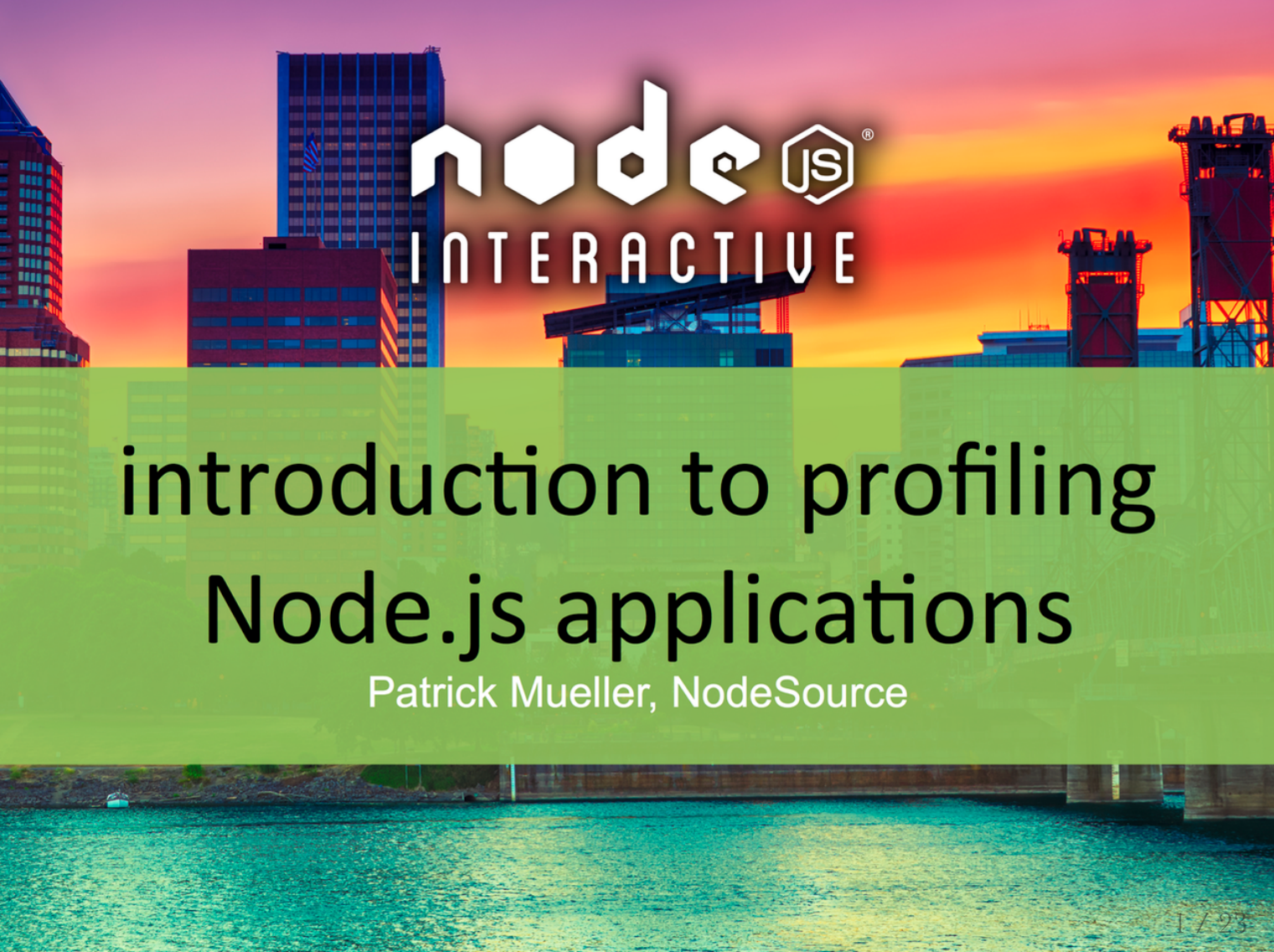



# introduction to profiling Node.js applications

Patrick Mueller, NodeSource

# introduction to profiling Node.js applications

Patrick Mueller **@pmuellr**, **muellerware.org**
senior node engineer at NodeSource

http://pmuellr.github.io/slides/2015/12-profiling-node-intro
http://pmuellr.github.io/slides/2015/12-profiling-node-intro/slides.pdf
http://pmuellr.github.io/slides/ (all of Patrick's slides)

# what kind of profiling?

- **performance** with V8's CPU profiler
- **memory** with V8's heap snapshots

# profiling performance

# what does V8's CPU profiler do?

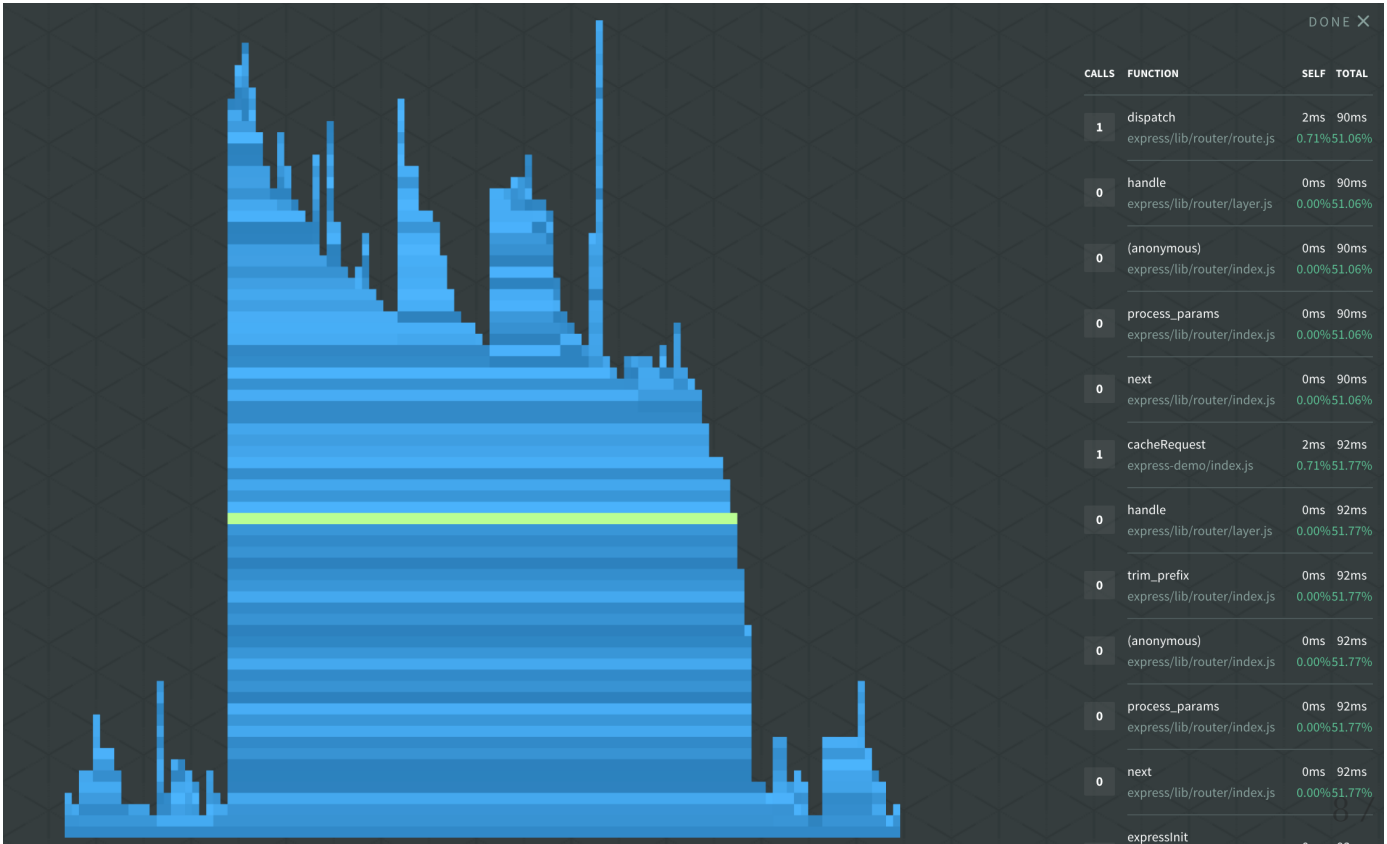
- turn profiler on / off
- when on, at regular intervals, V8 will capture current stack trace, with time stamp, and source file / line numbers
- when turned off, profiler will aggregate the information, and produce a JSON data structure for analysis tools

# time-line showing stack traces

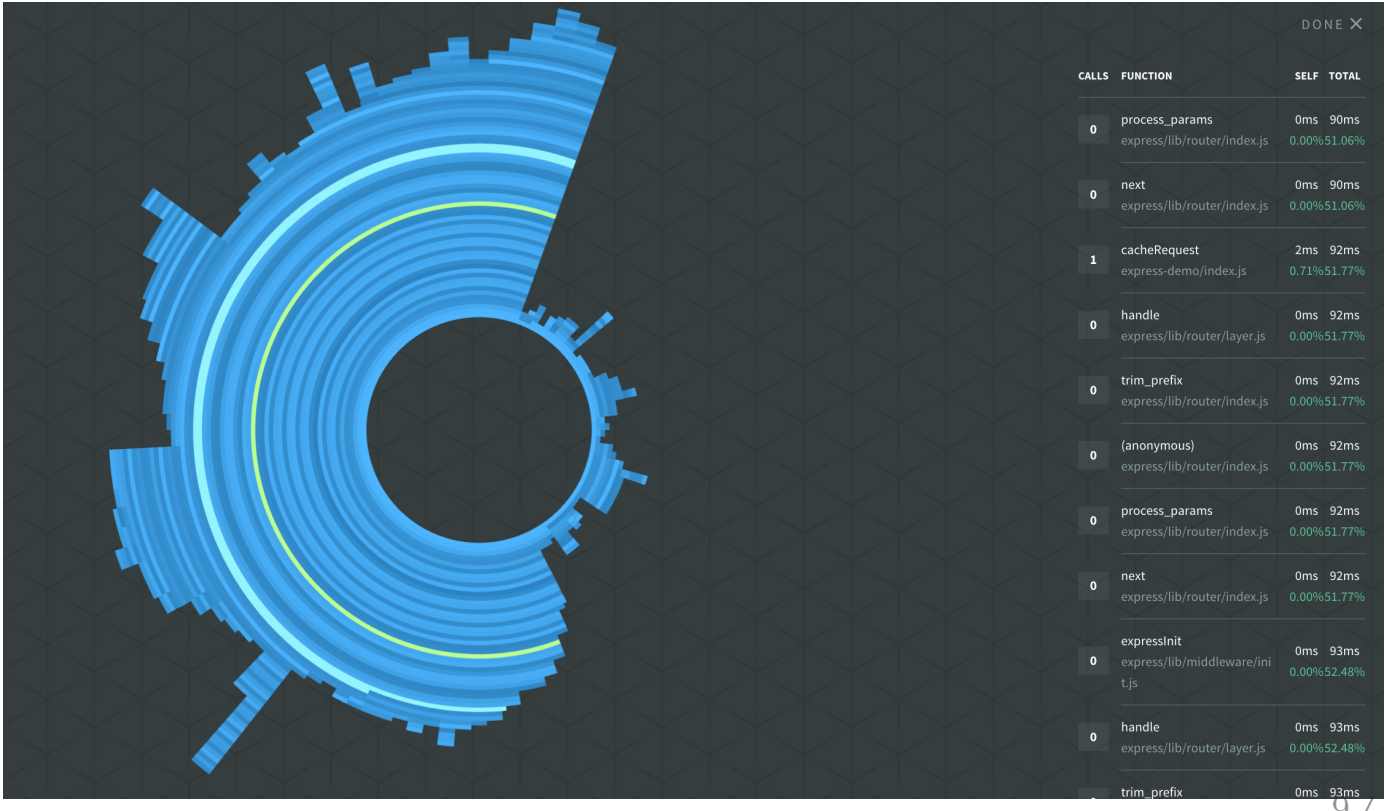
# table showing functions time

| Self | | Total | | Function | |
|---|---|---|---|---|---|
| 4831.8 ms | | 4831.8 ms | | (idle) | (program):-1 |
| 16.3 ms | 9.22 % | 16.3 ms | 9.22 % | (program) | (program):-1 |
| 12.5 ms | 7.09 % | 12.5 ms | 7.09 % | (garbage collector) | (program):-1 |
| 10.0 ms | 5.67 % | 13.8 ms | 7.80 % | ▶ c | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 7.5 ms | 4.26 % | 8.8 ms | 4.96 % | ▶ Lexer.next | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 6.3 ms | 3.55 % | 6.3 ms | 3.55 % | ▶ spawn | (program):-1 |
| 3.8 ms | 2.13 % | 3.8 ms | 2.13 % | ▶ now | (program):-1 |
| 3.8 ms | 2.13 % | 6.3 ms | 3.55 % | ▶ pp.eat | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 2.5 ms | 1.42 % | 18.8 ms | 10.64 % | ▶ pp.parseExprSubsc... | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 2.5 ms | 1.42 % | 2.5 ms | 1.42 % | ▶ pp.finishNode | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 2.5 ms | 1.42 % | 2.5 ms | 1.42 % | systemStats | nsolid.js:227 |
| 2.5 ms | 1.42 % | 2.5 ms | 1.42 % | ▶ posix.dirname | path.js:528 |
| 2.5 ms | 1.42 % | 30.0 ms | 17.02 % | ▶ parse | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 2.5 ms | 1.42 % | 95.0 ms | 53.90 % | ▶ app | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 2.5 ms | 1.42 % | 8.8 ms | 4.96 % | ▶ OutgoingMessage.end | _http_outgoing.js:513 |
| 2.5 ms | 1.42 % | 2.5 ms | 1.42 % | ▶ ServerResponse.writeHead | _http_server.js:159 |
| 2.5 ms | 1.42 % | 3.8 ms | 2.13 % | ▶ Agent.addRequest | _http_agent.js:109 |
| 2.5 ms | 1.42 % | 87.5 ms | 49.65 % | ▶ render | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 2.5 ms | 1.42 % | 2.5 ms | 1.42 % | ▶ slice | buffer.js:609 |
| 2.5 ms | 1.42 % | 108.8 ms | 61.70 % | ▶ emit | events.js:116 |
| 1.3 ms | 0.71 % | 91.3 ms | 51.77 % | ▶ cacheRequest | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 1.3 ms | 0.71 % | 1.3 ms | 0.71 % | ▶ posix.join | path.js:474 |
| 1.3 ms | 0.71 % | 1.3 ms | 0.71 % | ▶ pp.readString | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 1.3 ms | 0.71 % | 7.5 ms | 4.26 % | ▶ base.NewExpressio... | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 1.3 ms | 0.71 % | 1.3 ms | 0.71 % | ▶ _tokentype.types.b... | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 1.3 ms | 0.71 % | 11.3 ms | 6.38 % | ▶ pp.parseExprList | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |
| 1.3 ms | 0.71 % | 5.0 ms | 2.84 % | ▶ pp.parseIdent | /Users/pmuellr/Projects/slides/2015/12-profiling-node-intro/demos |

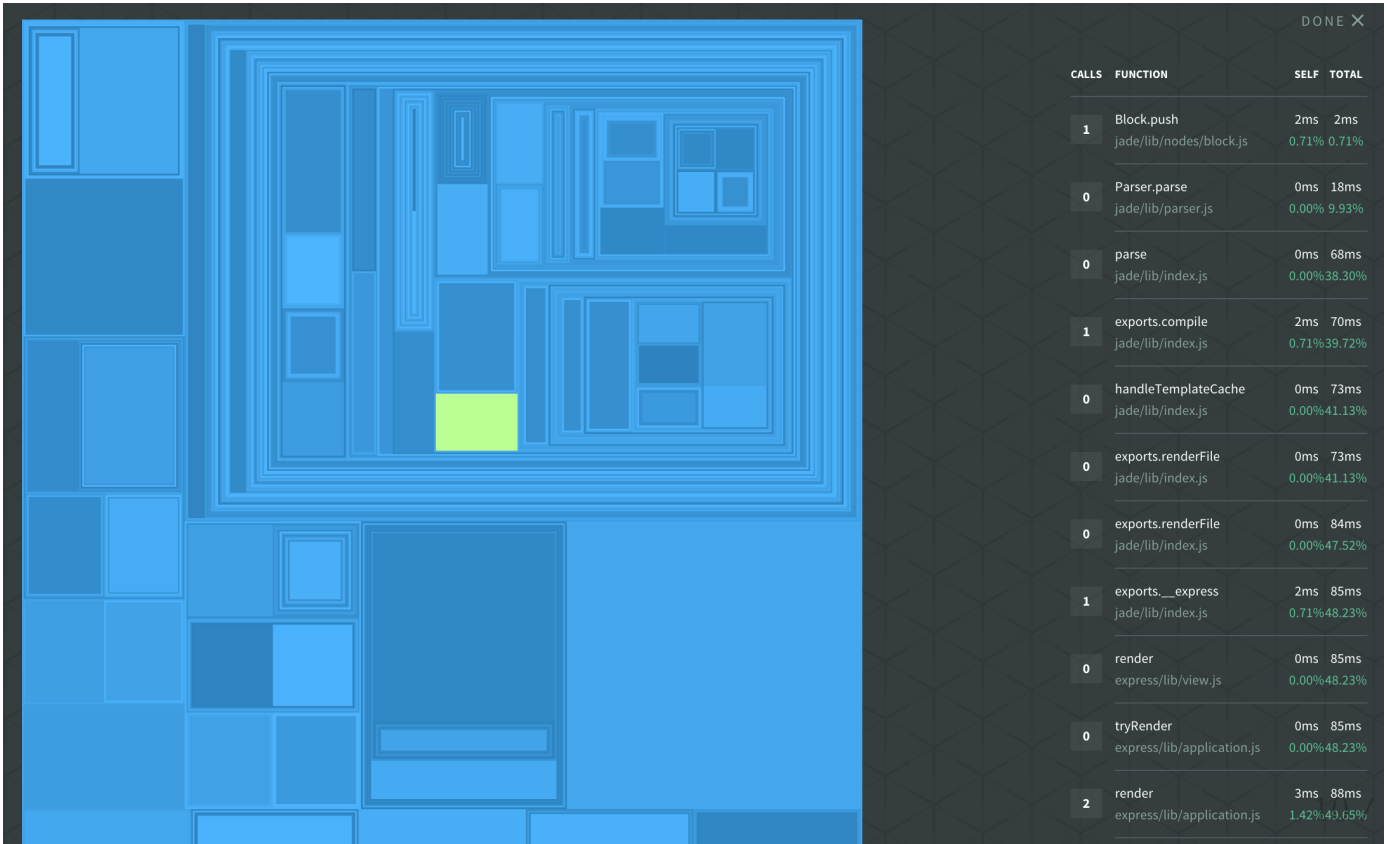
# flame graph


DONE
CALLS FUNCTION SELF TOTAL
1 dispatch 2ms 90ms
express/lib/router/route.js 0.71% 51.06%
0 handle 0ms 90ms
express/lib/router/layer.js 0.00% 51.06%
0 (anonymous) 0ms 90ms
express/lib/router/index.js 0.00% 51.06%
0 process_params 0ms 90ms
express/lib/router/index.js 0.00% 51.06%
0 next 0ms 90ms
express/lib/router/index.js 0.00% 51.06%
1 cacheRequest 2ms 92ms
express-demo/index.js 0.71% 51.77%
0 handle 0ms 92ms
express/lib/router/layer.js 0.00% 51.77%
0 trim_prefix 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 (anonymous) 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 process_params 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 next 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
expressInit

# sunburst


DONE
CALLS FUNCTION SELF TOTAL
0 process_params 0ms 90ms
express/lib/router/index.js 0.00% 51.06%
0 next 0ms 90ms
express/lib/router/index.js 0.00% 51.06%
1 cacheRequest 2ms 92ms
express-demo/index.js 0.71% 51.77%
0 handle 0ms 92ms
express/lib/router/layer.js 0.00% 51.77%
0 trim_prefix 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 (anonymous) 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 process_params 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 next 0ms 92ms
express/lib/router/index.js 0.00% 51.77%
0 expressInit 0ms 93ms
express/lib/middleware/init.js 0.00% 52.48%
0 handle 0ms 93ms
express/lib/router/layer.js 0.00% 52.48%
trim_prefix 0ms 93ms

# treemap


DONE
CALLS FUNCTION SELF TOTAL
1 Block.push 2ms 2ms
jade/lib/nodes/block.js 0.71% 0.71%
0 Parser.parse 0ms 18ms
jade/lib/parser.js 0.00% 9.93%
0 parse 0ms 68ms
jade/lib/index.js 0.00% 38.30%
1 exports.compile 2ms 70ms
jade/lib/index.js 0.71% 39.72%
0 handleTemplateCache 0ms 73ms
jade/lib/index.js 0.00% 41.13%
0 exports.renderFile 0ms 73ms
jade/lib/index.js 0.00% 41.13%
0 exports.renderFile 0ms 84ms
jade/lib/index.js 0.00% 47.52%
1 exports.__express 2ms 85ms
jade/lib/index.js 0.71% 48.23%
0 render 0ms 85ms
express/lib/view.js 0.00% 48.23%
0 tryRender 0ms 85ms
express/lib/application.js 0.00% 48.23%
2 render 3ms 88ms
express/lib/application.js 1.42% 49.65%

# understanding CPU profiling

- intro: Google Developers: Speed Up JavaScript Execution

- **self time** - the time it took to run the function, **not** including any functions that it called

- **total time** - the time it took to run the function, including any functions that it called

# how can you get CPU profiles?

- npm v8-profiler (requires instrumenting your code)
- npm node-inspector
- StrongLoop arc
- NodeSource N|Solid

# demo time!

- using N|Solid - getting started info
- see the instructions in demos/README.md
- source for the express-demo

# profiling memory

# what are V8 heap snapshots?

- blah blah

# what kind of output can you get?

- blah blah

# how can you get heap snapshots?

- blah blah

# demo time!

see the instructions in demos/README.md

source for the express-demo

# profiling tips

# profiling performance

- blah blah

# profiling memory

- blah blah

# fin


node JS
INTERACTIVE