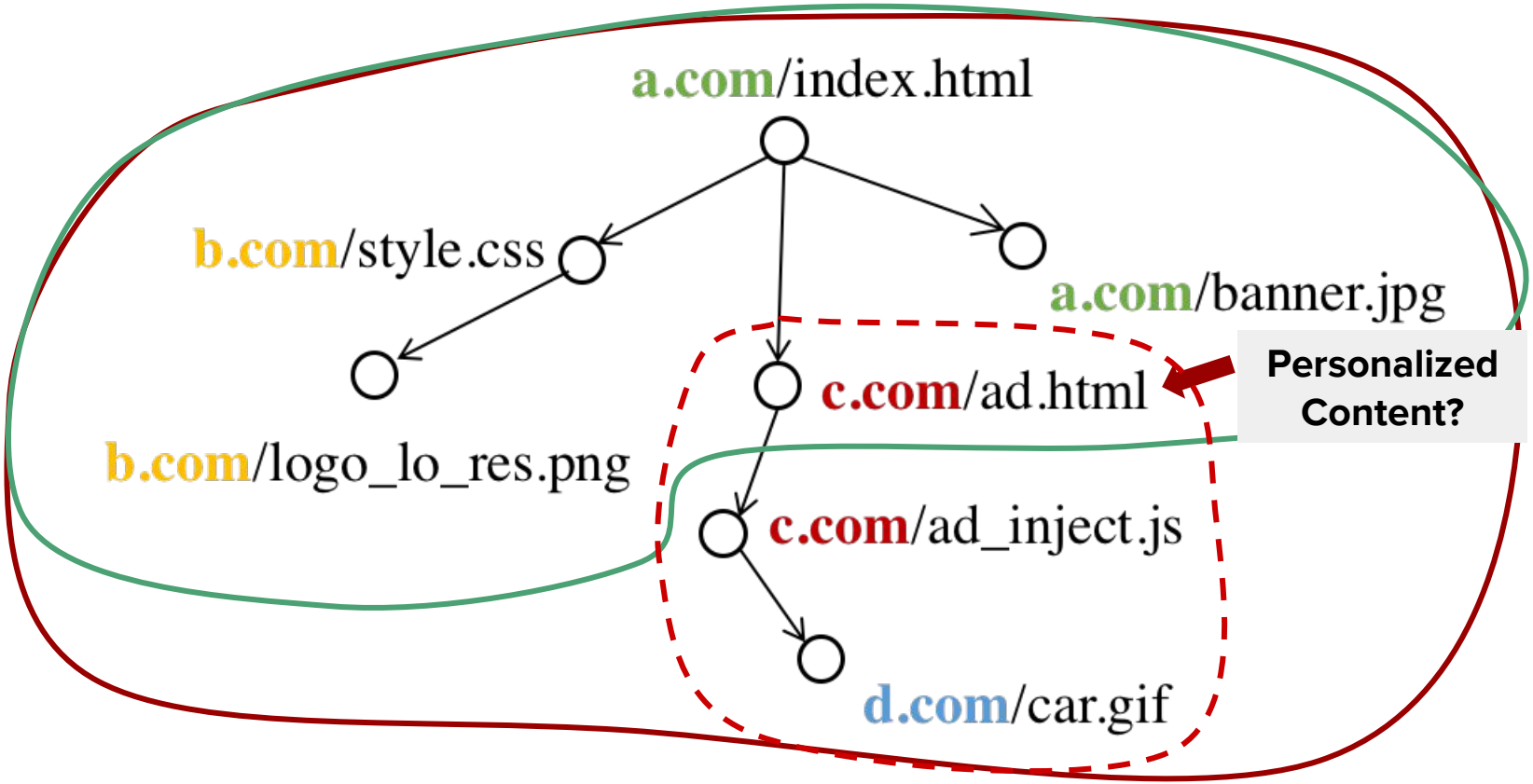
1. Why are page loads slow?
2. Our solution: Vroom
3. **Implications of Vroom**

# Making Vroom Practical

- H2 Push and Link Preload enable server-aided resource discovery
- ***Requires offline discovery of stable resources on pages***
  - Consumes CPU and network at servers

- **Crowdsource offline dependency resolution**
  - Browsers could upload URLs of resources seen in page loads

# Client Aiding Offline Dependency Resolution



# Prioritizing Preloads

- **Do not fetch all dependencies at the same time**
- Group dependencies into different priorities
- Perform fetch in stages based on dependency priorities
- Include priority with Link preload e.g.

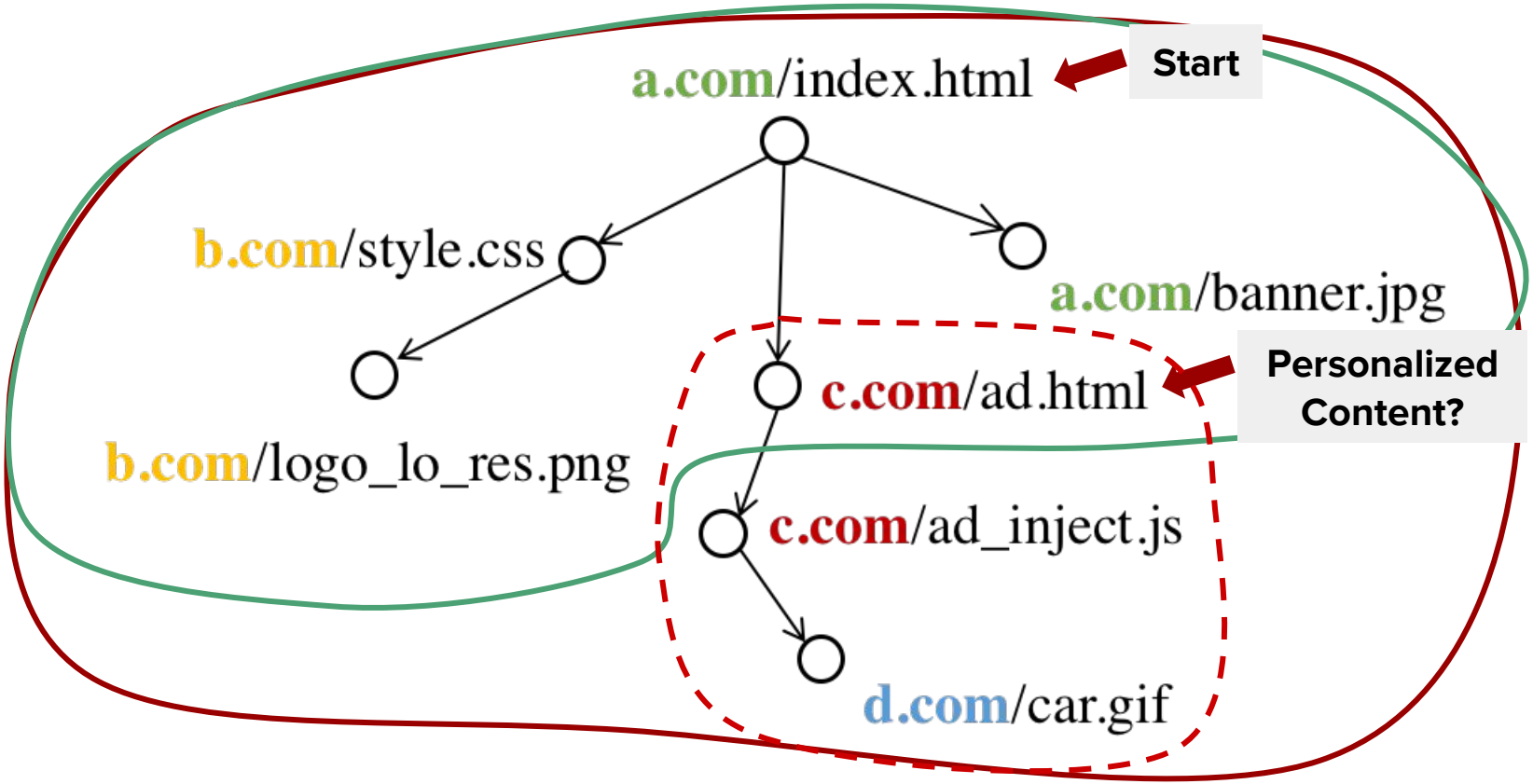
```
<link rel="preload" href="..." priority="high"></link>
```

# Conclusion

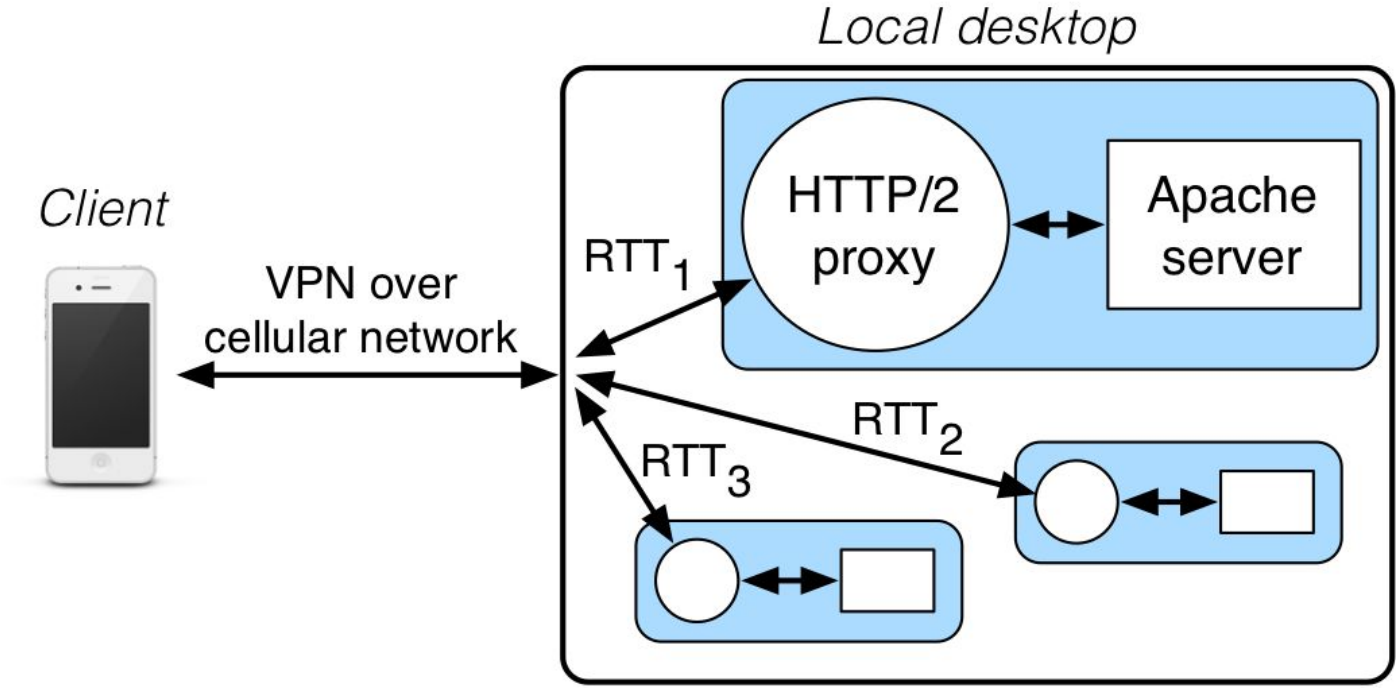
- **Vroom: End-to-end solution that fully utilizes CPU/Network**
- Decouples dependency discovery from parsing and execution
- Decreases median page load time by 5s for popular sites

# Backup

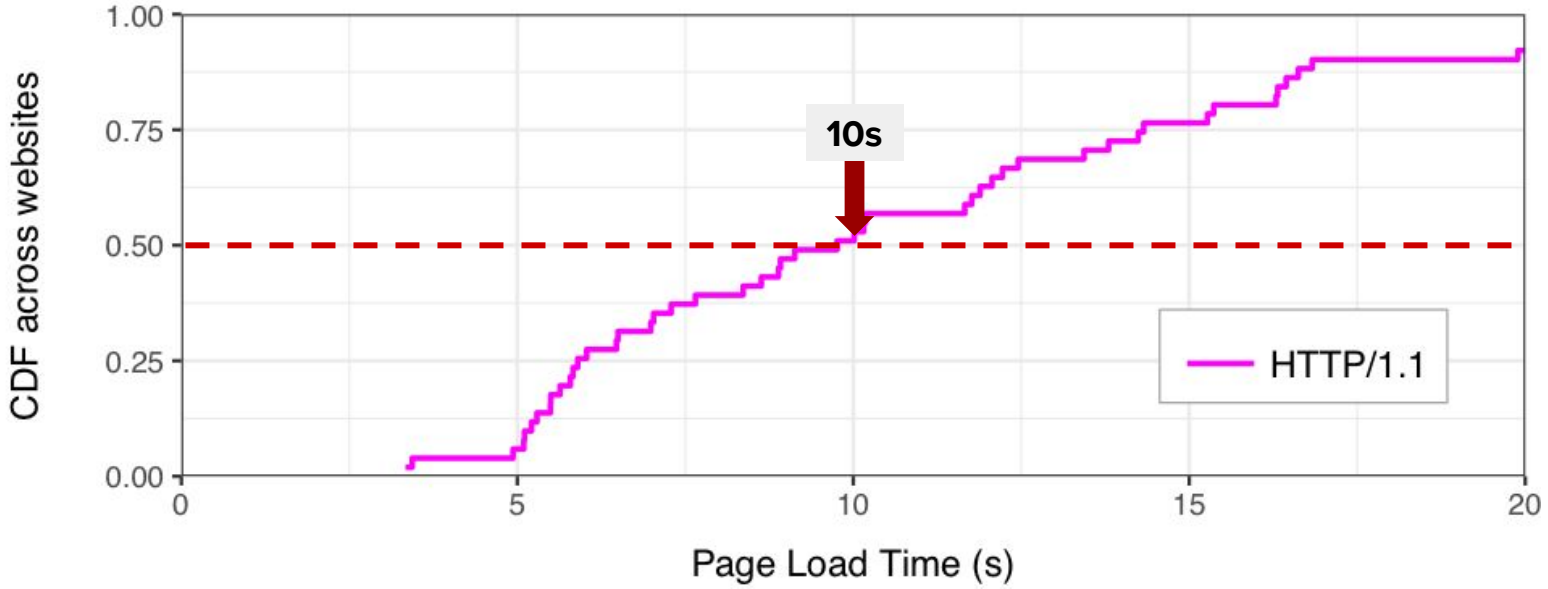
# Personalized Dependencies from Third-party Domains



# Evaluation Setup



# Vroom makes page loads much faster




**Alexa top 50 news and 50 sports sites**