Optimizing JavaScript Runtime Performance for Touch

Stephen Woods
@ysaw
Native-Like Experience
The Mobile Web is Slow
What Happens at runtime?

- Fetch, format and Display Data
- Handle UI Events
- Move Pixels
Fetching Data
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Download</td>
<td>2.43 Mb/s</td>
<td>Upload</td>
<td>0.88 Mb/s</td>
</tr>
<tr>
<td>Ping</td>
<td>53 ms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Grade:** D+  
(Slower than 62% of US)

**Carrier:** Verizon Wireless  
**Server:** San Francisco, CA  
**Device:** iPhones
Caching: The cause of, and solution to, all of life's problems
$.get('data.json', dataCallback);
$.get('/', data.json', dataCallback);

myDAL.getData('key', callback);
myDAL.getData('key', callback);

def getData(key, callback) {
    if(cache[key]) return cache[key];

    $.get('/data.json', function(data){
        cache[key] = data;
        callback(data);
    });
}

window.localStorage['cache'] = JSON.stringify(cache);
TTL

Writes Invalidate Cache
Eviction

5mbs of localStorage

Strings stored as UTF-16 Characters

Maximum Storage space: 2.5million Characters
Least Recently Used
Computing the size of the Cache.

`JSON.stringify(localStorage).length`
Handling Events
Event handling is a conversation
If a gesture is in progress, the UI must be in motion.
Moving Pixels
30fps  = 33.3ms Per Frame
60fps  = 16.6ms Per Frame
Write-Only DOM
getOffsetHeight:
74% slower than get className

<table>
<thead>
<tr>
<th></th>
<th>Code</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOM write and read</td>
<td><code>f.style.border = '1px solid red'; var word = f.style.className;</code></td>
<td>20,372 ±0.96% fastest</td>
</tr>
<tr>
<td>className</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write and get offsetHeight</td>
<td><code>f.style.border = '1px solid red'; var word = f.offsetHeight;</code></td>
<td>5,271 ±1.68% 74% slower</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If you can, cache the value, not the node
var offsetCache = {};

function getOffsetHeight(id) {
  if (offsetCache[id]) {
    return offsetCache[node.id];
  } else {
    offsetCache[id] =
        document.getElementById(id).offsetHeight
  }
}
this.nodePositionMap[thisId] = {
    top: thisNode.getY(),
    height: thisNode.get('offsetHeight'),
    defer: photoList[i].defer ? {img_url: 'bar'}: false
};
background-image: 'foo.jpg';
background-size: 100%;
transform: translate3d(10, 10, 1);
Start up with:

- ♦ Finder
- ♦♦♦♦ MultiFinder

Upon startup, automatically open:

- ○ Selected Items
- ○ Opened Applications and DAs
- ○ MultiFinder Only

[Buttons: Cancel, OK]
User feedback comes first
isSwiping = 1;
function handleXHR(resp) {
  if(isSwiping) {
    setTimeout(function() {
      handleXHR(resp);
    }, 20);
  }
  processData(desp);
}
• Load image ~ 300ms (async)
• Find Faces ~ 100ms (sync)
• Compute Entropy Crop ~ 30ms (sync)
• Smooth Animation (sync)
Run Animation in a Separate Thread
slide1.style.transition = transitionDuration + 's transform ease-out';
slide1.style.transform = computedEndFrame;
Run Slow Code in a Worker*
Run Slow Code in a Worker*

*except on Android Browser
worker = new Worker('./smart-crop-worker.js');
UI Thread Handles Events and Orchestration
- Cache data where possible
- Update the UI on events, use transforms when possible
- Multitask when you can, both with threads and yeilding
@ysaw
stephenwoods.net
touch-interfaces.com
flickr.jobs

flickr.com/photos/39747297@N05/5230479916/
flickr.com/photos/pandameixiang/6784480227/
flickr.com/photos/ryfter/5196109310/
flickr.com/photos/howvin/4054392601/
flickr.com/photos/fdrlibrary/