State of JS
Implementations, 2014 Edition
webengineshackfest.org
Andy Wingo
Agenda

History

New things
A brief history of JS

1996-2008: slow
2014: fastish
A brief history of JS

1996-2008: slow
2014: fastish
Environmental forcing functions
Visiting a page == installing an app
Cruel latency requirements
JS: speed via dynamic proof

“Adaptive optimization”

A revival of compilation techniques pioneered by Smalltalk, Self, Strongtalk, Java

$expr$ ifTrue: $block$

Inlining key for performance: build sizable proof term

JS focus: low-latency adaptive optimization (fast start)

- lazy parsing and compilation
All about the tiers

“Method JIT compilers”; Java’s HotSpot is canonical comparison

The function is the unit of optimization

asm.js code can start in IonMonkey / Turbofan; embedded static proof pipeline
Optimizing compiler awash in information

Operand and result types
Free variable values
Global variable values
Sets of values: mono-, poly-, mega-morphic
Optimizations: An inventory

Inlining

Code motion: CSE, DCE, hoisting, sea-of-nodes

Specialization

- Numeric: int32, uint32, float, ...
- Object: Indexed slot access
- String: Cons, packed, pinned, ...

Allocation optimization: coalescing, scalar replacement, sinking

Register allocation
Dynamic proof, dynamic bailout

Compilation is proof-driven term specialization

Dynamic assertions: the future will be like the past

Dynamic assertion failure causes proof invalidation: abort ("bailout") to baseline tier

Bailout enables static compilation techniques (FTL)
A brief history of JS

1996-2008: slow
2014: fastish
...via adaptive optimization.
New things in 2014
New things in 2014
SM, JSC, V8
First perf, then features
SpiderMonkey perf

SM won Octane!

Landing of precise GC, then generational GC

https://blog.mozilla.org/javascript/2013/07/18/clawing-our-way-back-to-precision/

Compacting GC in the works

Lots of Ion work
JSC perf

(I am less knowledgeable here)

“Fourth-tier LLVM” (FTL) JIT
SpiderMonkey (Firefox)
- Interpreter
- Baseline
- IonMonkey

JavaScriptCore (WebKit, Safari)
- Interpreter
- Baseline
- DFG
- FTL

V8 (Chrome)
- Baseline
- Crankshaft/Turbofan
V8 perf

End of the road for Crankshaft

New thing: Turbofan

Fully typed IR, more capable of reliably inferring types over big asm.js programs

Sea-of-nodes approach transparently enables code motion

Status: enabled, but for asm.js code only
Features

ECMAScript 6 (ES6) was supposed to arrive this year, punted to next year, but all implementors involved in process

All engines are actively implementing ES6 features

JSC has implemented some features but not as focussed

<table>
<thead>
<tr>
<th></th>
<th>IE 10</th>
<th>IE 11</th>
<th>IE Technical Preview</th>
<th>FF 31</th>
<th>FF 34</th>
<th>FF 35</th>
<th>FF 36</th>
<th>CH 39, OP 26</th>
<th>CH 40, OP 27</th>
<th>SF 6</th>
<th>SF 7.0</th>
<th>SF 7.1, SF 8</th>
<th>WK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6%</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>21%</td>
<td>0/9</td>
<td>0/9</td>
<td>8/9</td>
<td>7/9</td>
<td>7/9</td>
<td>7/9</td>
<td>7/9</td>
<td>4/9</td>
<td>4/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
</tr>
<tr>
<td>46%</td>
<td>0/7</td>
<td>8/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
<td>5/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
</tr>
<tr>
<td>59%</td>
<td>0/5</td>
<td>0/5</td>
<td>3/5</td>
<td>3/5</td>
<td>3/5</td>
<td>3/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
</tr>
<tr>
<td>69%</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>53%</td>
<td>0/8</td>
<td>8/8</td>
<td>6/8</td>
<td>6/8</td>
<td>8/8</td>
<td>8/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>2/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
</tr>
<tr>
<td>56%</td>
<td>0/8</td>
<td>8/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
</tr>
<tr>
<td>10%</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
</tr>
<tr>
<td>10%</td>
<td>0/3</td>
<td>0/3</td>
<td>3/3</td>
<td>0/3</td>
<td>3/3</td>
<td>3/3</td>
<td>3/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>0/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>24%</td>
<td>0/4</td>
<td>0/4</td>
<td>0/4</td>
<td>3/4</td>
<td>3/4</td>
<td>4/4</td>
<td>4/4</td>
<td>0/4</td>
<td>0/4</td>
<td>1/4</td>
<td>0/4</td>
<td>0/4</td>
<td>0/4</td>
</tr>
<tr>
<td>28%</td>
<td>0/8</td>
<td>0/8</td>
<td>6/8</td>
<td>6/8</td>
<td>8/8</td>
<td>8/8</td>
<td>8/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
<td>0/8</td>
</tr>
<tr>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
<td>16/40</td>
</tr>
<tr>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
<td>0/14</td>
</tr>
<tr>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
<td>0/16</td>
</tr>
<tr>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
</tr>
<tr>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>1/2</td>
<td>0/2</td>
<td>2/2</td>
<td>2/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
</tr>
<tr>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
</tr>
<tr>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>2/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
</tr>
<tr>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
<td>0/6</td>
</tr>
<tr>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
<td>0/9</td>
</tr>
<tr>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
<td>0/7</td>
</tr>
<tr>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
<td>0/5</td>
</tr>
<tr>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
<td>0/2</td>
</tr>
</tbody>
</table>
Trends

Architectural convergence

Ongoing perf work to make JS a better language to compile to (OdinMonkey, FTL, TF)

Ongoing ES6 feature work to make JS a better language to write
2015 predictions
TF landing?
More LLVM passes enabled in FTL?
ES6, ES7, other language experimentations?