Inspecting Node.js Performance

With N|Solid and Deoptigate

@thlorenz

NODESOURCE®

Inspecting Node.js Performance

@thlorenz

NODESOURCE®

Come to Colombia

Slides

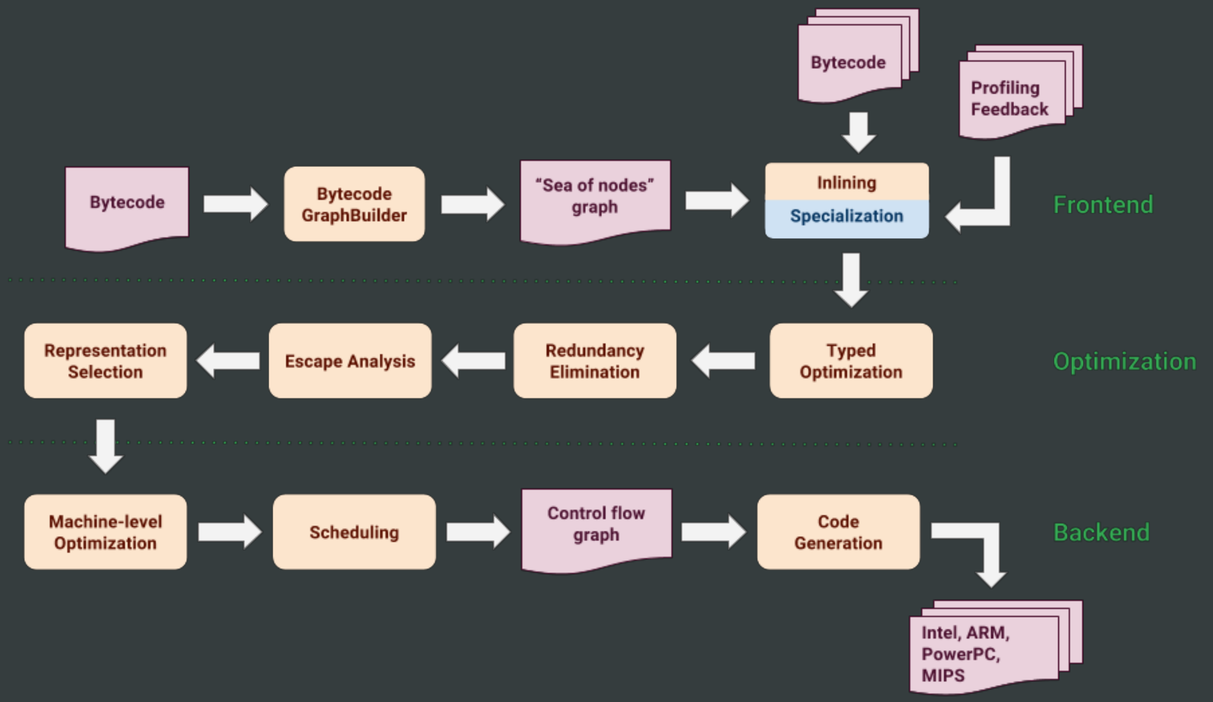
nsrc.io/deoptigate-slides

What changed in V8 and Why

- Overview of New Compiler Pipeline
- Why is it Faster?
- Which language features are supported?
- What does that mean for your JavaScript?

New V8 Compiler Pipeline

Cleanly split into *Frontend*, *Optimization Layer* and *Backend*



New V8 Compiler Pipeline

- Ignition Interpreter
- TurboFan optimizing Compiler

Ignition Interpreter

- runs highly optimized interpreter code
- *Inline Caches (aka ICs)* gather knowledge about types while program runs, stored in *FeedbackVector*

TurboFan Optimizing Compiler

- recompiles and optimizes *hot code* identified by the runtime profiler
- speculates that kinds of values seen in the past will be seen in the future and generates optimized code just for those cases

Many More Details

If interested watch this talk:

https://youtu.be/J9HAvlW7BqA

Just a Few More Things

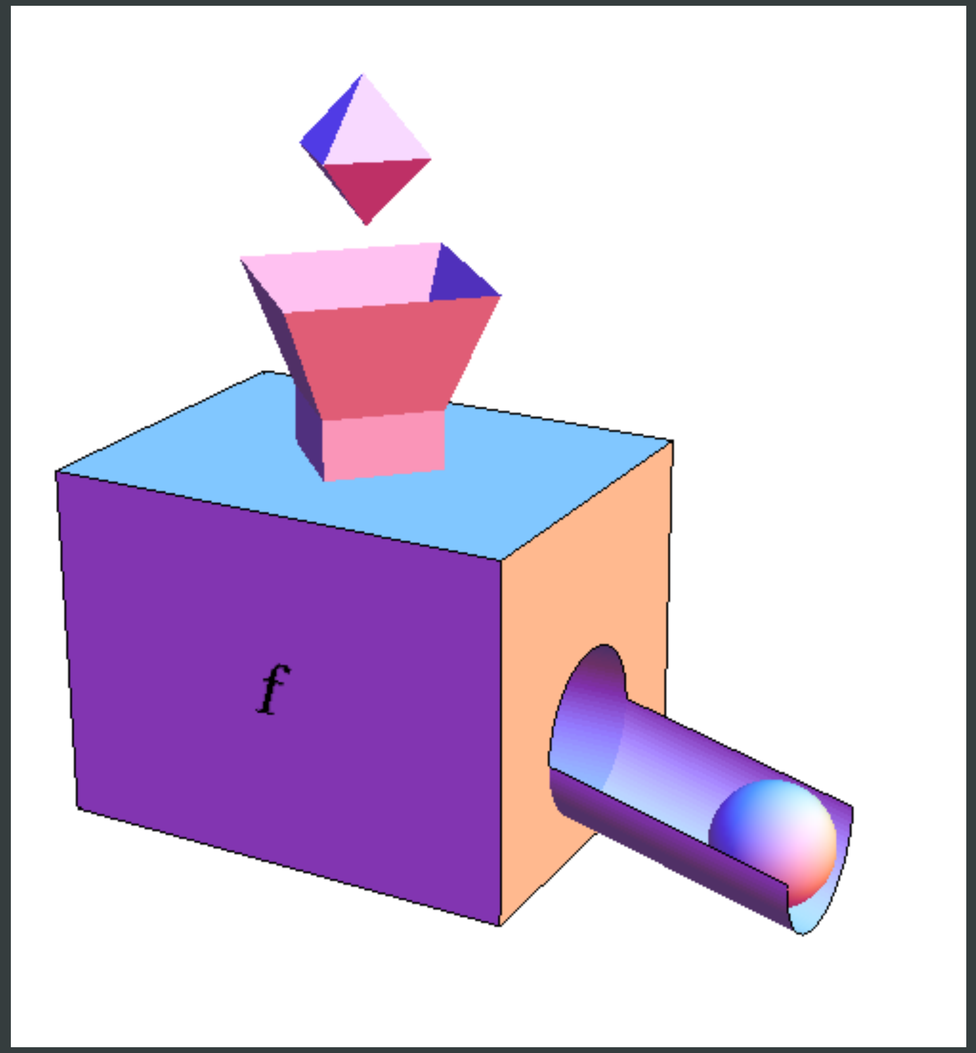
How to Write Code that runs Fast?

Optimized code for fewer cases is smaller and executes at peak speed.

Smaller is Better

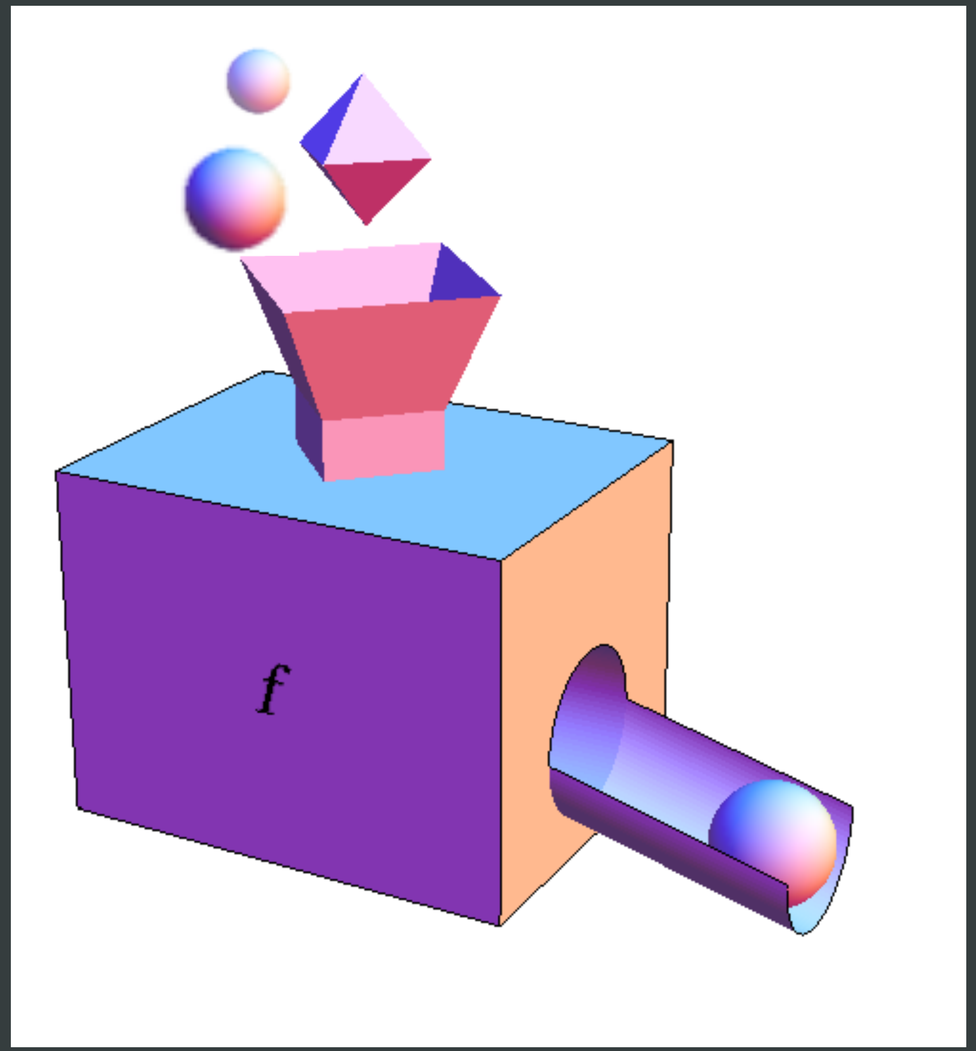
Mono-Poly-Megamorphism

Monomorphic === One Input Type



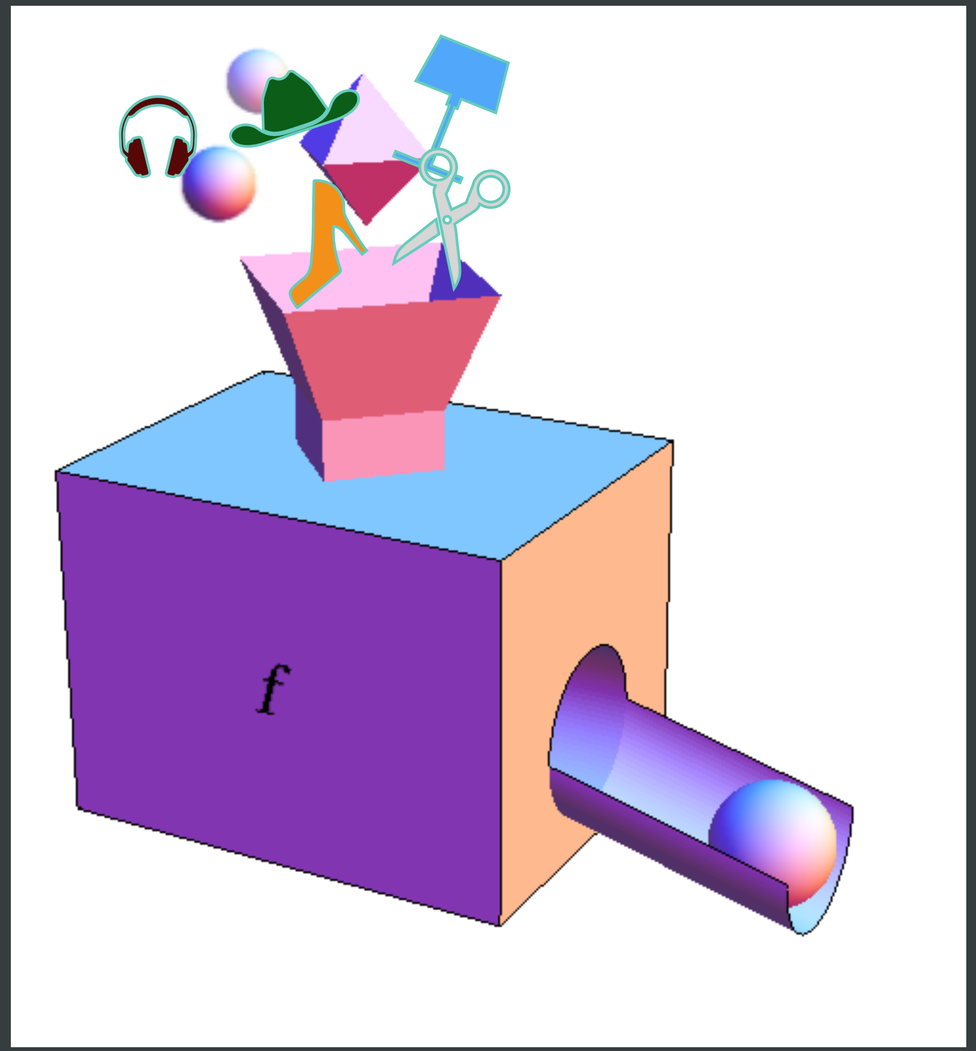
Mono-Poly-Megamorphism

Polymorhic === Two - Four Input Types



Mono-Poly-Megamorphism

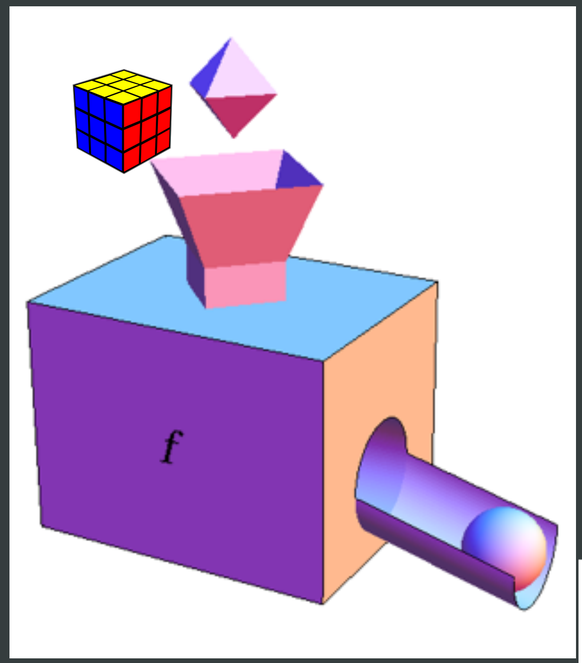
Megamorphic === Five+ Input Types

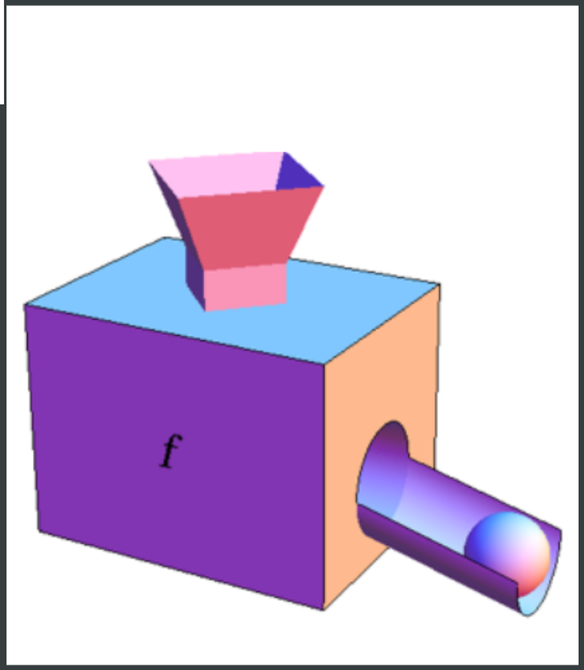


Mono-Poly-Megamorphism

Monomorphic operations are easier optimized

Prefer multiple monomorphic functions over one polymorphic function where performance matters



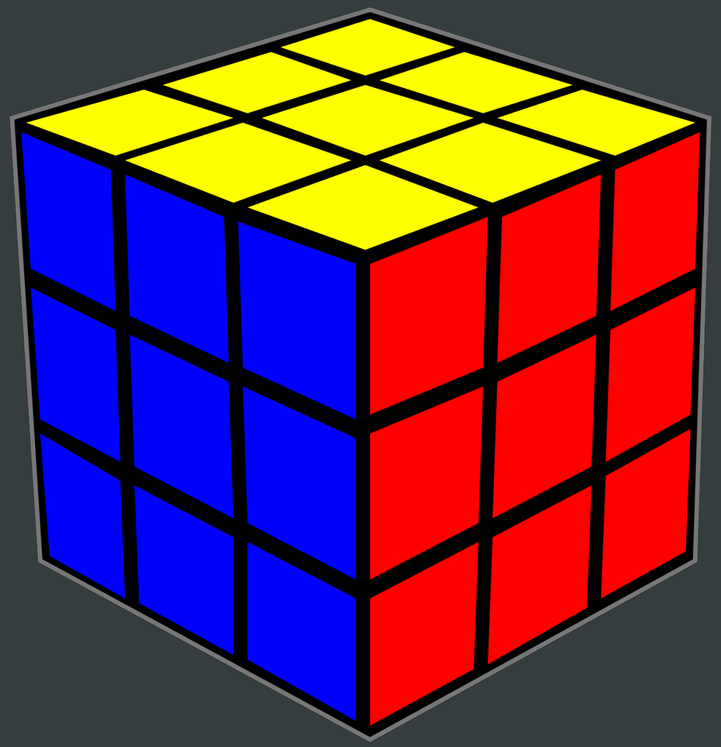


Mono-Poly-Megamorphism

V8 creates hidden classes for your JavaScript objects *aka* Maps

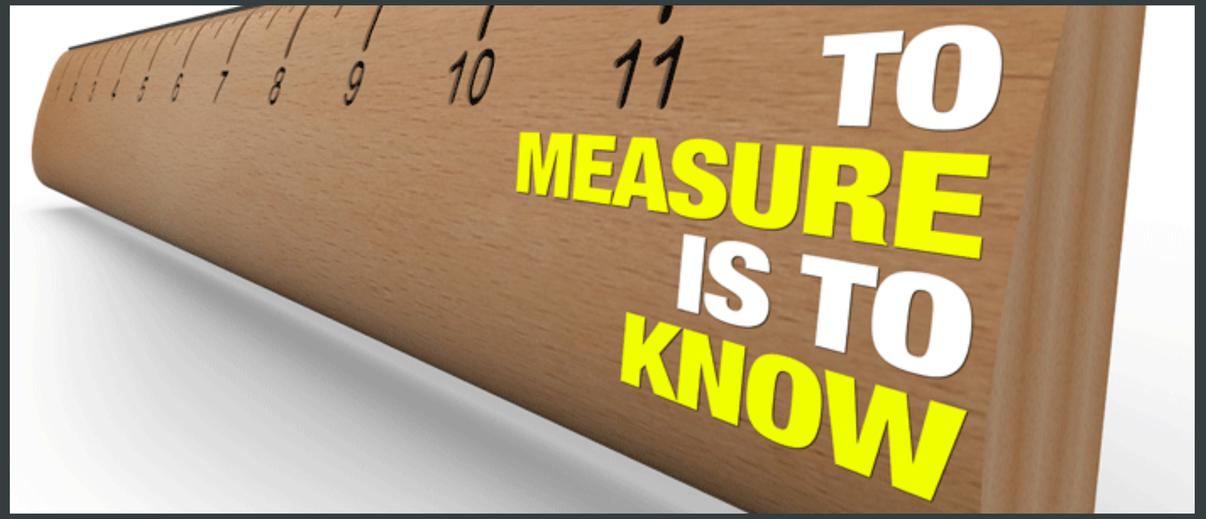
Mono-Poly-Megamorphism

Maps



Recommendations

Measure *before* and *after* each optimization you apply to your code



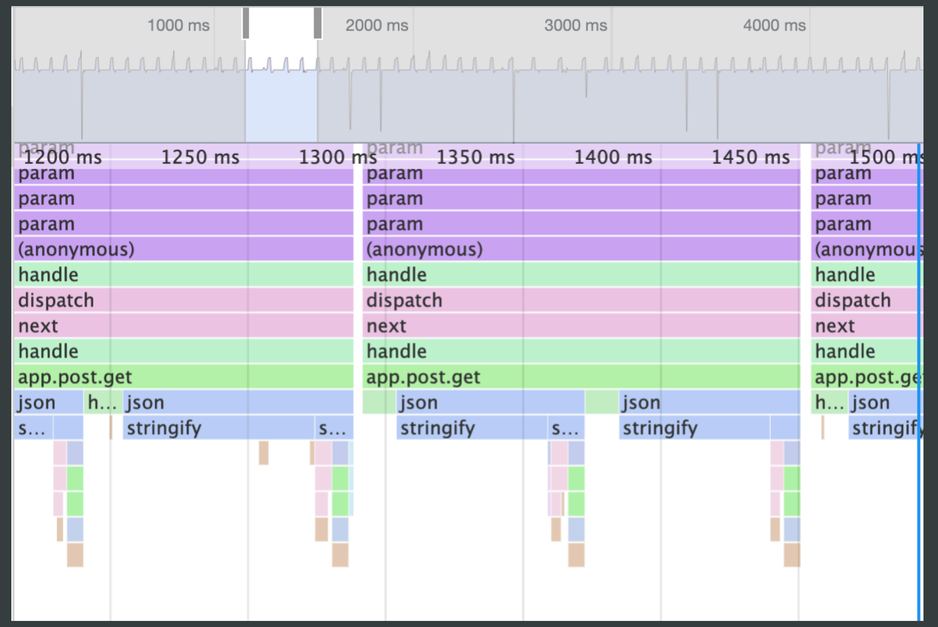
Measure Performance Under Load

Determine where bottlenecks are in your application.

Measure Performance Under Load

Determine where bottlenecks are in your application.

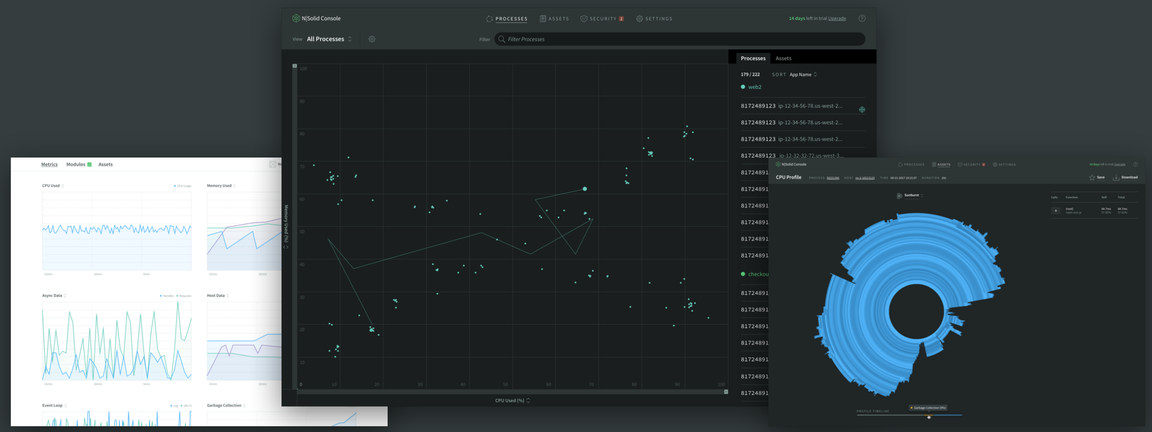
Chrome Devtools



N|Solid

Measure Performance Under Load

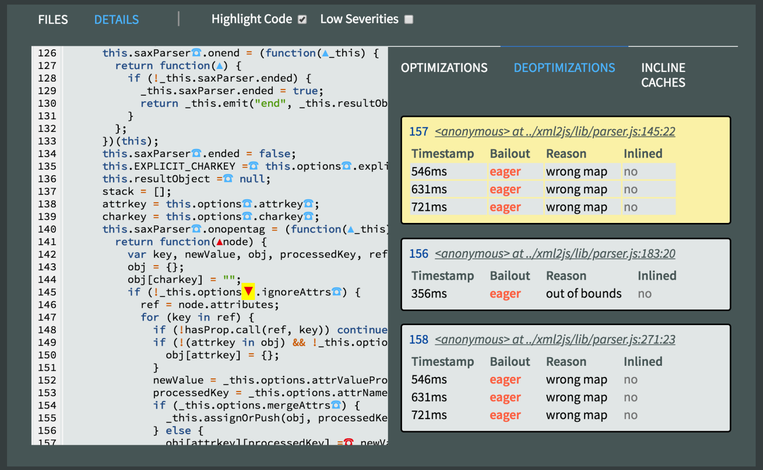
Determine where bottlenecks are in your application.



Deoptigate

Understand Optimizations/Deoptimizations

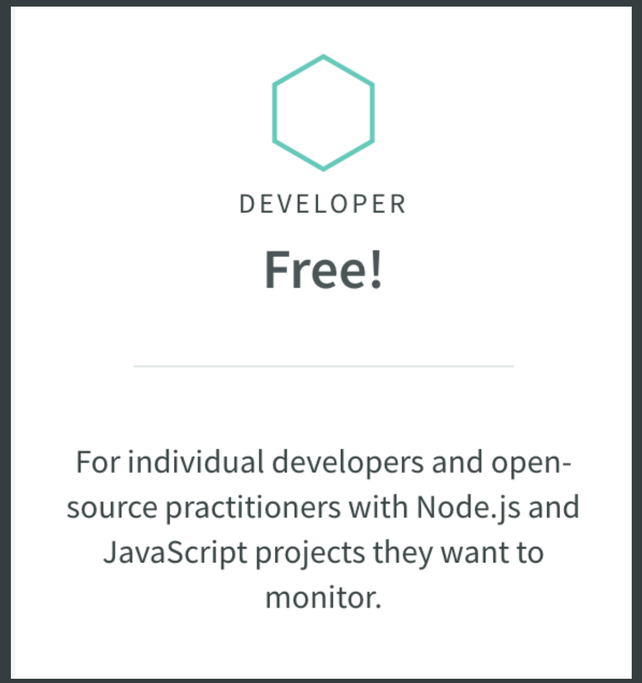
Determine if and why V8 runs your code slower than expected.



Demo

Getting Setup

- Try N|Solid free after signing up for a free account
  - https://nodesource.com/products/nsolid
- Install deoptigate `npm install -g deoptigate`
  - https://github.com/thlorenz/deoptigate



Resources V8

- **V8-perf resources**
  - V8 compiler
  - Language Features
- V8 code search

Resources Blog Posts And Documentation

- deoptigate
- N|Solid Overview and Documentation
- Why the New V8 is so Damn Fast

Thank you.

Thorsten Lorenz

@thlorenz