Optimization Coaching for JavaScript

Vincent St-Amour
PLT © Northwestern University

Shu-yu Guo
Mozilla Research

ECOOP 2015 — July 9th, 2015
Once upon a time in the Racket community...
It renders text, too!
provide
(let:
  \( h-1 \)
  \( \text{transform-compose} \)
  \( \text{int-y-max} \) : \( \int \to \mathbb{R} \)
  \( \text{new-x} \) : \( \mathbb{R} \)
  \( \text{w} \) : \( \mathbb{N} \)
  \( \text{fm} \)
  \( \text{fm2} \) : \( \mathbb{N} \)
  \( \text{k} \)
  \( \text{c} \)
  \( \text{flonum} \)

\( \text{h-1} \) returns the inverse of \( h \). The function \( h \) is defined as:

\[
  h(x) = \frac{x}{\text{w}}.
\]

The inverse function \( h^{-1} \) is:

\[
  h^{-1}(y) = \text{w} \times y.
\]

The code snippet also includes a definition for \( \text{transform-compose} \), which is not fully visible in the image.
1 hour later...
20 hours later...
define-inline

(define (flomap-lift-helper f) ...)

There must be a better way.
This could happen anywhere.
This could happen anywhere.
This could happen anywhere.
Racket Solution

[St-Amour et al., OOPSLA 2012]
Racket Solution

[St-Amour et al., OOPSLA 2012]
Optimization Coach

Seconds later
20:0:
flomap-lift-helper

Missed Inlining (0 success out of 46)
Consider using `define-inline`, `begin-encourage-inline` or turning this function into a macro to force inlining.
JavaScript Solution
This talk

JavaScript Solution

JS
Today's roadmap

What is Optimization Coaching?

How Does the Coach Work?

How Well Does the Coach Work?
What is Optimization Coaching?
Dialog between compilers and programmers

badness: 2023
raytrace.js:432:12
property: isHit

Can't inline assignment.
This operation may add a new property to objects.

Initialize the property in the constructor to enable optimizations.

Near misses

Recommendations
Compilers must be conservative

<table>
<thead>
<tr>
<th>System</th>
<th>Height</th>
<th>Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>saturn_V</td>
<td>110</td>
<td>3</td>
</tr>
<tr>
<td>ariane_5</td>
<td>52</td>
<td>2</td>
</tr>
</tbody>
</table>
Compilers must be conservative

\[
\text{saturn}_V = \{ \text{height: 110, stages: 3} \} \\
\text{ariane}_5 = \{ \text{stages: 2, height: 52} \} \\
\text{ariane}_5 = \{ \text{height: 52, stages: 2} \}
\]

Uniformity $\rightarrow$ Optimizations!
Compilers must be conservative

saturn_V =
{height: 110,
stages: 3}

ariane_5 =
{stages: 2,
height: 52}

ariane_5 =
{height: 52,
stages: 2}

My rocket has:
height 52
stages 2

≠

My rocket has:
stages 2
height 52
Compilers must be conservative

\[
\text{satin}_V = \{\text{height}: 110, \text{stages}: 3\} \quad \text{ariane}_5 = \{\text{stages}: 2, \text{height}: 52\} \quad \text{ariane}_5 = \{\text{height}: 52, \text{stages}: 2\}
\]

These properties are not always in the same location.

Try to always initialize them in the same order.

Recommendations can change semantics!
Programmers have limited bandwidth
Programmers have limited bandwidth
**Why JS for coaching?**

- Can it work beyond Racket?
- Different compilation model (JIT)
- Different language (OO)

**Why coaching for JS?**

- Hard to write performant code
- Performance matters
- Non-experts / multi-language programmers
How does it work?
Architecture

SpiderMonkey
  IonMonkey
  Optimizer Instrumentation

Profiler
  logs

Optimization Analysis
  profile
  logs

Recommendation Generation
  near misses
  reports

UI
Log optimization decisions
(attempts, successes, failures)
Emit profile events
(get logs out of the engine)
Architecture

Produce near miss reports
(pruning, merging, ranking)
Fill recommendation templates
( general advice + targeted info )
Architecture

Show reports and recommendations
(consumed by programmers)
Architecture

SpiderMonkey
  IonMonkey
    Optimizer Instrumentation
  logs → Profiler
  logs → Optimization Analysis
  profile → Recommendation Generation
  near misses
  reports → UI

... Profiling-Based Badness Temporal Merging ...

{ } JIT
Profiling-Based Badness

- Launch Rocket
- Track Rocket
Profiling-Based Badness
Profi ling-Based Badness

Compile launch-rocket
Profiling-Based Badness

Compile launch-rocket
Profiling-Based Badness

Compile launch-rocket

Compile track-rocket
Profiling-Based Badness

Compile launch-rocket

Compile track-rocket

Compile track-rocket
Profiling-Based Badness

- Compile `launch-rocket` at 0 to 3
- Compile `track-rocket` at 7 to 18
- Compile `track-rocket` at 18 to 20
Profiling-Based Badness

Compile `launch-rocket`
- `rocket.stages`
  - Trying: PIC
    - exotic object

Compile `track-rocket`
- `rocket.height`
  - Trying: Direct load
    - diff. locations

Compile `track-rocket`
- `rocket.height`
  - Trying: Direct load
    - diff. locations
Profiling-Based Badness

Compile `launch-rocket`

```
rocket.stages
Trying: PIC
✗ exotic object
Badness = 7 - 3
= 4
...
```

Compile `track-rocket`

```
rocket.height
Trying: Direct load
✗ diff. locations
Badness = 18 - 10
= 8
...
```

Compile `track-rocket`

```
rocket.height
Trying: Direct load
✗ diff. locations
Badness = 20 - 18
= 2
...
```
Temporal Merging

Compile `launch-rocket`

`rocket.stages`

Trying: PIC

× exotic object

Badness = 7 - 3

= 4

...

Compile `track-rocket`

`rocket.height`

Trying: Direct load

× diff. locations

Badness = 18 - 10

= 8

...

Compile `track-rocket`

`rocket.height`

Trying: Direct load

× diff. locations

Badness = 20 - 18

= 2

...
Temporal Merging

Compile `launch-rocket`

- `rocket.stages`
  - Trying: PIC
  - exotic object
  - Badness = 7 - 3
  - = 4
  - ...

Compile `track-rocket`

- `rocket.height`
  - Trying: Offset
  - diff. locations
  - Badness = 8 + 2
  - = 10
  - ...

Timeline:

- `0` - `3` (Gold)
- `3` - `7` (Orange)
- `7` - `10` (Purple)
- `10` - `20` (Black)
Further Sightseeing
(In the Paper)

- Solution Site Inference
- Same-Property Analysis
- By-Solution Merging
- By-Constructor Merging
- Profiler-Driven Instrumentation
- Profiling-Based Badness
- Temporal Merging
- Irrelevant Failure Pruning
- Partial Success Shortcircuited
How well does it work?
Hypothesis: Coaching improves performance

- Baseline: Non-optimized
- Coached: Followed recommendations (Minutes of work)

Speedup, higher is better

Experiment

- Take Octane benchmarks
- Run the coach
- Follow recommendations
- Measure performance impact (Octane score)
Hypothesis: Coaching improves performance

Baseline: Non-optimized
Coached: Followed recommendations  (Minutes of work)

Speedup, higher is better

V8

JavaScriptCore
Hypothesis: Coaching improves performance

Baseline: Non-optimized
Coached: Followed recommendations (Minutes of work)

Speedup, higher is better

Cross-engine speedups

Engines overfit to Octane

V8
JavaScriptCore
**Hypothesis: Recommendations are low-effort**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Total LOC</th>
<th>LOC Added</th>
<th>LOC Deleted</th>
<th>LOC Edited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richards</td>
<td>538</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>881</td>
<td>12</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>RayTrace</td>
<td>903</td>
<td>10</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>Splay</td>
<td>422</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>415</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>PdfJS</td>
<td>33,053</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Crypto</td>
<td>1,698</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Box2D</td>
<td>10,970</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

At most 42 LOC!

Simple, mechanical changes!
Further Sightseeing
(In the Paper)

- Recommendation quality
- Discussion of individual recommendations
Wrapping Up
Coming soon to a browser near you
This Talk

Coaching works

• beyond Racket
• with JIT compilation
• with OO optimizations
The Coaching Philosophy

Compilers are great, but they can fail

Compilers gather tons of information

Liberate it, and show it to programmers!

They may succeed where compilers fail
The Coaching Philosophy

are great, but they can fail

gather tons of information

Liberate it, and show it to programmers!

They may succeed where fail
The Coaching Philosophy

Runtimes are great, but they can fail

Runtimes gather tons of information

Liberate it, and show it to programmers!

They may succeed where Runtimes fail
The Coaching Philosophy

OSes are great, but they can fail

OSes gather tons of information

Liberate it, and show it to programmers!

They may succeed where OSes fail
The Coaching Philosophy

? (noun) are great, but they can fail

? (noun) gather tons of information

Liberate it, and show it to programmers!

They may succeed where ____(noun)____ fail
The Coaching Philosophy

? are great, but they can fail
(noun)

? gather tons of information
(noun)

Liberate it, and show it to programmers!

They may succeed where ? fail
(noun)

Thank you!