Understanding Mobile Web
Browser Performance

Rajiv Vijayakumar
Qualcomm Incorporated
Topics

- Mobile browser architecture
- Mobile network latencies
- Android HTTP: Parallel downloads and caching
- Power consumption
SmartPhone Web Browsers
What is a “WebKit” Browser?
Browser Architecture
Browser Architecture

Layout Engine
Browser Architecture

WebKit
Browser Architecture

WebKit

JavaScript Engine
Browser Architecture

WebKit

JavaScriptCore or V8
Browser Architecture

WebKit

JavaScriptCore or V8

Rendering

Networking
Browser Architecture

- Browser UI
- WebKit
- Rendering
- JavaScriptCore or V8
- Networking
Android
Android Browser ≠ Chrome
<img src="http://www.qualcomm.com/imgA.jpg">
DNS

Time
DNS  TCP
<img src="http://www.qualcomm.com/imgA.jpg">
DNS  TCP  HTTP

GET /imgA.jpg

HTTP/1.1 200 OK
Latencies in 3G/4G Networks
Latencies in 3G/4G Networks

- Initial connection setup: > 1s (on older 3G)
Latencies in 3G/4G Networks

- Initial connection setup: > 1s (on older 3G)
- Round trip time (RTT): 80ms or higher (typically)
  - 100 - 200ms common
RTTs at speed of light

- San Diego – New York: ~26ms
- San Diego – Johannesburg: ~110ms
Every round trip is at least 80-100ms
Every round trip is at least 80-100ms

Including HTTP retrievals from a CDN
DNS

Every new lookup takes 1 RTT + Lookup Time

Histogram of DNS Lookup Times

Source: Data collected over 30 sites (280 domains) on Qualcomm corporate network
Domain Sharding

- Use of multiple domains (cdn1.domain.com, cdn2.domain.com)
- Requests may not happen in parallel...
Parallel Downloads

8 images across 2 domains

cdn1.domain.com/img1A.png
cdn1.domain.com/img1B.png
cdn1.domain.com/img1C.png
cdn1.domain.com/img1D.png

cdn2.domain.com/img2A.png
cdn2.domain.com/img2B.png
cdn2.domain.com/img2C.png
cdn2.domain.com/img2D.png
Browsers typically use 4-6 sockets per host

t=0
Android

![Diagram of Android components and CDNs](cdn1)

![Diagram of Android components and CDNs](cdn2)
Android

cdn1

img1A -> img1B -> img1C -> img1D

cdn2

img2A -> img2B -> img2C -> img2D
Android

cdn1

- img1A
- img1B
- img1C

cdn2

- img2A
Android

- “Thread per socket” model
- 4 HTTP threads
### Dalvik Debug Monitor (DDMS)

#### Threads

<table>
<thead>
<tr>
<th>ID</th>
<th>Tid</th>
<th>Status</th>
<th>utime</th>
<th>stime</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>350</td>
<td>native</td>
<td>340</td>
<td>124</td>
<td>main</td>
</tr>
<tr>
<td>2</td>
<td>351</td>
<td>vmwait</td>
<td>9</td>
<td>150</td>
<td>HeapWorker</td>
</tr>
<tr>
<td>3</td>
<td>352</td>
<td>vmwait</td>
<td>9</td>
<td>1</td>
<td>GC</td>
</tr>
<tr>
<td>4</td>
<td>353</td>
<td>vmwait</td>
<td>0</td>
<td>0</td>
<td>Signal Catcher</td>
</tr>
<tr>
<td>5</td>
<td>354</td>
<td>running</td>
<td>2</td>
<td>26</td>
<td>JDWP</td>
</tr>
<tr>
<td>6</td>
<td>355</td>
<td>vmwait</td>
<td>19</td>
<td>28</td>
<td>Compiler</td>
</tr>
<tr>
<td>7</td>
<td>356</td>
<td>native</td>
<td>0</td>
<td>0</td>
<td>Binder Thread #1</td>
</tr>
<tr>
<td>8</td>
<td>357</td>
<td>native</td>
<td>0</td>
<td>0</td>
<td>Binder Thread #2</td>
</tr>
<tr>
<td>9</td>
<td>359</td>
<td>native</td>
<td>4</td>
<td>12</td>
<td>CookieSyncManager</td>
</tr>
<tr>
<td>10</td>
<td>361</td>
<td>wait</td>
<td>0</td>
<td>5</td>
<td>AsyncTask #1</td>
</tr>
<tr>
<td>11</td>
<td>363</td>
<td>native</td>
<td>118</td>
<td>135</td>
<td>WebViewCoreThread</td>
</tr>
<tr>
<td>12</td>
<td>364</td>
<td>wait</td>
<td>0</td>
<td>7</td>
<td>AsyncTask #2</td>
</tr>
<tr>
<td>13</td>
<td>365</td>
<td>wait</td>
<td>0</td>
<td>4</td>
<td>AsyncTask #3</td>
</tr>
<tr>
<td>14</td>
<td>366</td>
<td>native</td>
<td>3</td>
<td>4</td>
<td>Thread-17</td>
</tr>
<tr>
<td>15</td>
<td>367</td>
<td>wait</td>
<td>4</td>
<td>8</td>
<td>http0</td>
</tr>
<tr>
<td>16</td>
<td>368</td>
<td>wait</td>
<td>7</td>
<td>6</td>
<td>http1</td>
</tr>
<tr>
<td>17</td>
<td>369</td>
<td>wait</td>
<td>1</td>
<td>4</td>
<td>http2</td>
</tr>
<tr>
<td>18</td>
<td>370</td>
<td>wait</td>
<td>1</td>
<td>0</td>
<td>http3</td>
</tr>
<tr>
<td>19</td>
<td>371</td>
<td>native</td>
<td>29</td>
<td>34</td>
<td>WebViewWorkerThread</td>
</tr>
<tr>
<td>20</td>
<td>372</td>
<td>wait</td>
<td>0</td>
<td>1</td>
<td>AsyncTask #4</td>
</tr>
<tr>
<td>21</td>
<td>373</td>
<td>wait</td>
<td>0</td>
<td>2</td>
<td>AsyncTask #5</td>
</tr>
</tbody>
</table>

#### Source

Source: DDMS (from Android SDK) connected to Android 2.3.3 emulator
Android

- “Thread per socket” model
- 4 HTTP threads
Android

- "Thread per socket" model
- 4 HTTP threads
- Only 4 sockets can have outstanding HTTP requests ("active")
Android
Android

Pipelining
Android

- Send multiple requests without waiting for each response
  - Responses must arrive in order

Pipelining
Android

- Send multiple requests without waiting for each response
  - Responses must arrive in order
- Android pipelines up to 3 requests on each socket
Pipelining

- Not supported, or not enabled by default on most major browsers

- Historically caused problems (e.g. due to badly behaving proxies)

- Millions of Android devices actively using pipelining
  - Time to revisit...?
HTTP Caching
HTTP Caching

- WebKit maintains a memory cache
  - Default size = 8 MB

- Android has a persistent (file system) HTTP cache
  - Default size = 6 MB
Cache Eviction
Cache Eviction

- What gets evicted to make room?
Cache Eviction

- What gets evicted to make room?

- Android policy: Prioritize base on expiration date
  - Far future expiration dates have higher priority
  - Evict lowest priority items first

Expires: Wed, 09 Jun 2021 21:12:36 GMT
Example of Cache Contents

Visited several sites, read blog posts and linked articles etc.
Example of Cache Contents

Visited several sites, read blog posts and linked articles etc.

sqlite3 webviewCache.db 'select expires from cache order by expires desc;'
Example of Cache Contents

Visited several sites, read blog posts and linked articles etc.

```
sqlite3 webviewCache.db 'select expires from cache order by expires desc;'
```

<table>
<thead>
<tr>
<th>Expiration Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>2036</td>
</tr>
<tr>
<td>2035</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>2011</td>
</tr>
</tbody>
</table>
Example of Cache Contents

Visited several sites, read blog posts and linked articles etc.

```sql
sqlite3 webviewCache.db 'select expires from cache order by expires desc;'
```
- Cache eviction policy affects performance
- Consider real-world cache occupancy when measuring cached page load performance
Caveats

- Devices can (and do!) differ from stock Android

- Differences in:
  - Number of HTTP threads
  - Cache size and eviction algorithm
  - etc.
What is the power drawn from the battery while loading a web page?
Battery Current (mA)

Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network
Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network

- Baseline (200 mA)
- 11s download (400 mA)
- 10s idle (300 mA)
Power

- Any network traffic brings up the radio for several seconds
  - Network dormancy timer: typically 10-15 seconds
- Use long periods for any periodic Ajax calls etc.

Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network
TCP FINs and Power

Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network
TCP FINs and Power

- Android closes TCP sockets after ~6s of inactivity
  - Sockets get closed at different times

Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network
TCP FINs and Power

- Android closes TCP sockets after ~6s of inactivity
  - Sockets get closed at different times
- Every socket close generates a TCP FIN
  - Extends the dormancy timer
  - Keeps the modem on for a 1-bit communication

Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network
TCP FINs and Power

- Android closes TCP sockets after ~6s of inactivity
  - Sockets get closed at different times
- Every socket close generates a TCP FIN
  - Extends the dormancy timer
  - Keeps the modem on for a 1-bit communication
- Optimization: close all sockets after page load completes

Source: Power measurements made on MDP MSM8655™ running Android 2.3 over commercial 3G network
JavaScript Performance

- SunSpider benchmark numbers have improved dramatically in recent years
  - Android (Jun 2011) < 2.5s
  - Desktop Chrome (Sep 2008) ~2.8 s

Source: http://ejohn.org/blog/javascript-performance-rundown/
DEVELOP MOBILE WEB APPS THAT ARE DESIGNED TO RUN LIKE NATIVE APPS

Get ready for a few realities about developing for the mobile web:

- **Your users want the kind of rich multimedia content and far-reaching applications that rely on the browser and JavaScript.**
- **Rich content and complex Web applications also rely on hardware resources deep inside the mobile device.**
- **You need to give your Web users desktop-quality performance on mobile devices before your competitors do.**

To make this easier for you, the Web Technologies initiative from Qualcomm Incorporated and Qualcomm Innovation Center, Inc. (QIC) enables a series of software features and hardware-tuned performance optimizations that give the Web application environment deep reach into the mobile device. The end result—a level of performance from your Web app that users typically expect from native applications and even desktop applications.
Q&A
Nothing in these materials is an offer to sell any of the components or devices referenced herein. Certain components for use in the U.S. are available only through licensed suppliers. Some components are not available for use in the U.S.