Caching Doesn’t Improve Mobile Web Performance*

Jamshed Vesuna†  Colin Scott†
Michael BuettnerΔ  Michael PiatekΔ
Arvind Krishnamurthy*  Scott Shenker†‡

†UC Berkeley  ΔGoogle  *University of Washington  ‡ICSI

Special thanks to our shepherd Dan Tsafrir
Flywheel NSDI’15 Results

Increasing the cache hit ratio of their proxy from 22% to 32% resulted in only 1-2% reduction in median mobile page load time.
Goal:

Understand the effects of caching on mobile web performance
Outline

- Motivation
- Background
- Model (Estimating Page Load Time)
- Methodology for empirical results
- Corroborating model with empirical results
- Conclusion
Background - Loading a Web Page

**Diagram:**
- **Internet**
- **Object Loading**
- **HTML Parsing**
- **Rendering**
- **Cache**
- **Evaluation**
- **DOM**

User Interface
Background - Critical Path

Critical Path: the longest chain of dependent browser tasks
Fetch Delay = Network Delay
Render Delay = Computational Delay
Background - Page Load Time (PLT)
Outline

- Motivation
- Background
- **Model (Estimating Page Load Time)**
- Methodology for empirical results
- Corroborating model with empirical results
- Conclusion
Performance Model - Estimating PLT

\[ E_{PLT}[X] = C + N \cdot (1 - K \cdot X) - f(X) \]

- **C** - computational delays
- **N** - network delays
- **K** - fraction of objects on the critical path that are cacheable
- **X** - cache hit ratio (out of all objects)
- **f()** - overlap of **C** and **N** on the critical path
Performance Model - Building an Intuition

$E_{PLT}[X] = C + N \cdot (1 - K \cdot X)$

- Cold cache ($X = 0$):
  - Original Page Load Time = $C + N$
- Perfect cache for a “perfectly cacheable page”
  - $X = 1, K = 1$
  - Strict upper bound on improved page load time:
    - $E_{PLT}[1] = C$
In practice, $K \sim 0.2 = \frac{1}{5}$

$$E_{PLT}[\text{max}] \leq C + \frac{4}{5}N$$
Prediction: Upper Bound on Caching Benefits

\[ C : N \sim \frac{2}{3} \text{ for mobile devices} \]

\[ PLT^0 = E_{PLT}[0] \leq C + N = \frac{5}{2} C \]
\[ E_{PLT}[\text{max}] \leq \frac{11}{5} C \]

Reduction in PLT: \( \frac{E_{PLT}[X] - PLT^0}{PLT^0} \)
\[ \leq \frac{3}{25} \text{ (12% with a perfect cache!)} \]
Prediction: Desktop Benefits from Caching

\[ C : N \sim \frac{1}{6} \text{ for fast desktop devices} \]

\[ PLT^0 = E_{PLT}[0] \leq C + N = 7 \ C \]
\[ E_{PLT}[\text{max}] \leq \frac{21}{5} \ C \]

Reduction in PLT: \( \frac{(E_{PLT}[X] - PLT^0)}{PLT^0} \leq \frac{2}{5} \) (40% with a perfect cache!)
Explanation: C is Small for Desktop

\[ C : N \sim \frac{1}{5} \text{ for } 2\text{GHz CPU} \]

*Demystifying Page Load Performance with WProf. NSDI ’13*
Explanation: C is Small for Desktop

\[ C:N \sim \frac{1}{5} \text{ for } 2\text{GHz CPU}^* \]

*Demystifying Page Load Performance with WProf. NSDI '13
Explanation: C is Small for Desktop

\[ C : N \sim \frac{1}{5} \text{ for 2GHz CPU}^* \]

*Demystifying Page Load Performance with WProf. NSDI '13
Explanation: C is Small for Desktop

\[ C:N \sim \frac{1}{5} \text{ for 2GHz CPU}^* \]

*Demystifying Page Load Performance with WProf. NSDI ’13
Explanation: C is Larger for Mobile

\[ C:N \sim \frac{2}{3} \text{ for } 1\text{GHz CPU} \]
Outline

● Motivation
● Background
● Model (Estimating Page Load Time)
● Methodology for empirical results
● Corroborating model with empirical results
● Conclusion
Measurement Methodology

(a) Web Page Replay

(b) WPR Archive

(c) Telemetry

(d) Desktop Browser

(e) Tablet Browser
Measurement Methodology
Measurement Methodology

1. Record the original page
Measurement Methodology

1. Record the original page
Measurement Methodology

1. Record the original page
2. Then, replay with:
   a. With a “perfect cache”
   b. Or a “partial cache”
Measurement Methodology

1. Record the original page
2. Then, replay with:
   a. With a “perfect cache”
   b. Or a “partial cache”
3. Repeat
Outline

● Motivation
● Background
● Model (Estimating Page Load Time)
● Methodology for empirical results
● Corroborating model with empirical results
● Conclusion
Workload Characteristics

CDF of web pages

Fraction of bytes that are cacheable

CDF of web pages

Total response (MB)
Workload Characteristics

CDF of web pages

Fraction of bytes that are cacheable

CDF of web pages

Total response (MB)
Workload Characteristics

CDF of web pages vs. Fraction of bytes that are cacheable

CDF of web pages vs. Total response (MB)
Increasing Cache Hits - Flywheel Result

Increased cache hit ratio from 20% to 30% → 1-2% reduction in page load time
Desktop vs Mobile, Perfect Cache

Reduction Defined As:
(Original PLT - PLT with a perfect cache) / (Original PLT)
Desktop vs Mobile, Perfect Cache

Median reduction in PLT for 3.2 GHz desktop is 34%
Desktop vs Mobile, Perfect Cache

Median reduction in PLT for mobile is 13%
Isolating the Bottleneck Resource

Constrained CPU similar to Mobile
Isolating the Bottleneck Resource

Constrained RAM similar to Desktop
Isolating the Bottleneck Resource

CPU is the key difference, not RAM
Slower CPUs Show Reduced Improvements

As CPU is throttled, caching has a reduced impact on PLT
Slower CPUs Show Reduced Improvements

As CPU is throttled, caching has a reduced impact on PLT
Caching Benefits are Limited by Slow CPUs

- We know: slower CPUs increase computational delays (C)
- For desktop, network delay (N) dominates (C)
- For mobile*, network delay (N) is comparable to (C) (3:2)
- Caching only reduces (N)

→ Mobile devices benefit less from web caching

*Assumption: “All else being equal” (including b/w)
Implications

● Content providers:
  ○ Stop paying for CDNs* [for mobile users]

● Analyze what’s on the critical path
  ○ Cache critical path items
  ○ Make use of SPDY or HTTP/2 prioritization levels

*If you only care about end user latency
Conclusion

- Caching doesn’t decrease mobile PLT much
  - Items on the critical path are often not cacheable*
  - CPU is the key bottleneck resource on mobile
- Key contribution: predictive performance model

jamshed.vesuna@gmail.com        cs@cs.berkeley.edu

This Presentation: https://goo.gl/plH4HE
PLT Analysis: https://github.com/colin-scott/page_load_time
Open Source Tools: https://github.com/JamshedVesuna/telemetry

*Demystifying Page Load Performance with WProf. NSDI ’13