OPTIMIZING MEDIA QUERIES

Kimberly Blessing
@obiwankimberly
Director, Think Brownstone
Philadelphia, PA

Responsive Web Design Summit, April 16, 2013
DOES THE WAY WE WRITE MEDIA QUERIES IMPACT PAGE PERFORMANCE?

WHICH TECHNIQUE YIELDS THE BEST PERFORMANCE?
PART 1: TEST MEDIA QUERY SPECIFICITY

Media queries that use cascading expressions

```css
@media screen and (min-width: 600px) {
  body {color:black}
}
@media screen and (min-width: 768px) {
  body {margin:1em}
}
@media screen and (min-width: 975px) {
  body {background:url(…)}
}
```

Media queries bound to specific screen sizes

```css
@media screen and (min-width: 600px) and (max-width: 767px) {
  body {color:black}
}
@media screen and (min-width: 768px) and (max-width: 974px) {
  body {color:black; margin:1em}
}
@media screen and (min-width: 975px) {
  body {color:black; margin:1em; background:url(…)}
}
```
PART 2: TEST CODE STRUCTURE

@media blocks consolidated at end

/* All CSS rules for MVP/LCD */
@media screen and ...
    /* All for “tablet” */
@
@media screen and ...
    /* All for “desktop” */
@

@media blocks interspersed throughout

/* Component CSS for MVP/LCD */
@media screen and ...
    /* Component for “tablet” */
@
@media screen and ...
    /* Component for “desktop” */
@

MVP = minimum viable product
LCD = lowest common denominator

Optimizing Media Queries by @obiwankimberly, Responsive Web Design Summit, April 16, 2013
THE TEST SUBJECT: WEBLINC.COM

- Responsive design with three breakpoints, but no responsive images
- All tests run against production hardware, which is CDN-fronted and runs mod_pagespeed
- Tests run on Saturdays and Sundays around 3 AM (minimal traffic/load)
TESTING METHODOLOGY

Test Scenarios

1. CSS with no media queries (MVP/LCD)
2. Cascading media queries consolidated at end
3. Cascading media queries interspersed throughout
4. Targeted screen size media queries consolidated at end
5. Targeted screen size media queries interspersed throughout

Data Collection

- File size and code complexity metrics
- CSS profiling statistics, collected using Opera’s Developer Tools
- Page load times for home page (first and repeat views) on major desktop browsers and iPhone, collected with WebPageTest
## File Size and Code Complexity

<table>
<thead>
<tr>
<th></th>
<th>Minified (bytes)</th>
<th>Min + GZIP (bytes)</th>
<th># MQs</th>
<th># MQs applied</th>
<th># CSS rules applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MQs</td>
<td>22,226</td>
<td>5,467</td>
<td>0</td>
<td>0</td>
<td>251</td>
</tr>
<tr>
<td>Cascade/End</td>
<td>43,694</td>
<td>9,954</td>
<td>8</td>
<td>4</td>
<td>485</td>
</tr>
<tr>
<td>Cascade/Inter</td>
<td>44,434</td>
<td>9,298</td>
<td>28</td>
<td>21</td>
<td>485</td>
</tr>
<tr>
<td>Targeted/End</td>
<td>53,221</td>
<td>10,107</td>
<td>8</td>
<td>2</td>
<td>445</td>
</tr>
<tr>
<td>Targeted/Inter</td>
<td>54,152</td>
<td>9,477</td>
<td>29</td>
<td>13</td>
<td>445</td>
</tr>
</tbody>
</table>

* Number of media queries/rules applied at 1024px screen width
# CSS Profiling Statistics

<table>
<thead>
<tr>
<th>CSS Parsing</th>
<th>Reflow</th>
<th>Style Recalculation</th>
<th>Layout</th>
<th>Paint</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MQs</td>
<td>6.0</td>
<td>2.2</td>
<td>13.4</td>
<td>9.6</td>
</tr>
<tr>
<td>Cascade/End</td>
<td>8.0</td>
<td>3.8</td>
<td>17.6</td>
<td>16.8</td>
</tr>
<tr>
<td>Cascade/Inter</td>
<td>9.4</td>
<td>3.2</td>
<td>18.4</td>
<td>17.6</td>
</tr>
<tr>
<td>Targeted/End</td>
<td>9.6</td>
<td>2.8</td>
<td>16.8</td>
<td>17.2</td>
</tr>
<tr>
<td>Targeted/Inter</td>
<td>9.6</td>
<td>3.0</td>
<td>20.2</td>
<td>17.4</td>
</tr>
</tbody>
</table>

*Times in milliseconds*

Average of 5 test runs using Opera Developer Tools
## Page Load Times* (First View)

<table>
<thead>
<tr>
<th></th>
<th>IE 8</th>
<th>IE 9</th>
<th>IE 10</th>
<th>Firefox</th>
<th>Chrome</th>
<th>iPhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MQs</td>
<td>1.1</td>
<td>0.9</td>
<td>1.0</td>
<td>1.8</td>
<td>1.0</td>
<td>3.5</td>
</tr>
<tr>
<td>Cascade/End</td>
<td>1.0</td>
<td>0.8</td>
<td>1.0</td>
<td>1.9</td>
<td>1.2</td>
<td>4.8</td>
</tr>
<tr>
<td>Cascade/Inter</td>
<td>1.1</td>
<td>1.0</td>
<td>1.0</td>
<td>2.1</td>
<td>1.2</td>
<td>4.5</td>
</tr>
<tr>
<td>Targeted/End</td>
<td>1.2</td>
<td>1.3</td>
<td>1.0</td>
<td>1.9</td>
<td>1.1</td>
<td>4.7</td>
</tr>
<tr>
<td>Targeted/Inter</td>
<td>1.4</td>
<td>0.9</td>
<td>1.0</td>
<td>2.0</td>
<td>1.2</td>
<td>4.6</td>
</tr>
</tbody>
</table>

* Times in seconds

Average of 10 test runs with WebPageTest. Outliers have been removed.
PAGE LOAD TIMES (FIRST VIEW)

Optimizing Media Queries by @obiwankimberly, Responsive Web Design Summit, April 16, 2013
# Page Load Times* (Repeat View)

<table>
<thead>
<tr>
<th></th>
<th>IE 8</th>
<th>IE 9</th>
<th>IE 10</th>
<th>Firefox</th>
<th>Chrome</th>
<th>iPhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MQs</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>1.4</td>
<td>0.4</td>
<td>1.4</td>
</tr>
<tr>
<td>Cascade/End</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>1.4</td>
<td>0.5</td>
<td>1.7</td>
</tr>
<tr>
<td>Cascade/Inter</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>1.5</td>
<td>0.5</td>
<td>1.6</td>
</tr>
<tr>
<td>Targeted/End</td>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>1.3</td>
<td>0.4</td>
<td>1.7</td>
</tr>
<tr>
<td>Targeted/Inter</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>1.5</td>
<td>0.5</td>
<td>1.6</td>
</tr>
</tbody>
</table>

*Times in seconds*

Average of 10 test runs with WebPageTest. Outliers have been removed.
ANOTHER TEST: MICROSOFT.COM

- Responsive design with six(-ish) breakpoints and responsive images
- Tests run from my personal server with mod_pagespeed
- Tests run on a Saturday night while watching Doctor Who
- Copies of test files and data available at http://presentations.kimberlyblessing.com/2013/rwdsummit/
# Page Load Times* (First View)

<table>
<thead>
<tr>
<th></th>
<th>IE 8</th>
<th>IE 9</th>
<th>IE 10</th>
<th>Firefox</th>
<th>Chrome</th>
<th>iPhone</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MQs</td>
<td>4.4</td>
<td>4.0</td>
<td>2.6</td>
<td>2.8</td>
<td>2.8</td>
<td>13.7</td>
<td>15.2</td>
</tr>
<tr>
<td>Cascade/End</td>
<td>4.5</td>
<td>4.3</td>
<td>2.6</td>
<td>3.6</td>
<td>3.6</td>
<td>12.8</td>
<td>12.6</td>
</tr>
<tr>
<td>Cascade/Inter</td>
<td>4.2</td>
<td>3.7</td>
<td>2.7</td>
<td>3.5</td>
<td>3.5</td>
<td>13.1</td>
<td>13.7</td>
</tr>
<tr>
<td>Targeted/End</td>
<td>5.2</td>
<td>4.6</td>
<td>2.6</td>
<td>4.0</td>
<td>3.4</td>
<td>12.4</td>
<td>12.7</td>
</tr>
<tr>
<td>Targeted/Inter</td>
<td>4.6</td>
<td>4.0</td>
<td>2.7</td>
<td>3.5</td>
<td>3.0</td>
<td>12.6</td>
<td>13.5</td>
</tr>
</tbody>
</table>

*Times in seconds

Average of 10 test runs with WebPageTest. Outliers have been removed.
PAGE LOAD TIMES (FIRST VIEW)
# Page Load Times* (Repeat View)

<table>
<thead>
<tr>
<th></th>
<th>IE 8</th>
<th>IE 9</th>
<th>IE 10</th>
<th>Firefox</th>
<th>Chrome</th>
<th>iPhone</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MQs</td>
<td>1.4</td>
<td>1.3</td>
<td>0.9</td>
<td>1.6</td>
<td>1.1</td>
<td>5.4</td>
<td>9.4</td>
</tr>
<tr>
<td>Cascade/End</td>
<td>1.3</td>
<td>1.3</td>
<td>0.8</td>
<td>1.6</td>
<td>1.2</td>
<td>5.5</td>
<td>8.7</td>
</tr>
<tr>
<td>Cascade/Inter</td>
<td>1.3</td>
<td>1.3</td>
<td>0.9</td>
<td>1.6</td>
<td>1.3</td>
<td>5.6</td>
<td>8.5</td>
</tr>
<tr>
<td>Targeted/End</td>
<td>1.6</td>
<td>1.3</td>
<td>0.9</td>
<td>1.8</td>
<td>1.4</td>
<td>5.5</td>
<td>9.0</td>
</tr>
<tr>
<td>Targeted/Inter</td>
<td>1.4</td>
<td>1.4</td>
<td>0.9</td>
<td>1.5</td>
<td>1.2</td>
<td>5.6</td>
<td>8.5</td>
</tr>
</tbody>
</table>

*Times in seconds

Average of 10 test runs with WebPageTest. Outliers have been removed.
PAGE LOAD TIMES (REPEAT VIEW)

![Chart showing page load times for different browsers and devices, with categories such as Cascade/End, Cascade/Inter, Targeted/End, and Targeted/Inter.](chart-url)
YES, media query syntax and code structure have an impact on performance...

...however no particular technique stands as performance silver bullet. Writing optimized CSS overall is best!
CSS OPTIMIZATION RECOMMENDATIONS

Understand any resets and frameworks you use. Trim unused code and rewrite inefficient selectors or declarations.

Schedule periodic code reviews to prune unused code, based both on old content and old browsers.

Reevaluate your browser support matrix. Limit the use of browser hacks, polyfills, and prefixed properties.

Profile CSS selectors and optimize for right-to-left parsing.

Regularly test site performance, compare data before and after code changes to understand impact.
REQUIRED READING

Refloors & Repaints: CSS Performance making your JavaScript slow?
by stubbornella (Nicole Sullivan)
http://www.stubbornella.org/content/2009/03/27/refloors-repaints-css-performance-making-your-javascript-slow/

Profiling CSS for fun and profit. Optimization notes.
by kangax (Juriy Zaytsev)

Real World RWD Performance
by Guy Podjarny

And… follow updates from O’Reilly’s Velocity Web Performance and Operations Conference
http://velocityconf.com/