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

Vaspol Ruamviboonsuk\textsuperscript{1}, Ravi Netravali\textsuperscript{2}, Muhammed Uluyol\textsuperscript{1}, Harsha V. Madhyastha\textsuperscript{1}

\textsuperscript{1}University of Michigan, \textsuperscript{2}MIT
Mobile Web Dominant ... but Slow...

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

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

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
Neither CPU nor network is fully utilized
CPU is the bottleneck

![Graph showing page load times between network bound and CPU bound scenarios. The CPU bound scenario takes significantly longer, indicated by a red bar reaching 5 seconds.]
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-side Scheduler

Client

GET https://foo.com

1. HTTP Response
2. HTTP/2 Push
3. Dependency Hints (e.g., Link preload)

Server

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?
GET https://foo.com

1. HTTP Response
2. HTTP/2 Push and Dependency Hints
Push All + Fetch ASAP Approach

GET https://foo.com

1. HTTP Response
2. HTTP/2 Push and Dependency Hints

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

T=0

Fetch all HTML, JS, CSS

Parse HTML and CSS, Execute JS

Fetch other hinted dependencies

Onload
Vroom

Client

GET https://foo.com
1. HTTP Response
2. HTTP/2 Push
3. Dependency Hints (e.g., Link preload)

Server

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
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

Page Load Time (s)

HTTP/1.1  HTTP/2  Vroom  Lower Bound

7.5 seconds
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

Client-side Scheduler

GET https://foo.com

1. HTTP Response
2. HTTP/2 Push and Dependency Hints
Incrementally Deploying Vroom

![Diagram showing the process of incrementally deploying Vroom]

**Client**
- Client-side Scheduler

**Server**
- Dependency Resolution
  - foo.com
- Domain B

GET https://foo.com

1. HTTP Response
2. HTTP/2 Push and Dependency Hints

*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

● **Crowdsourced offline dependency resolution**
  ○ Browsers could upload URLs of resources seen in page loads
Client Aiding Offline Dependency Resolution

a.com/index.html

b.com/style.css

b.com/logo_lo_res.png

c.com/ad.html

c.com/ad_inject.js

d.com/car.gif

Personalized Content?
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

```
Start

a.com/index.html

b.com/style.css

b.com/logo_lo_res.png

c.com/ad.html

c.com/ad_inject.js

d.com/car.gif

Personalized Content?
```
Evaluation Setup

Client

VPN over cellular network

HTTP/2 proxy

Apache server

Local desktop

RTT₁

RTT₂

RTT₃
Vroom makes page loads much faster

Alexa top 50 news and 50 sports sites