JSMeter: Characterizing the Behavior of JavaScript Web Applications

Paruj Ratanaworabhan
Kasetsart University, Thailand

Ben Livshits and Ben Zorn
Microsoft Research, Redmond
JavaScript

One Word:

Standard for scripting web applications

Fast JITs widely available

Support in every browser

Lots of code present in all major web sites
Browser Wars!

ZDNet 29 May 2008

ghawks.net Dec. 2008
Artificial Benchmarks versus Real World Sites

7 V8 programs:
- richards
- deltable
- crypto
- raytrace
- earley-boyer
- regexp
- splay

8 SunSpider programs:
- 3-draytrace
- access-nbody
- bitops-nsieve
- controlflow

11 real sites:

Goals of JSMeter Project
- Instrument JavaScript execution and measure behavior
- Compare behavior of JavaScript benchmarks against real sites
- Consider how benchmarks can mislead design decisions
How We Measure JavaScript

\ie\script\*.cpp

Source-level instrumentation of Internet Explorer 8

custom jscript.dll

website visits

custom trace files

 Offline analyzers

custom trace files

Graph showing percentages of various operations such as Constant, Other Str Ops, and Concat Op.
Visiting the Real Sites

- Getting past page load performance
- Attempted to use each site in “normal” way:

<table>
<thead>
<tr>
<th>Site</th>
<th>Activity Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon</td>
<td>Search a book, add to shopping cart, sign in, and sign out</td>
</tr>
<tr>
<td>bing</td>
<td>Type in a search query and also look for images and news</td>
</tr>
<tr>
<td>bingmap</td>
<td>Search for a direction from one city to another</td>
</tr>
<tr>
<td>cnn</td>
<td>Read front page news</td>
</tr>
<tr>
<td>ebay</td>
<td>Search for a notebook, bid, sing in, and sign out</td>
</tr>
<tr>
<td>economist</td>
<td>Read front page news, view comments</td>
</tr>
<tr>
<td>facebook</td>
<td>Log in, visit a friend pages, browse through photos and comments</td>
</tr>
<tr>
<td>gmail</td>
<td>Sign in, check inbox, delete a mail, and sign out</td>
</tr>
<tr>
<td>google</td>
<td>Type in a search query and also look for images and news</td>
</tr>
<tr>
<td>googlemap</td>
<td>Search for a direction from one city to another</td>
</tr>
<tr>
<td>hotmail</td>
<td>Sign in, check inbox, delete a mail, and sign out</td>
</tr>
</tbody>
</table>
Understanding JavaScript Behavior

- Code
- JavaScript
- Events
- Objects
Code Behavior

- Function size
- Instructions/call
- Code locality
- Instruction mix
Total Bytes of JavaScript Source

Source size (kilobytes)

- **Real Sites**: Sites like google, hotmail, and peace.
- **V8**: Sites like crypto, raytrace, and ssl.
- **SunSpider**: Sites like 3d-raytrace and access-nobody.
function(a,b) {
    var i=0, elem, pos=a.length;
    if(D.browser.msie) {
        while(elem=b[i++])
            if(elem.nodeType!=8)
                a[pos++]=elem;
    } else
        while(elem=b[i++])
            a[pos++]=elem;
    return a
}
Fraction of Code Executed

Most code not executed
Object Allocation Behavior

- Allocation by types
- Live heap composition
- Lifetime distribution
Heap Data by Type

Few benchmarks allocate much data

Many functions
Rest are strings

Real Sites

V8

SunSpider
Functions grow steadily. GC reduces size of heap.
Heaps repeatedly created, discarded

Heap contains mostly functions

Heap drops to 0 on page load
2 Search Websites, 2 Architectures

- You stay on the same page during your entire visit
- Code loaded once
- Heap is bigger

- Every transition loads a new page
- Code loaded repeatedly
- Heap is smaller

Bing

Google
Event Handlers in JavaScript

- Number of events
- Sizes of handlers
Event-driven Programming Model

- JavaScript applications interact with a user

- Characteristics of JavaScript that differ from Java:
  - Single-threaded, non-preemptive event handlers
  - Example handlers: onabort, onclick, etc.
  - Very different from batch processing of benchmarks
  - Handler responsiveness critical to user experience
Total Events Handled

- Amazon
- Bing
- Bingmap
- CNN
- eBay
- Economist
- Facebook
- Gmail
- Google
- Googlemap
- Hotmail

Almost no events

Real Sites

V8
Median Bytecodes / Event Handled

- Code
- Objects
- Events

<table>
<thead>
<tr>
<th>Platform</th>
<th>Median Bytecodes</th>
<th>Event Handled</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bingmap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cnn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebay</td>
<td></td>
<td></td>
</tr>
<tr>
<td>economist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>facebook</td>
<td>506</td>
<td>2137</td>
</tr>
<tr>
<td>gmail</td>
<td></td>
<td></td>
</tr>
<tr>
<td>google</td>
<td></td>
<td></td>
</tr>
<tr>
<td>googlemap</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hotmail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Impact of Benchmarks

• What gets emphasis
  • Making tight loops fast
  • Optimizing small amounts of code

• Important issues ignored
  • Garbage collection (especially of strings)
  • Managing large amounts of code
  • Optimizing event handling
  • Considering JavaScript context between page loads
Related Work

- JavaScript
  - Richards, Lebresne, Burg, and Vitek (PLDI’10)
  - They draw similar conclusions
- Java
  - Doufour et al. (OOPSLA’03), Dieckmann and U. Hölzle (ECOOP’99)
- Other languages
  - C++: Calder et al. (JPL’95)
  - Interpreted languages: Romer et al. (ASPLOS’96)
JSMeter Conclusions

• JSMeter is an instrumentation framework
  • Used to measure and compare JavaScript applications
  • High-level views of behavior promote understanding
• JavaScript **is not Java or C#**
  • Integration with the DOM
  • Built-in support for asynchronous, distributed applications
• Benchmarks differ **significantly** from real sites
  • Misleads designers, skews implementations
• Next steps
  • Develop and promote **better benchmarks**
  • Promote **better performance tools** for JavaScript developers
  • **Design and evaluate better JavaScript runtimes** (e.g. IE9)
Additional Resources

- **Video**: *Project JSMeter: JavaScript Performance Analysis in the Real World* - MSDN Channel 9 interview with Erik Meier, Ben Livshits, and Ben Zorn
Microsoft Research
Faculty Summit 2010
Data Driven Software Engineering Track
Dynamic Languages and Browsers of the Future