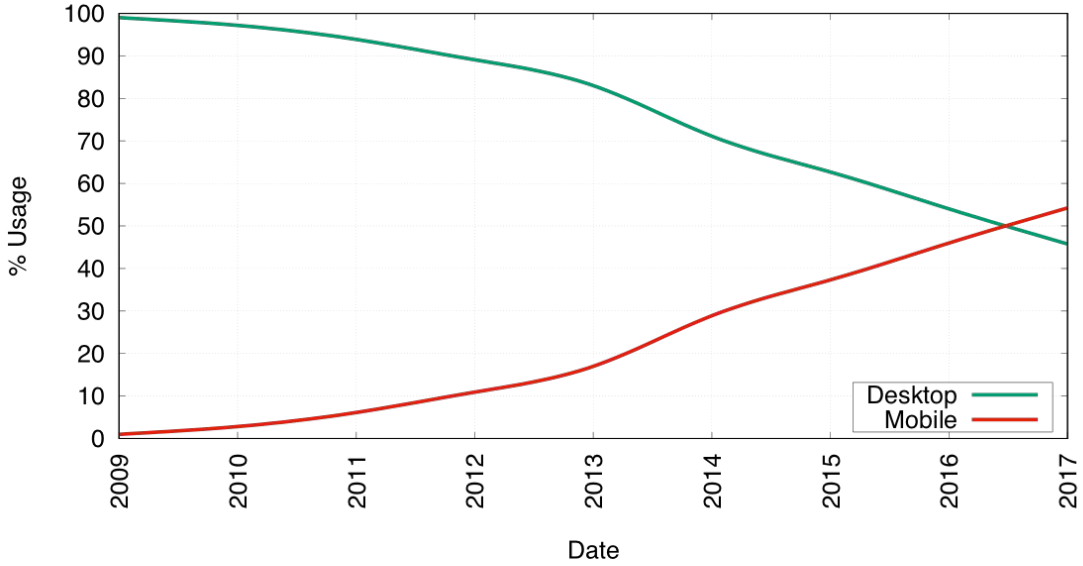
Vroom: Accelerating the Mobile Web with Server-Aided Dependency Resolution

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

[1]University of Michigan, [2]MIT

Mobile Web Dominant … but Slow…

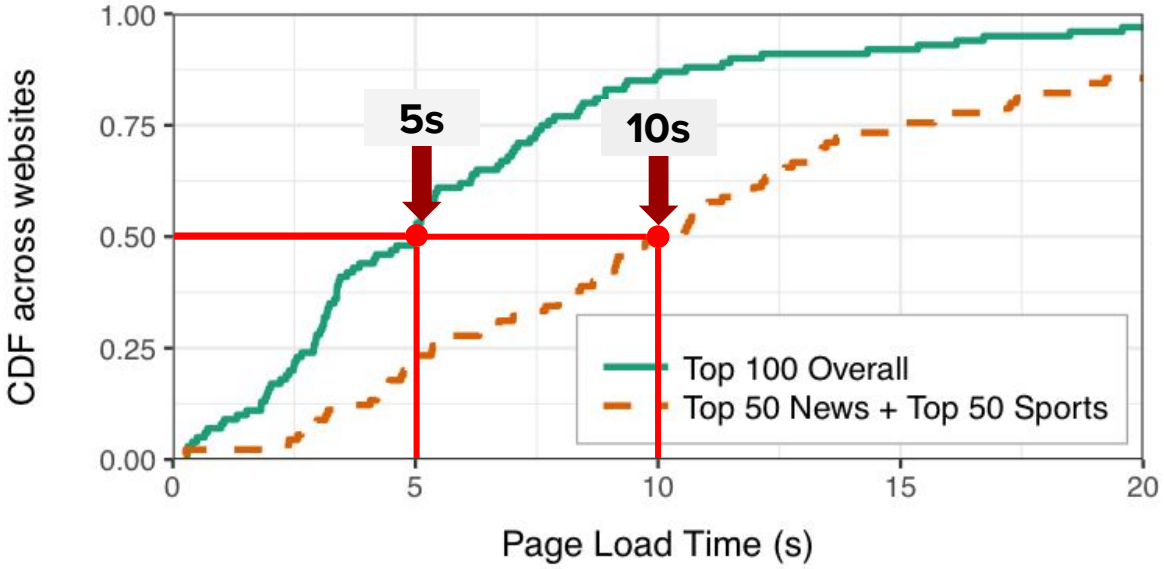


“9.85s to load median mobile retail sites” - Keynote Systems

“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,
State of the Art Phone, Good LTE network

Simple Example Page

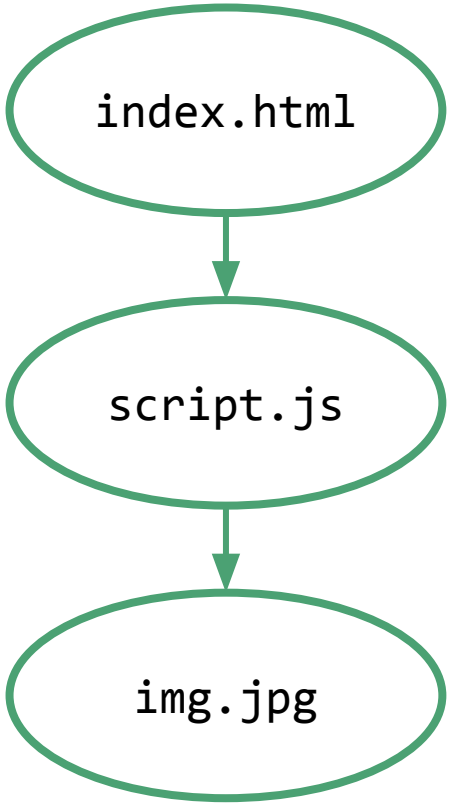
a.com/index.html

```
<html>
  …
  <script src=”script.js”></script>
  …
</html>
```

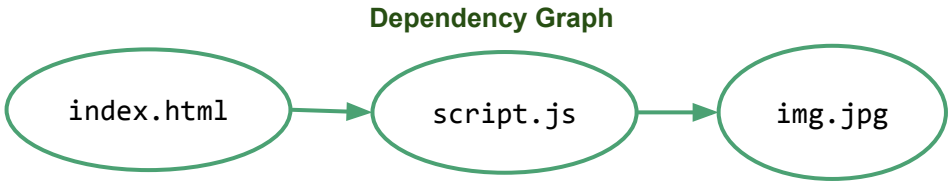
a.com/script.js

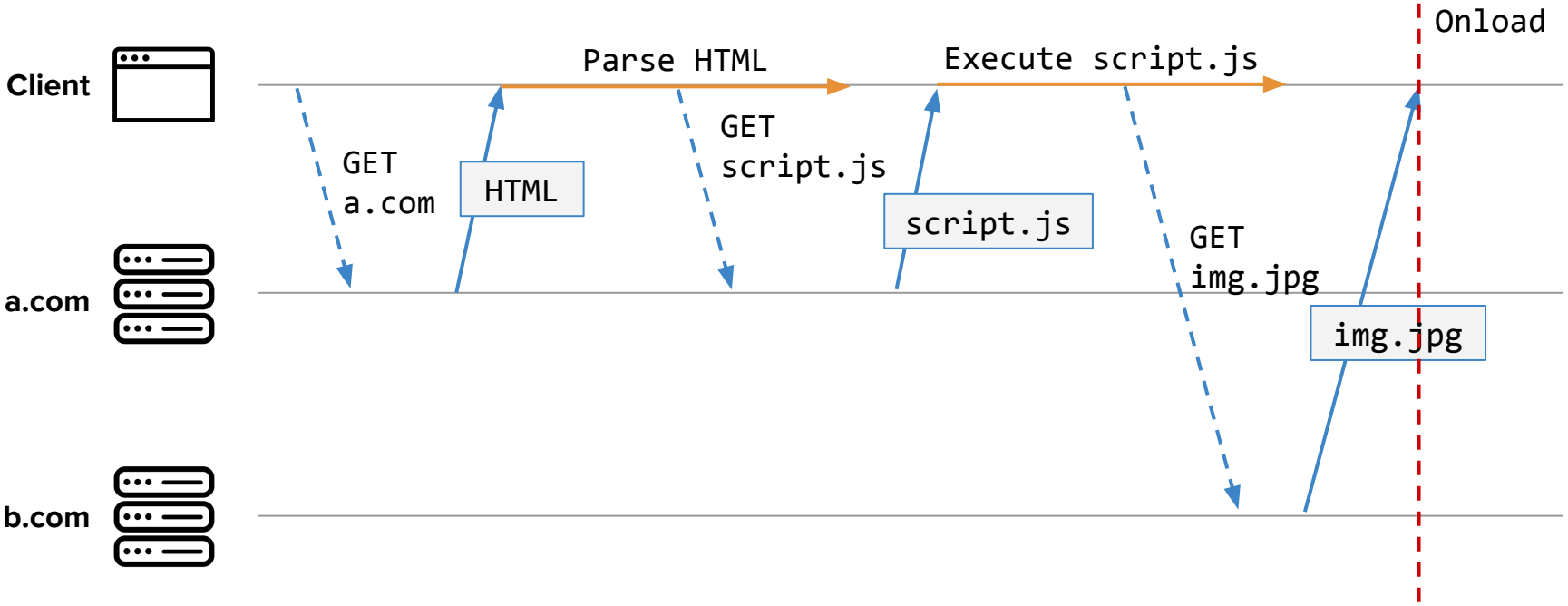
```
var img = new Image();
img.src = “b.com/img.jpg”;
document.body.appendChild(img);
```

Dependency Graph

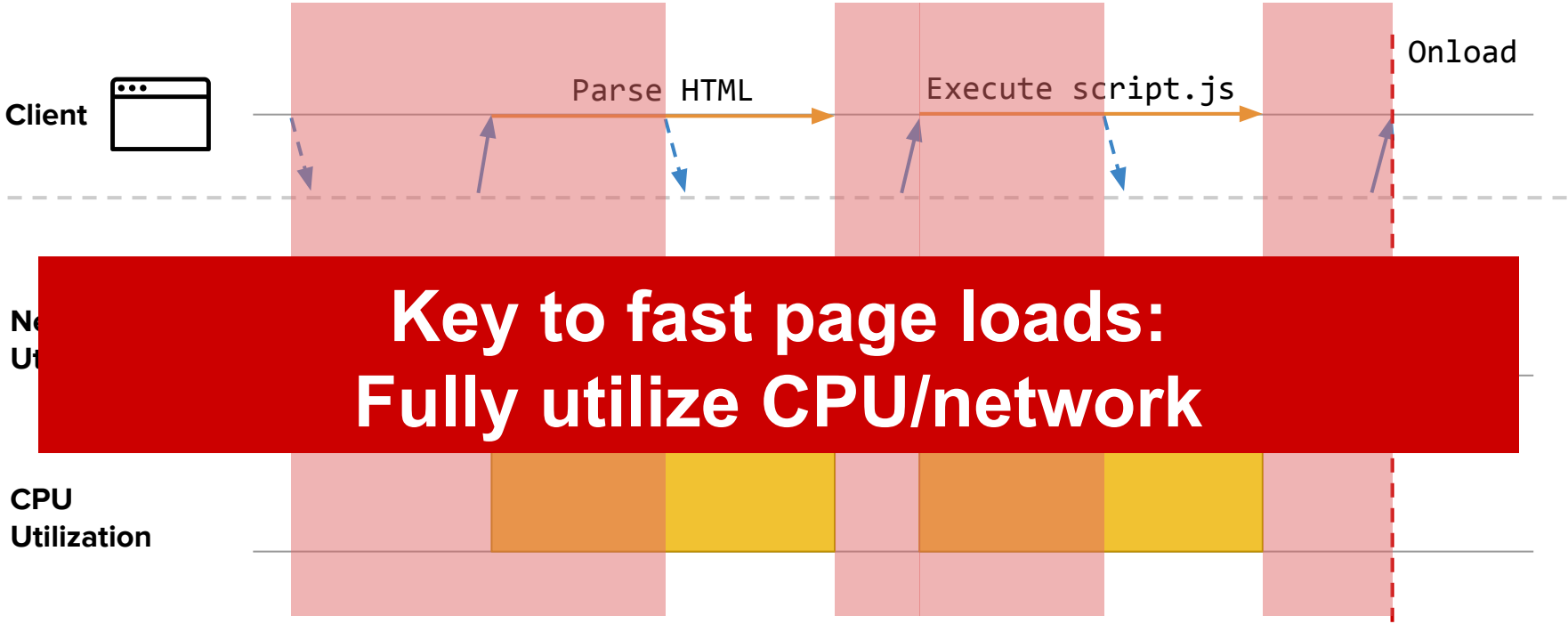


Loading a Web Page

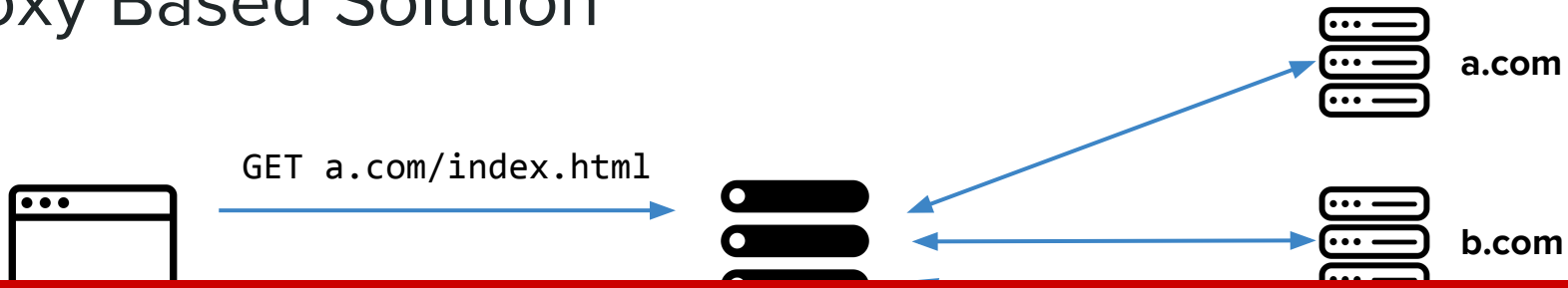




Waiting on CPU blocks network and vice versa



Proxy Based Solution



Web servers must aid client in discovering resources

- Client must trust HTTPS content pushed by proxy
- Proxy needs access to user's cookies for all domains

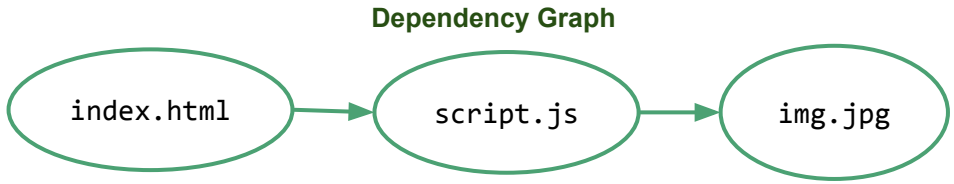
Challenges to approach

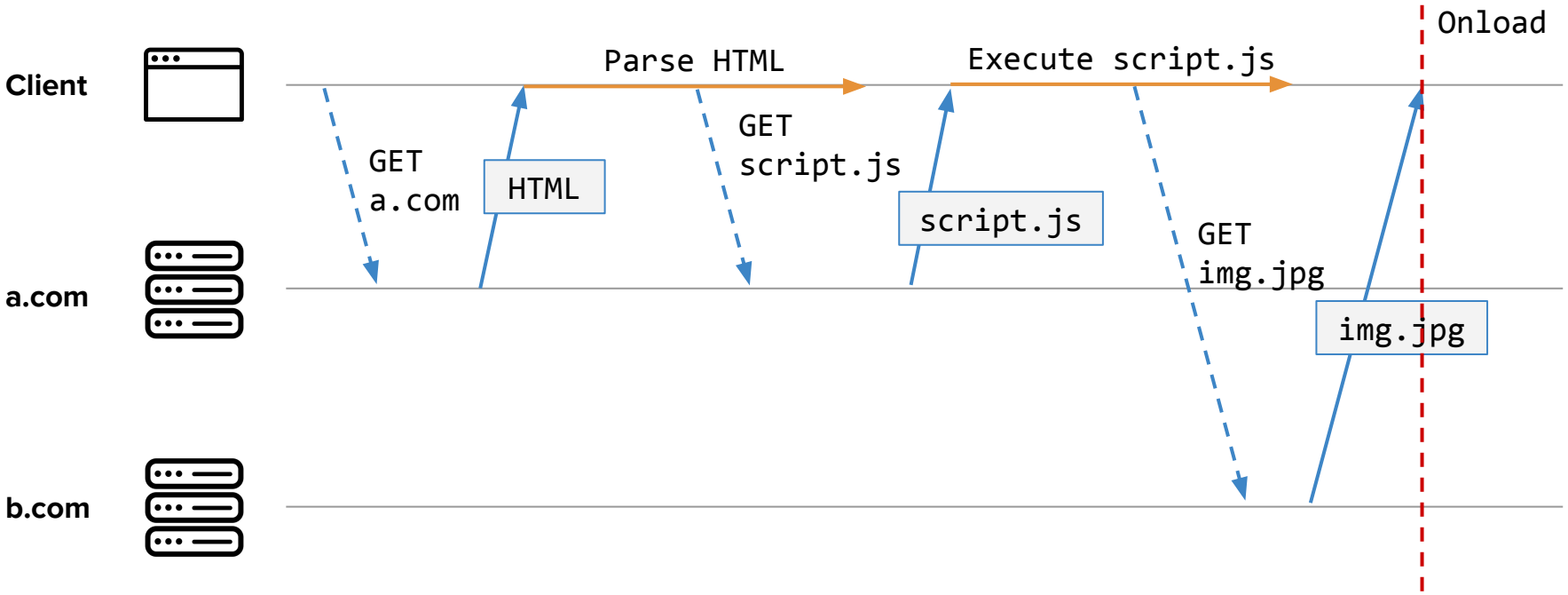
1. How can web servers discover dependencies?
2. How do web servers inform clients of discovered dependencies?
3. How should clients use input from servers?

Challenges to approach

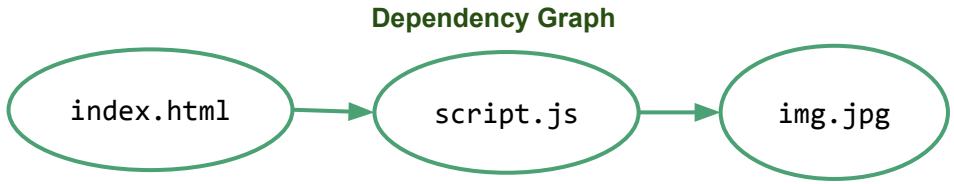
1. How can web servers discover dependencies?
2. **How do web servers inform clients of discovered dependencies?**
3. How should clients use input from servers?

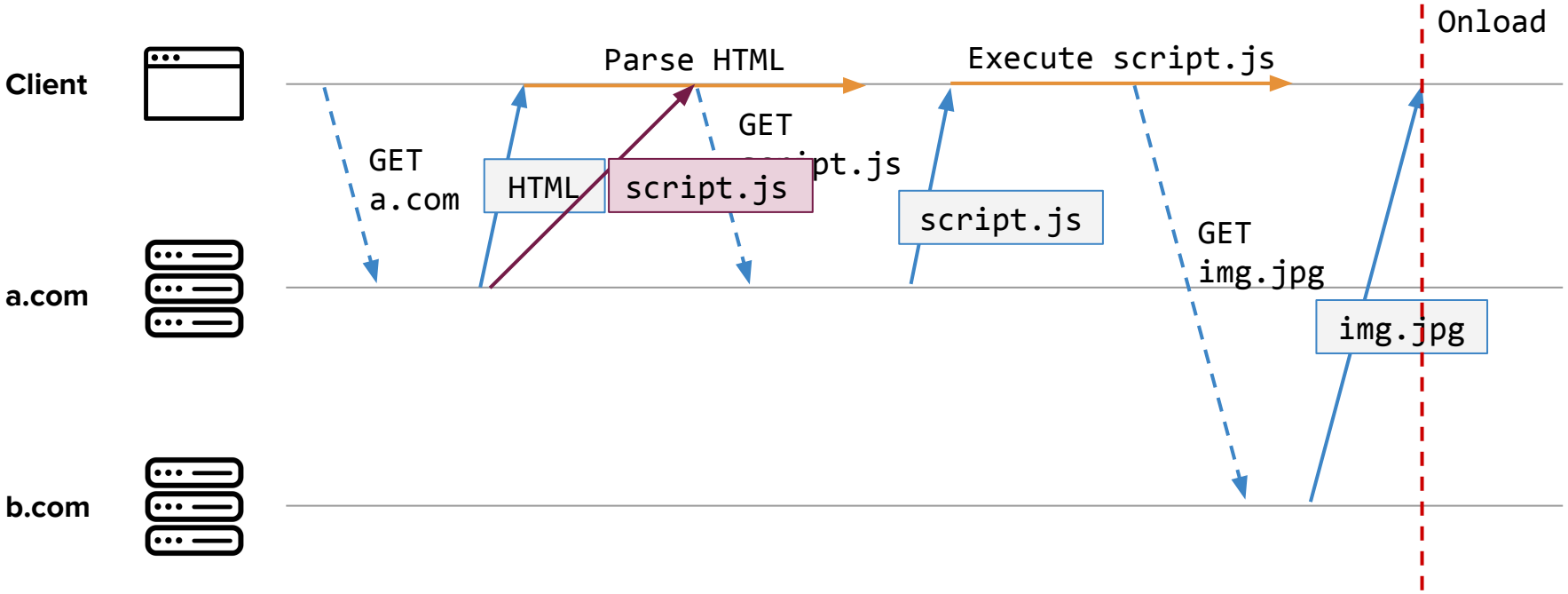
Inefficient Page Load



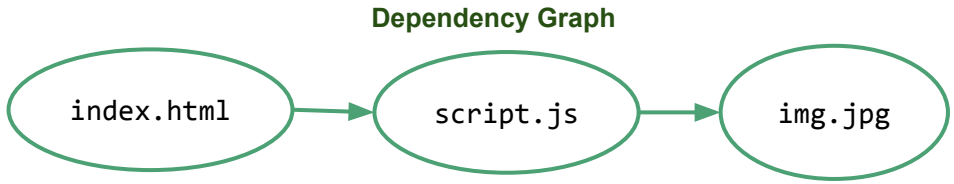


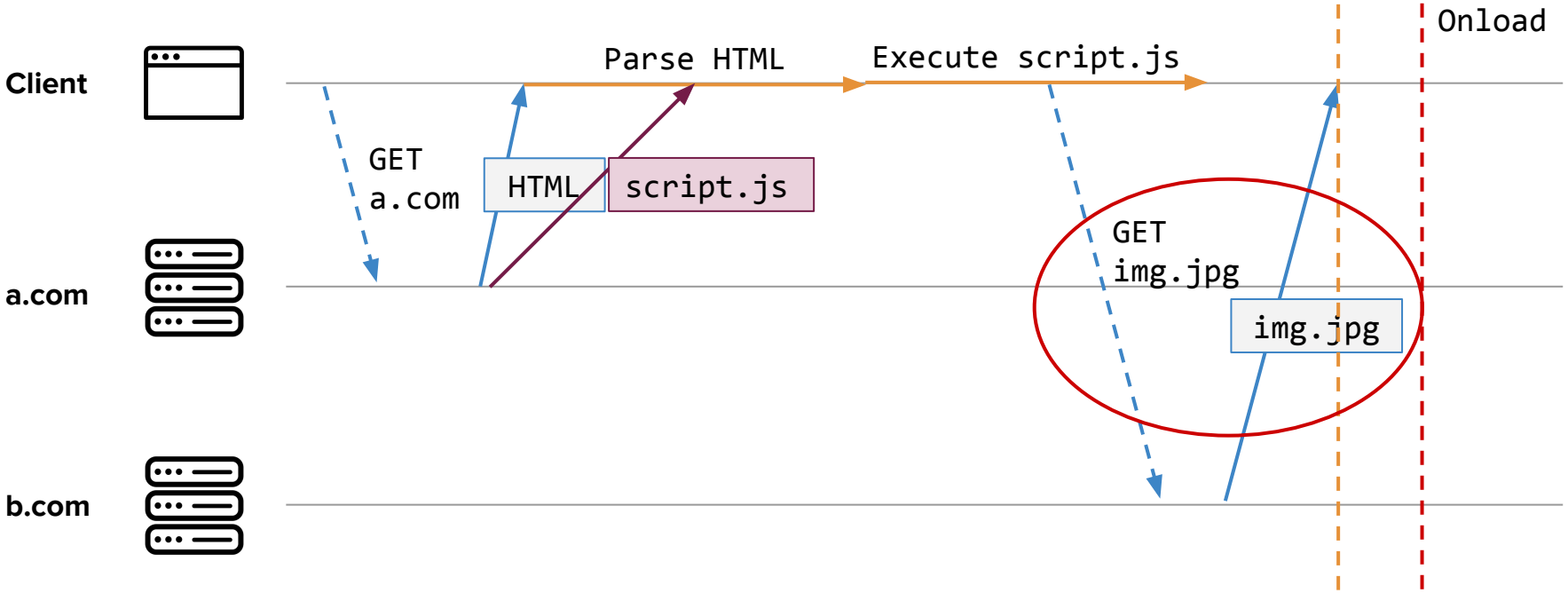
HTTP/2 Push


Dependency Graph
index.html
script.js
img.jpg

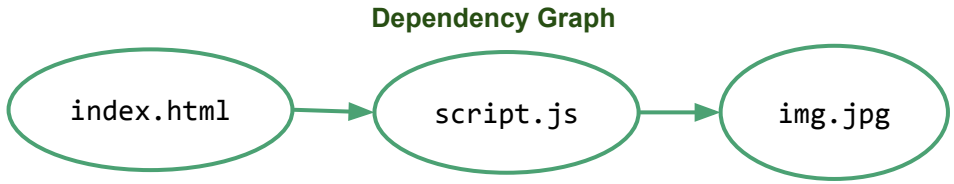

Client
a.com
b.com
Parse HTML
Execute script.js
Onload
GET
a.com
HTML
script.js
GET
script.js
GET
img.jpg
img.jpg

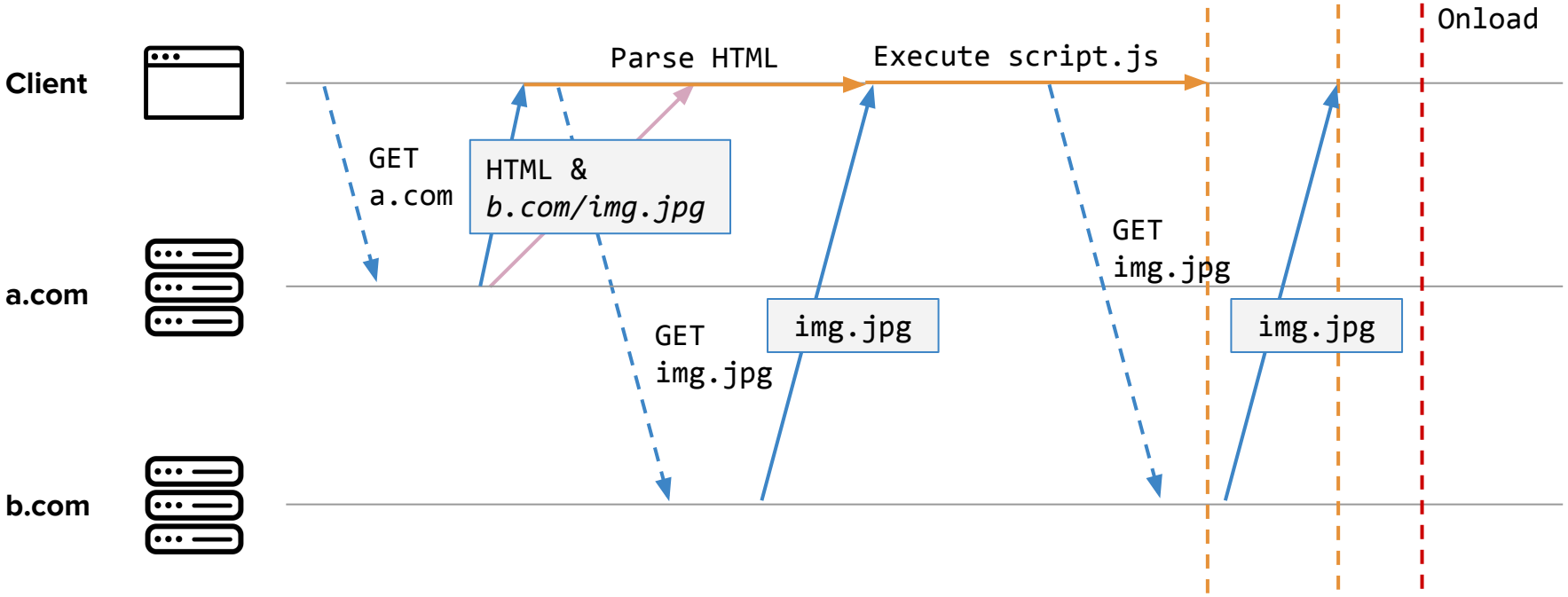
Push Only Load





Dependency Hints





Challenges to approach

1. **How can web servers discover dependencies?**

2. How do web servers inform clients of discovered dependencies?
   - *HTTP/2 Push + Dependency Hints*

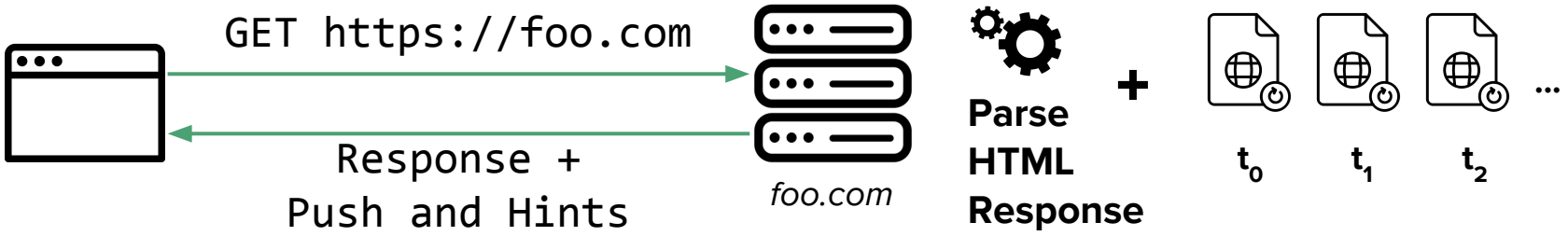
3. How should clients use input from servers?

Strawman Dependency Resolution

Drawbacks

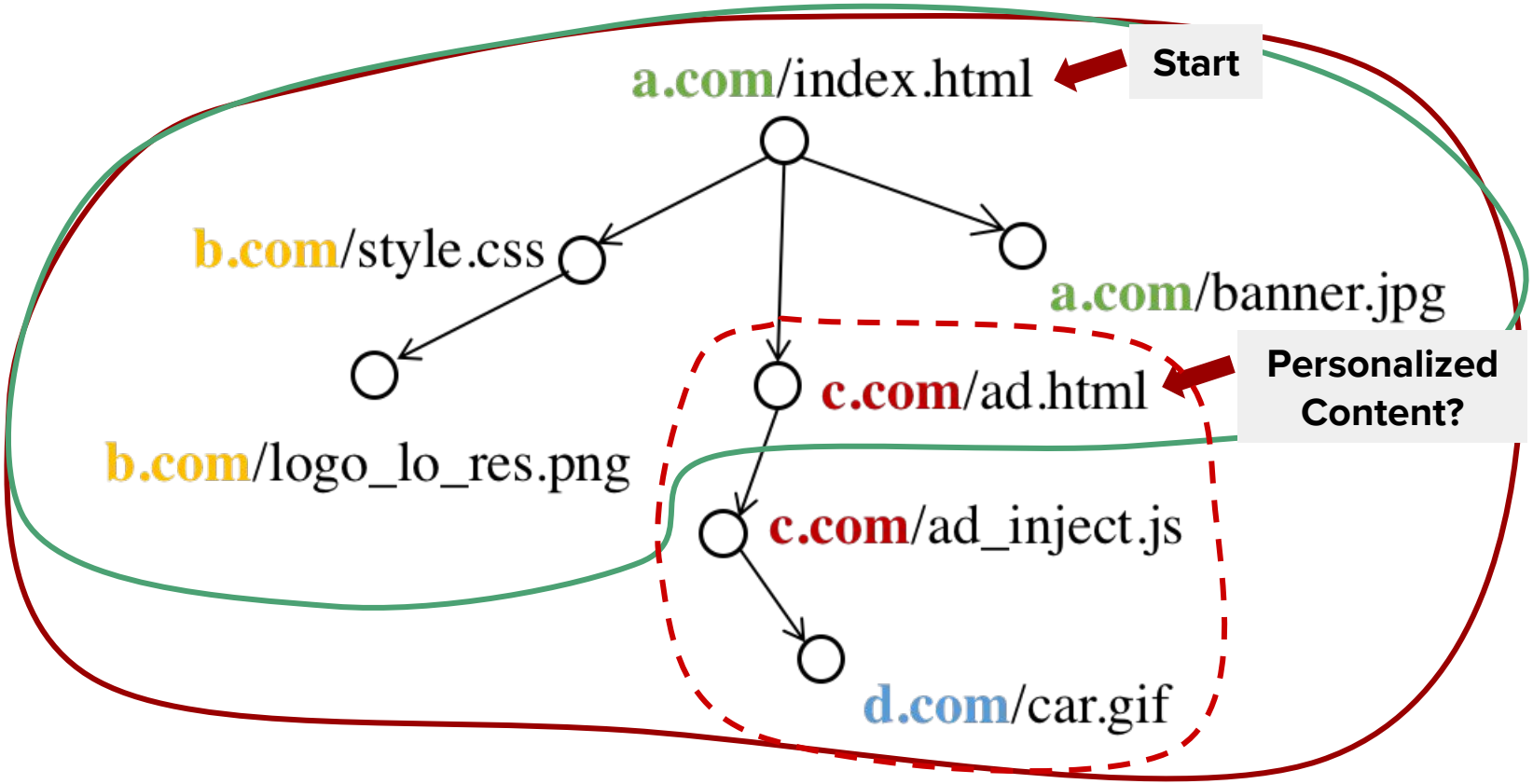
- Back-to-back loads differ
- Server cannot account for personalization

Combined Offline-Online Discovery



- **Stable dependencies:** Intersection of offline loads
- **Dynamic Content:** Online Parsing of HTML

Personalized Dependencies from Third-party Domains



Challenges to approach

1. How do web servers discover dependencies?
   - *Combine offline and online + Defer to third parties*

2. How do web servers inform clients of discovered dependencies?
   - *HTTP/2 Push + Dependency Hints*

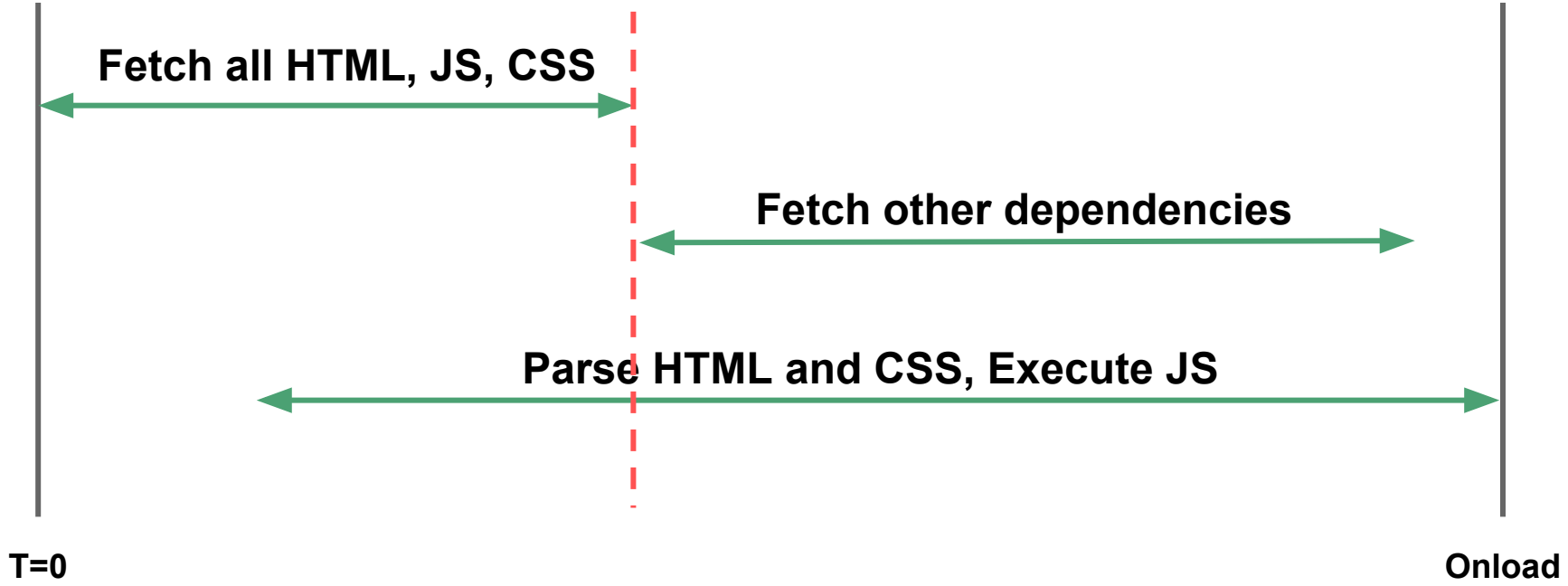
3. **How do clients use input from servers?**

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
  - Account for 20% of bytes on average

Vroom scheduler in action



Results overview

- **Accuracy of dependency discovery**
  - Median: 0% false positives and < 5% false negatives

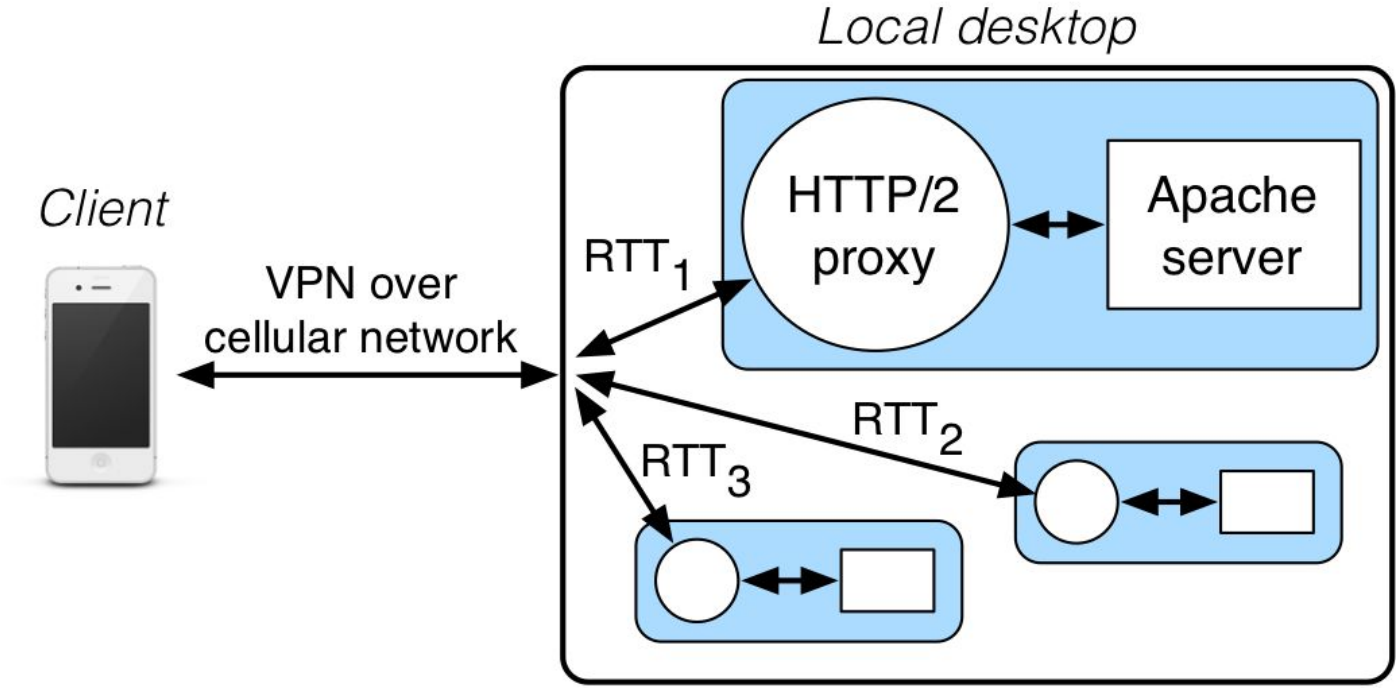
- **Improvements in client perceived performance**
  - Speedup over status quo
  - Simple strawmans don't suffice
  - Speedup even with warm caches

Results overview

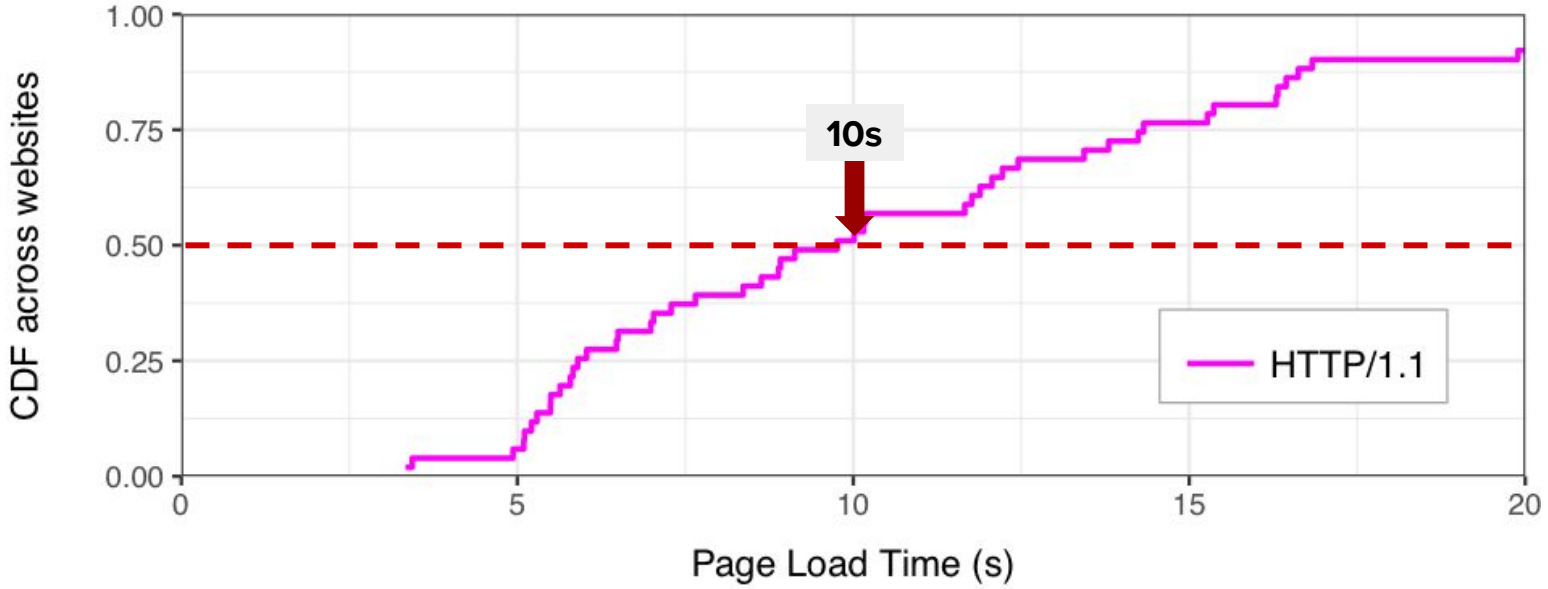
- **Accuracy of dependency discovery**
  - Median: 0% false positives and < 5% false negatives

- **Improvements in client perceived performance**
  - *Speedup over status quo*
  - Simple strawmans don't suffice
  - Speedup even with warm caches

Evaluation Setup

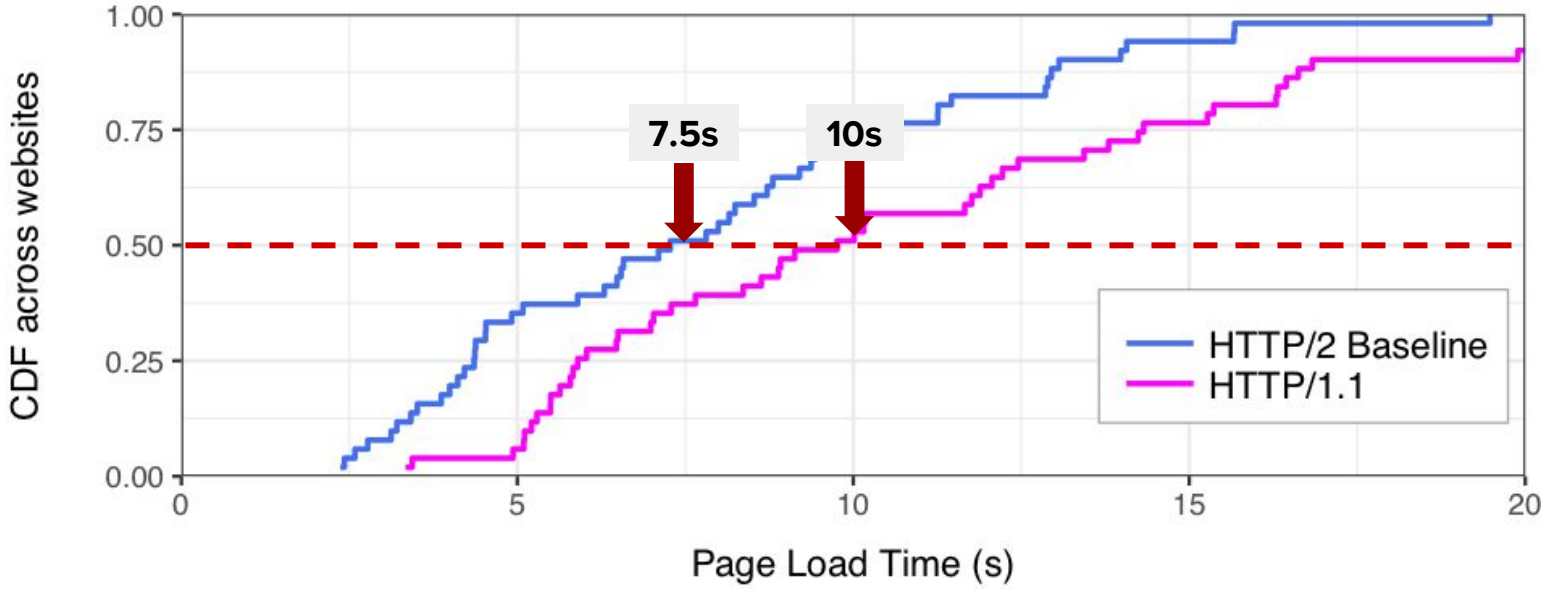


Vroom makes page loads much faster

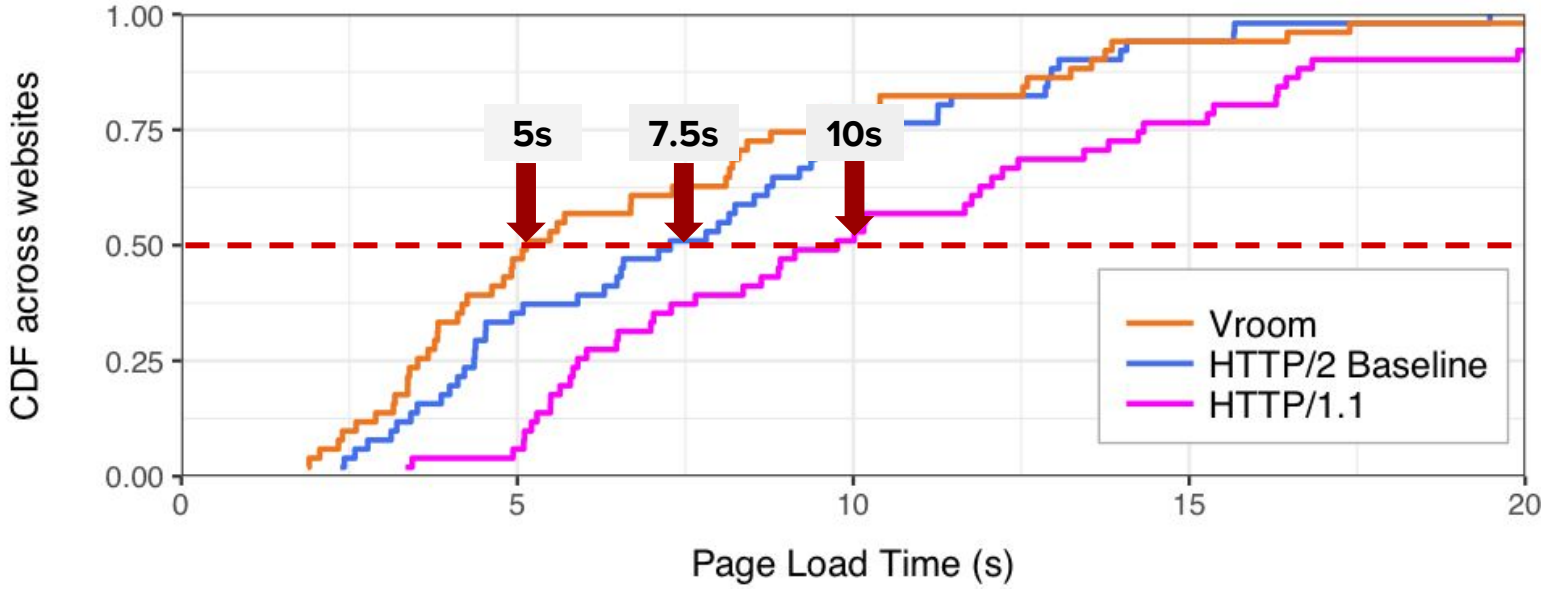


Alexa top 50 news and 50 sports sites

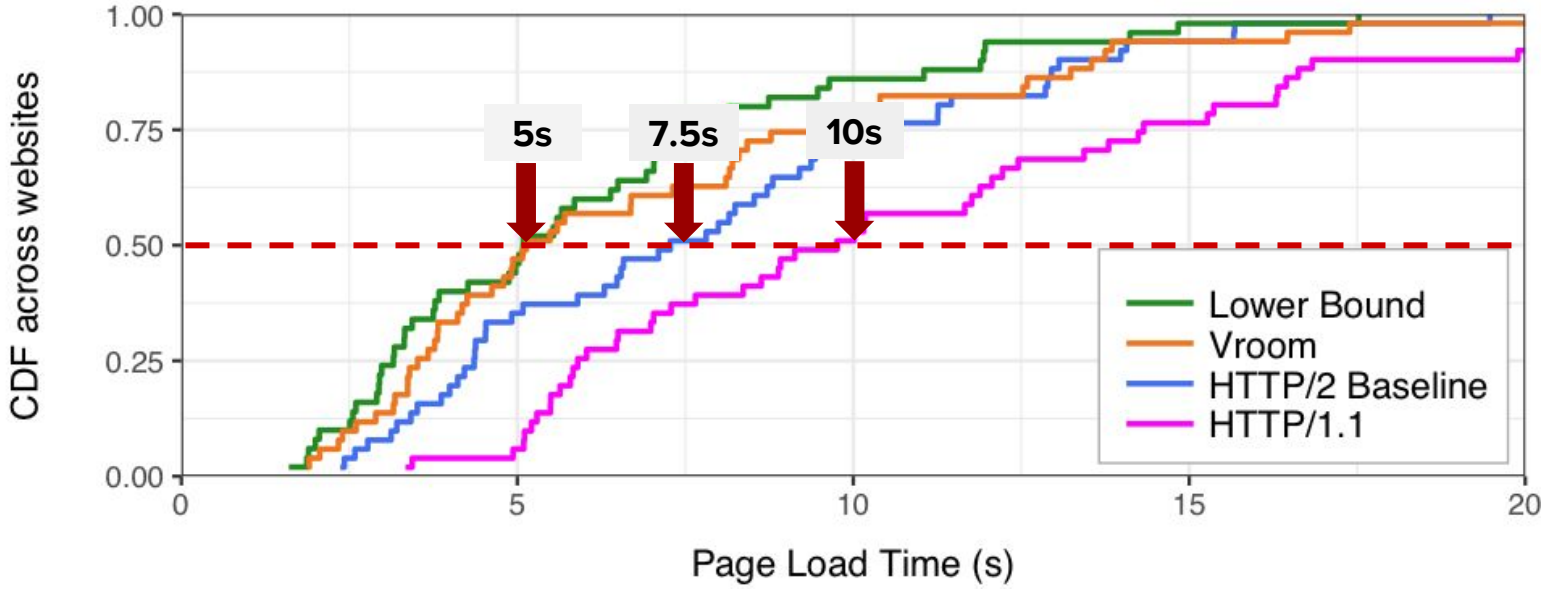
Vroom makes page loads much faster



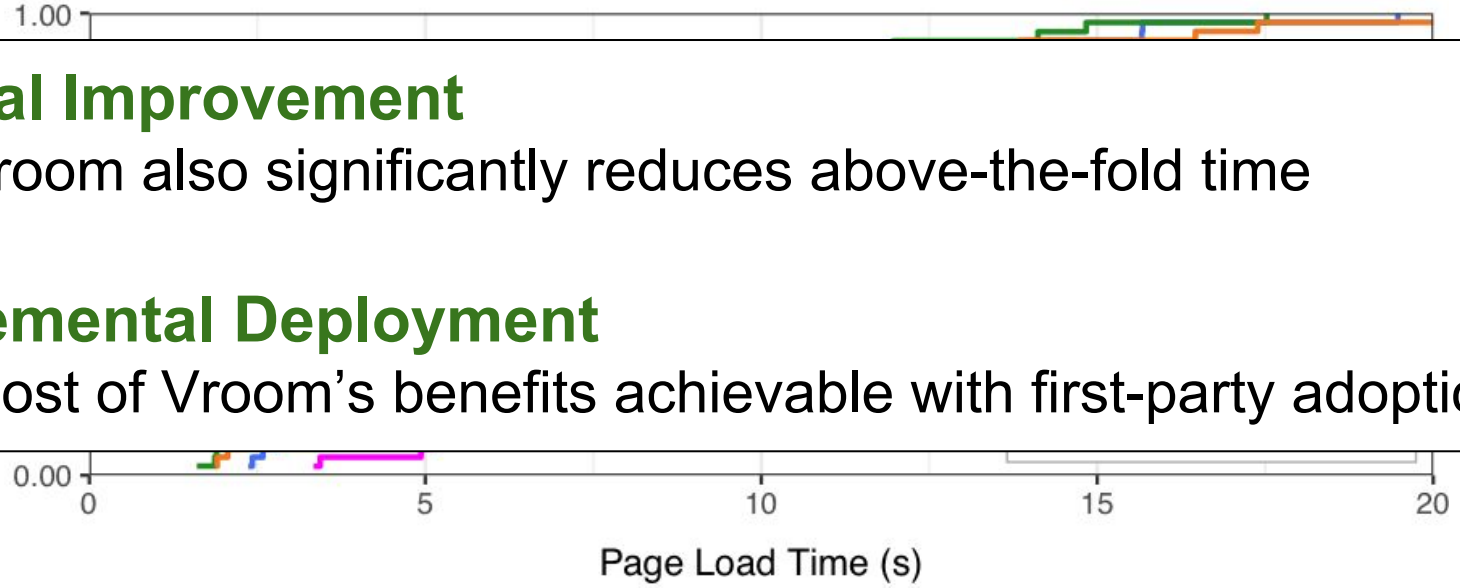
Vroom makes page loads much faster



Vroom makes page loads much faster



Vroom makes page loads much faster



Visual Improvement

- Vroom also significantly reduces above-the-fold time

Incremental Deployment

- Most of Vroom's benefits achievable with first-party adoption

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