

Performance Profiling for V8



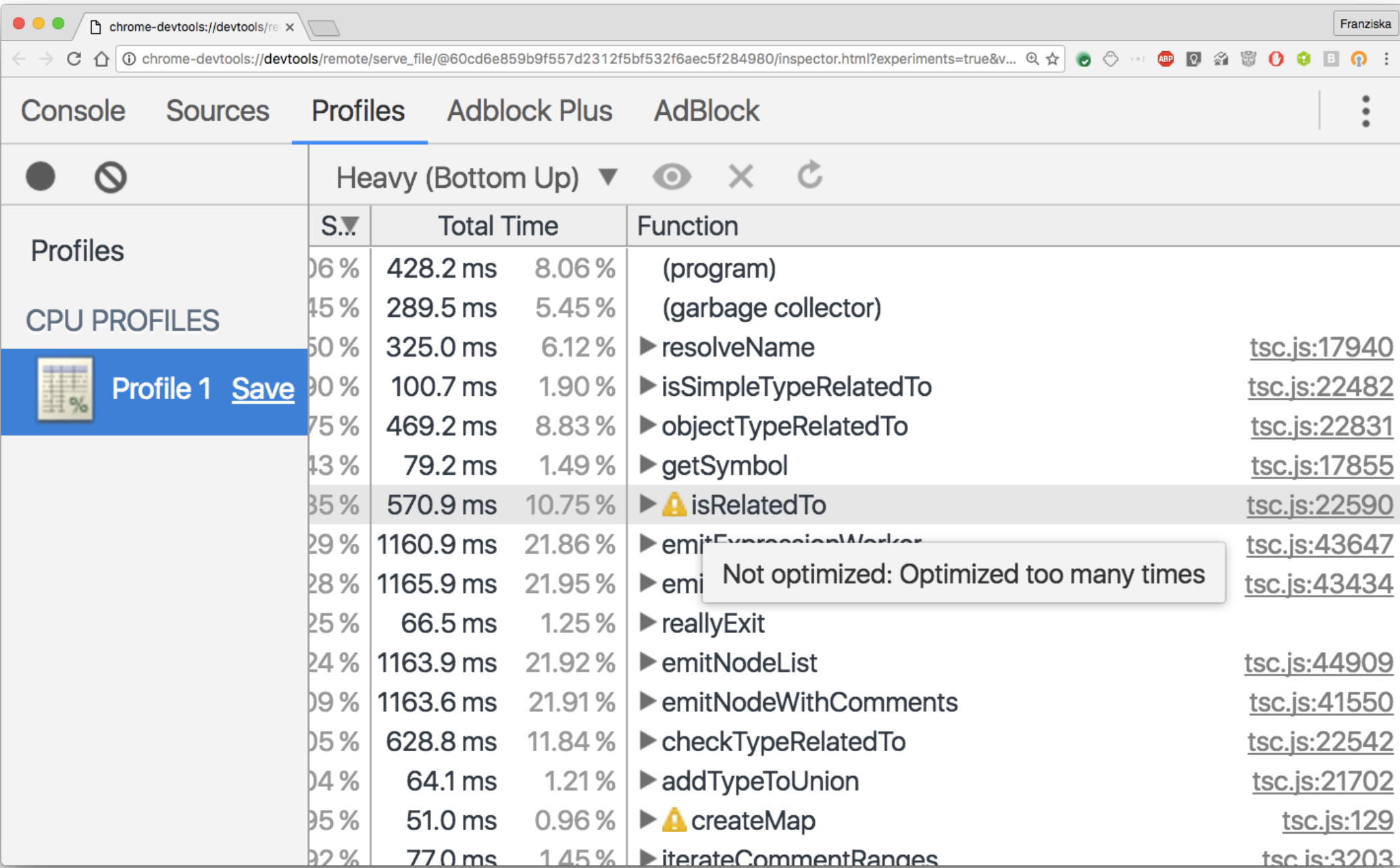
- Browser: ChakraCore, JSC, Spidermonkey, V8
- Node.js: ChakraCore, V8
- IoT: Duktape, JerryScript
- Electron: V8



Profiling V8

- Just in time (JIT) compilation
- Inline Caches (IC)
- Optimizing compiler
- Machine code

Chrome DevTools CPU Profile



The screenshot shows the Chrome DevTools interface with the CPU Profile tab selected. The profile is titled 'Heavy (Bottom Up)'. The table below lists the functions and their execution times, sorted by total time.

Profiles	S.▼	Total Time	Function
	06 %	428.2 ms	8.06 % (program)
	45 %	289.5 ms	5.45 % (garbage collector)
	50 %	325.0 ms	6.12 % ▶ resolveName tsc.js:17940
	90 %	100.7 ms	1.90 % ▶ isSimpleTypeRelatedTo tsc.js:22482
	75 %	469.2 ms	8.83 % ▶ objectTypeRelatedTo tsc.js:22831
	43 %	79.2 ms	1.49 % ▶ getSymbol tsc.js:17855
	35 %	570.9 ms	10.75 % ▶ ⚠ isRelatedTo tsc.js:22590
	29 %	1160.9 ms	21.86 % ▶ emitExpressionMarker tsc.js:43647
	28 %	1165.9 ms	21.95 % ▶ emit tsc.js:43434
	25 %	66.5 ms	1.25 % ▶ reallyExit
	24 %	1163.9 ms	21.92 % ▶ emitNodeList tsc.js:44909
	09 %	1163.6 ms	21.91 % ▶ emitNodeWithComments tsc.js:41550
	05 %	628.8 ms	11.84 % ▶ checkTypeRelatedTo tsc.js:22542
	04 %	64.1 ms	1.21 % ▶ addTypeToUnion tsc.js:21702
	05 %	51.0 ms	0.96 % ▶ ⚠ createMap tsc.js:129
	02 %	77.0 ms	1.45 % ▶ iterateCommentRanges tsc.js:3203

Profile 1 [Save](#)

Not optimized: Optimized too many times

	Function	
06 %	(program)	
45 %	(garbage collector)	
12 %	▶ resolveName	<u>tsc</u>
90 %	▶ isSimpleTypeRelatedTo	<u>tsc</u>
83 %	▶ objectTypeRelatedTo	<u>tsc</u>
49 %	▶ getSymbol	<u>tsc</u>
75 %	▶ ⚠ isRelatedTo	<u>tsc</u>
86 %	▶ emitExpressionMarker	<u>tsc</u>
95 %	▶ emit	<u>tsc</u>
25 %	▶ reallyExit	
92 %	▶ emitNodeList	<u>tsc</u>
91 %	▶ emitNodeWithComments	<u>tsc</u>
84 %	▶ checkTypeRelatedTo	<u>tsc</u>

Not optimized: Optimized too many times

JS is dynamically typed

- Not statically typed (Like C++, Java, Rust).

```
var obj = {  
    x: 1,  
    y: 1  
};
```

```
delete obj.x;  
obj.z = 1;
```

- Type information only available at runtime.

Just In Time (JIT) Compilation

Generate machine code during runtime, not **ahead of time** (AOT).

Property Access

```
function load(obj) {  
    return obj.x;  
}
```

- TypeError
- undefined
- prototype chain

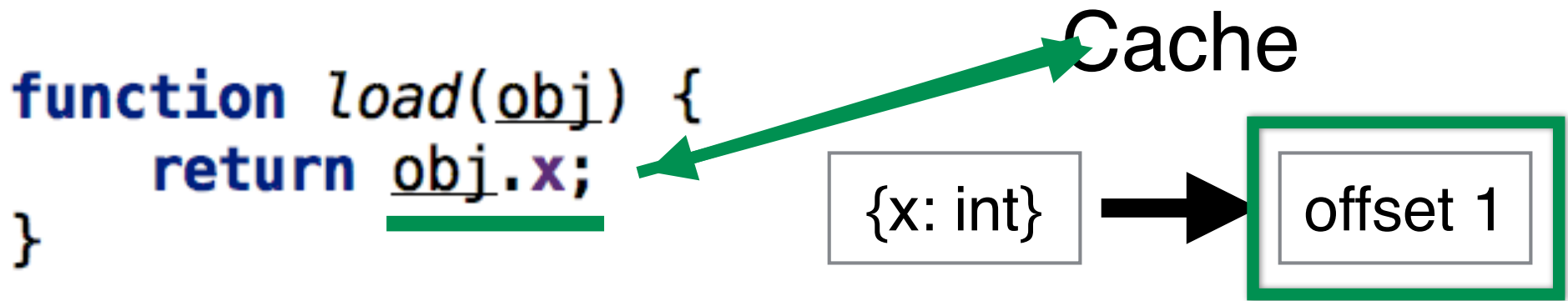
9.1.8.1 OrdinaryGet (*O*, *P*, *Receiver*)

When the abstract operation OrdinaryGet is called with Object *O*, proper ECMAScript language value *Receiver*, the following steps are taken:

1. Assert: **IsPropertyKey**(*P*) is **true**.
2. Let *desc* be ? *O*.**[[GetOwnProperty]]**(*P*).
3. If *desc* is **undefined**, then
 - a. Let *parent* be ? *O*.**[[GetPrototypeOf]]**().
 - b. If *parent* is **null**, return **undefined**.
 - c. Return ? *parent*.**[[Get]]**(*P*, *Receiver*).
4. If **IsDataDescriptor**(*desc*) is **true**, return *desc*.**[[Value]]**.
5. Assert: **IsAccessorDescriptor**(*desc*) is **true**.
6. Let *getter* be *desc*.**[[Get]]**.
7. If *getter* is **undefined**, return **undefined**.
8. Return ? **Call**(*getter*, *Receiver*).

EcmaScript specification

obj . x



load({x: 5});

load({x: 17});

9.1.8.1 OrdinaryGet (O, P, Receiver)

When the abstract operation OrdinaryGet is called with Object *O*, property key *P*, and ECMAScript language value *Receiver*, the following steps are taken:

1. Assert: `IsPropertyKey(P)` is `true`.
2. Let *desc* be ? `O.[[GetOwnProperty]](P)`.
3. If *desc* is `undefined`, then
 - a. Let *parent* be ? `O.[[GetPrototypeOf]]()`.
 - b. If *parent* is `null`, return `undefined`.
 - c. Return ? `parent.[[Get]](P, Receiver)`.
4. If `IsDataDescriptor(desc)` is `true`, return `desc.[[Value]]`.
5. Assert: `IsAccessorDescriptor(desc)` is `true`.
6. Let *getter* be `desc.[[Get]]`.
7. If *getter* is `undefined`, return `undefined`.
8. Return ? `Call(getter, Receiver)`.

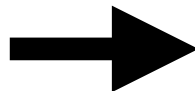
Inline Cache (IC)

```
function load(obj) {  
  if(cond) {  
    return obj.x;  
  } else {  
    return obj.x + 1;  
  }  
}
```

Shape of object = map = hidden class

{shape of object} → {fast path to property}

{x: int}



offset 1

Optimizing compiler

- Modern engines have optimizing compilers
- Basic compiler runs first and collects information, “hot functions” are then compiled by optimizing compiler

Optimization + IC

=

Speed

Optimized Machine Code

```
function load(obj) {  
    return obj.x;  
}
```

```
$ d8 --allow-natives-syntax --trace-opt --print-opt-code --code-comments load-opt.js  
[compiling method 0x9508e1f30c1 <JS Function load (SharedFunctionInfo 0xc3433e59a11)> using Crankshaft]  
[optimizing 0x9508e1f30c1 <JS Function load (SharedFunctionInfo 0xc3433e59a11)> - took 5.019, 0.103,  
0.089 ms]
```

```

;;; <@12,#7> context
032f7a584c2a 10 488b45f8 REX.W movq rax,[rbp-0x8]
;;; <@13,#7> gap
032f7a584c2e 14 488945e8 REX.W movq [rbp-0x18],rax
;;; <@16,#11> ----- B2 -----
;;; <@17,#11> gap
032f7a584c32 18 488bf0 REX.W movq rsi,rax
;;; <@18,#13> stack-check
032f7a584c35 21 493ba5100c0000 REX.W cmpq rsp,[r13+0xc10]
032f7a584c3c 28 7305 jnc 35 (0x132f7a584c43)
032f7a584c3e 30 e8bdd5f4ff call StackCheck (0x132f7a4d2200) ;; code: BUILTIN
;;; <@20,#13> lazy-bailout
;;; <@21,#13> gap
032f7a584c43 35 488b4510 REX.W movq rax,[rbp+0x10]
;;; <@22,#15> check-non-smi
032f7a584c47 39 a801 test al,0x1
032f7a584c49 41 0f8427000000 jz 86 (0x132f7a584c76)
;;; <@24,#16> check-maps
032f7a584c4f 47 49baf9afa8795f080000 REX.W movq r10,0x85f79a8aff9 ;; object: 0x85f79a8a
032f7a584c59 57 4c3950ff REX.W cmpq [rax-0x1],r10
032f7a584c5d 61 0f8518000000 jnz 91 (0x132f7a584c7b)
;;; <@26,#17> load-named-field
032f7a584c63 67 8b401b movl rax,[rax+0x1b]
;;; <@28,#21> smi-tag
032f7a584c66 70 8bd8 movl rbx,rax
032f7a584c68 72 48c1e320 REX.W shlq rbx, 32
;;; <@29,#21> gap
032f7a584c6c 76 488bc3 REX.W movq rax,rbx
;;; <@30,#19> return
032f7a584c6f 79 488be5 REX.W movq rsn,rbn

```

Jump table -----

call 0x3a9097b8400a

call 0x3a9097b84014

; deoptimization bailout 1

; deoptimization bailout 2

le.

IC States

- Uninitialized
- Monomorphic: 1 map
- Polymorphic: 2-4 map
- Megamorphic: more than 4 map


```

;;; <@12,#7> context
32f7a584c2a 10 488b45f8      REX.W movq rax,[rbp-0x8]
;;; <@13,#7> gap
32f7a584c2e 14 488945e8      REX.W movq [rbp-0x18],rax
;;; <@16,#11> ----- B2 -----
;;; <@17,#11> gap
32f7a584c32 18 488bf0        REX.W movq rsi,rax
;;; <@18,#13> stack-check
32f7a584c35 21 493ba5100c0000 REX.W cmpq rsp,[r13+0xc10]
32f7a584c3c 28 7305          jnc 35 (0x132f7a584c43)
32f7a584c3e 30 e8bdd5f4ff     call StackCheck (0x132f7a4d2200)    ;; code: BUILTIN
;;; <@20,#13> lazy-bailout
;;; <@21,#13> gap
32f7a584c43 35 488b4510      REX.W movq rax,[rbp+0x10]
;;; <@22,#15> check-non-smi
32f7a584c47 39 a801          test al,0x1
32f7a584c49 41 0f8427000000 jz 86 (0x132f7a584c76)
;;; <@24,#16> check-maps
32f7a584c4f 47 49baf9afa8795f080000 REX.W movq r10,0x8
32f7a584c59 57 4c3950ff      REX.W cmpq [rax-0x1],r10
32f7a584c5d 61 0f8518000000 jnz 91 (0x132f7a584c7b)
;;; <@26,#17> load-named-field
32f7a584c63 67 8b401b        movl rax,[rax+0x1b]
;;; <@28,#21> smi-tag
32f7a584c66 70 8bd8          movl rbx,rax
32f7a584c68 72 48c1e320      REX.W shlq rbx, 32
;;; <@29,#21> gap
32f7a584c6c 76 488bc3        REX.W movq rax,rbx
;;; <@30,#19> return
32f7a584c6f 79 488be5        REX.W movq rsp,rbp
32f7a584c72 82 5d           pop rbp
32f7a584c73 83 c21000        ret 0x10
;;; ----- Jump table -----
32f7a584c76 86 e88ff3d7ff     call 0x132f7a30400a    ;; deoptimization bailout 1
32f7a584c7b 91 e894f3d7ff     call 0x132f7a304014    ;; deoptimization bailout 2
;;; Safepoint table.

```

1 map in IC

`-js-flags="-print-opt-code -code-comments"`

```
--- Optimized code ---
optimization_id = 0
source_position = 15
kind = OPTIMIZED_FUNCTION
name = load
stack_slots = 5
compiler = crankshaft
```

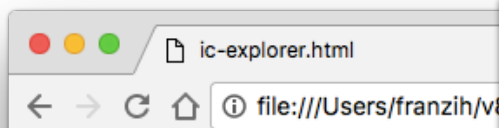
```
Instructions (size = 163)
```

0x2c845eb04d80	0	55	push rbp	
0x2c845eb04d81	1	4889e5	REX.W movq rbp, rsp	
0x2c845eb04d84	4	56	push rsi	
0x2c845eb04d85	5	57	push rdi	
0x2c845eb04d86	6	4883ec08	REX.W subq rsp, 0x8	
0x2c845eb04d8a	10	488b45f8	REX.W movq rax, [rbp-0x8]	
0x2c845eb04d8e	14	488945e8	REX.W movq [rbp-0x18], rax	
0x2c845eb04d92	18	488bf0	REX.W movq rsi, rax	
0x2c845eb04d95	21	493ba5100c0000	REX.W cmpq rsp, [r13+0xc10]	
0x2c845eb04d9c	28	7305	jnc 35 (0x2c845eb04da3)	
0x2c845eb04d9e	30	e85dd4f4ff	call StackCheck (0x2c845ea52200)	;; code: BUILTIN
0x2c845eb04da3	35	488b4510	REX.W movq rax, [rbp+0x10]	
0x2c845eb04da7	39	a801	test al, 0x1	
0x2c845eb04da9	41	0f8457000000	jz 134 (0x2c845eb04e06)	
0x2c845eb04daf	47	49baf9af8034610e0000	REX.W movq r10, 0xe613480aff9	;; object: 0x...
0x2c845eb04db9	57	4c3950ff	REX.W cmpq [rax-0x1], r10	
0x2c845eb04dbd	61	7434	jz 115 (0x2c845eb04df3)	
0x2c845eb04dbf	63	49ba01b18034610e0000	REX.W movq r10, 0xe613480b101	;; object: 0x...
0x2c845eb04dc9	73	4c3950ff	REX.W cmpq [rax-0x1], r10	
0x2c845eb04dcd	77	7424	jz 115 (0x2c845eb04df3)	
0x2c845eb04dcf	79	49ba59b18034610e0000	REX.W movq r10, 0xe613480b159	;; object: 0x...
0x2c845eb04dd9	89	4c3950ff	REX.W cmpq [rax-0x1], r10	
0x2c845eb04ddd	93	7414	jz 115 (0x2c845eb04df3)	
0x2c845eb04ddf	95	49bab1b18034610e0000	REX.W movq r10, 0xe613480b1b1	;; object: 0xe613480...
0x2c845eb04de9	105	4c3950ff	REX.W cmpq [rax-0x1], r10	
0x2c845eb04ded	109	0f8518000000	inz 139 (0x2c845eb04e0b)	
0x2c845eb04df3	115	8b401b	movl rax, [rax+0x1b]	
0x2c845eb04df6	118	8bd8	movl rbx, rax	
0x2c845eb04df8	120	48c1e320	REX.W shlq rbx, 32	
0x2c845eb04dfc	124	488bc3		
0x2c845eb04dff	127	488be5		
0x2c845eb04e02	130	5d		
0x2c845eb04e03	131	c21000		
0x2c845eb04e06	134	e8fff1		
0x2c845eb04e0c	139	e804f2		

Source positions:
pc offset position

4 maps in IC

```
; deoptimization bailout 1
; deoptimization bailout 2
```



`-js-flags="--trace_ic" ... > trace.txt`

ICE

Your IC-Explorer.

Usage

Run your script with `--trace_ic` and upload on this page:
`/path/to/d8 --trace_ic your_script.js > trace.txt`

Data

Choose File trace.txt

trace entries: 109620

Result

Group-Key: state

details	57.51%	41727	(N->N)
details	16.06%	11655	(0->.)
details	7.95%	5769	(.->1)
details	4.44%	3220	(P->P)
details	3.88%	2817	((UNINITIALIZED+UNINITIALIZED=UNINITIALIZED)->(SMI+SMI))
details	3.58%	2600	(1->P)

0 Uninitialized
1 Monomorphic
P Polymorphic
N Megamorphic

ic-explorer.html

file:///Users/franzih/v8/tools/ic-explorer.html

isRElatedTo 1 of 14

details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:33765 ~
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:11745 ~declareModuleMember+1289
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:11744 ~declareModuleMember+1176
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:1 *getTypeFromTypeLiteralOrFunctionOrConstructorTypeNode+254
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:1 *declareSymbolAndAddToSymbolTableWorker+83
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:3 *isContextSensitive+4549
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:16537 ~resolveAnonymousTypeMembers+1089
details	0.01%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:18327 ~isRelatedTo+2622

type [top 20 out of 1]

details	100%	7	LoadIC
---------	------	---	--------

category [top 20 out of 1]

details	100%	7	Load
---------	------	---	------

file [top 20 out of 1]

details	100%	7	/usr/local/google/home/franzih/TypeScript/lib/tsc.js:18327
---------	------	---	--

state [top 20 out of 6]

details	28.57%	2	(P->P)
details	14.29%	1	(0->.)
details	14.29%	1	(.->1)
details	14.29%	1	(1->P)

```
--trace-opt --trace-deopt
```

```
$ node --trace-opt --trace-deopt load-opt.js  
[compiling method 0x1b9f780f3139 <JS Function  
load (SharedFunctionInfo 0x3697a6859ad1)> using  
Crankshaft]  
[optimizing 0x1b9f780f3139 <JS Function load  
(SharedFunctionInfo 0x3697a6859ad1)> - took  
0.910. 0.052, 0.058 ms]  
[evicting entry from optimizing code map (notify  
deoptimized) for 0x3697a6859ad1  
<SharedFunctionInfo load>]
```

	Function	
06 %	(program)	
45 %	(garbage collector)	
12 %	▶ resolveName	tsc
90 %	▶ isSimpleTypeRelatedTo	tsc
83 %	▶ objectTypeRelatedTo	tsc
49 %	▶ getSymbol	tsc
75 %	▶ ⚠ isRelatedTo	tsc
86 %	▶ emitExpressionMarker	tsc
95 %	▶ emit	tsc
25 %	▶ reallyExit	
92 %	▶ emitNodeList	tsc
91 %	▶ emitNodeWithComments	tsc
84 %	▶ checkTypeRelatedTo	tsc

Not optimized: Optimized too many times

chrome-devtools://devtools/			
chrome-devtools://devtools/remote/serve_file/@60cd8e859b9f557d2312f5f32f6aec5f284980/inspector.html?experiments=true&v=...			
Insole Sources Profiles			
Heavy (Bottom Up) X			
Self Ti...	Total ...	Function	
0.90 %	6.65 %	isRelatedTo	
0.23 %	6.59 %	checkTypeRelatedTo	
0.18 %	0.06 %	compareSignaturesRelated	
0.18 %	0.01 %	typeRelatedToSomeType	
0.14 %	0.01 %	someTypeRelatedToType	
0.09 %	0.30 %	eachTypeRelatedToType	
0.04 %	1.02 %	propertiesRelatedTo	
0.01 %	0.01 %	isRelatedTo	
0.01 %	0.01 %	typeArgumentsRelatedTo	
0.00 %	0.01 %	typeInfoRelatedTo	

Optimized code			
optimization_id = 0			
source_position = 15			
kind = OPTIMIZED_FUNCTION			
name = load			
stack_slots = 5			
compiler = crankschaft			
Instructions (size = 163)			
0x2c845eb04d80	0	55	push rbp
0x2c845eb04d81	1	4889e5	REX.W movq rbp, rsp
0x2c845eb04d84	4	56	push rsi
0x2c845eb04d85	5	57	push rdi
0x2c845eb04d86	6	4883ec08	REX.W subq rsp, 0x8
0x2c845eb04d8a	10	488b45f8	REX.W movq rax, [rbp-0x8]
0x2c845eb04d8e	14	488945e8	REX.W movq [rbp-0x10], rax
0x2c845eb04d92	18	488b7f0	REX.W movq rsi, rax
0x2c845eb04d95	21	493ba5100c0000	REX.W cmqp rsp, [r13+0xc10]
0x2c845eb04d9c	28	7305	jnc 35 (0x2c845eb04da3)
0x2c845eb04d9e	30	e05d4f4ff	call StackCheck (0x2c845eb052200) ; code: BUILTIN
0x2c845eb04da3	35	488b4510	REX.W movq rax, [rbp+0x10]
0x2c845eb04da7	39	a801	test al, 0x1
0x2c845eb04da9	41	0f8457000000	jz 134 (0x2c845eb04e06)
0x2c845eb04daf	47	490af09a0834610e0000	REX.W movq r10, 0xe613480aff9 ; object: 0xe613480aff9 <Map(FAST_H
0x2c845eb04db9	57	4c3950ff	REX.W cmqp [rax-0x1], r10
0x2c845eb04dbd	61	7434	jz 115 (0x2c845eb04df3)
0x2c845eb04dbf	63	490ab1b18034610e0000	REX.W movq r10, 0xe613480b101 ; object: 0xe613480b101 <Map(FAST_H
0x2c845eb04dc9	73	4c3950ff	REX.W cmqp [rax-0x1], r10
0x2c845eb04dcd	77	7424	jz 115 (0x2c845eb04df3)
0x2c845eb04dcf	79	490a59b18034610e0000	REX.W movq r10, 0xe613480b159 ; object: 0xe613480b159 <Map(FAST_H
0x2c845eb04dd9	89	4c3950ff	REX.W cmqp [rax-0x1], r10
0x2c845eb04ddd	93	7414	jz 115 (0x2c845eb04df3)
0x2c845eb04ddf	95	490ab1b18034610e0000	REX.W movq r10, 0xe613480b1b1 ; object: 0xe613480b1b1 <Map(FAST_H
0x2c845eb04de9	105	4c3950ff	REX.W cmqp [rax-0x1], r10
0x2c845eb04ded	109	0f0510000000	jnz 129 (0x2c845eb04e0b)
0x2c845eb04df3	115	8b481b	movl rax, [rax+0x1b]
0x2c845eb04df6	118	8bd8	movl rbx, rax
0x2c845eb04df8	120	48c1e320	REX.W shlq rbx, 32
0x2c845eb04dfc	124	488bc3	REX.W movq rax, rbx
0x2c845eb04dff	127	488be5	REX.W movq rsp, rbp
0x2c845eb04e02	130	5d	pop rbp
0x2c845eb04e03	131	c21000	ret 0x10
0x2c845eb04e06	134	e0ff1f1d7ff	call 0x2c845eb08400a ; deoptimization bailout 1
0x2c845eb04e0b	139	e004f2d7ff	call 0x2c845eb084014 ; deoptimization bailout 2
Source positions:			
pc offset position			

TypeScript — vi lib/tsc.js — 157x41	
22583	if ((globalStringType === source && stringType === target)
22584	(globalNumberType === source && numberType === target)
22585	(globalBooleanType === source && booleanType === target)
22586	(getGlobalESSymbolType() === source && esSymbolType === target)) {
22587	reportError(ts.Diagnostics._0_is_a_primitive_but_1_is_a_wrapper_object_f
22588	}
22589	}
22590	function isRelatedTo(source, target, reportErrors, headMessage) {
22591	var result;
22592	if (source.flags & 96 && source.flags & 16777216) {
22593	source = source.regularType;
22594	}
22595	if (target.flags & 96 && target.flags & 16777216) {
22596	target = target.regularType;
22597	}
22598	if (source === target)
22599	return -1;
22600	if (relation === identityRelation) {
22601	return isIdenticalTo(source, target);
22602	}
22603	if (isSimpleTypeRelatedTo(source, target, relation, reportErrors ? reportEr
22604	return -1;
22605	if (source.flags & 8388608 && source.flags & 16777216) {
22606	if (hasExcessProperties(source, target, reportErrors)) {
22607	if (reportErrors) {
22608	reportRelationError(headMessage, source, target);
22609	}

ICE

Your IC-Explorer.

Usage

Run your script with `--trace_ic` and upload on this page:
`/path/to/d8 --trace_ic your_script.js > trace.txt`

Data

Choose File trace.txt

trace entries: 109620

Result

Group-Key: state

details 57.51% 41727 (N->N)

details 16.06% 11655 (0->.)

details 7.95% 5769 (.->1)

details 4.44% 3220 (P->P)

details 3.88% 2817 ((UNINITIALIZED+UNINITIALIZED=UNINITIALIZED)->(SMI+SMI))

details 3.58% 2600 (1->P)

Total Time		Function
ms	19.85 %	▼ isRelatedTo
ms	3.53 %	▶ compareSignaturesRelated
ms	19.85 %	▶ checkTypeRelatedTo
ms	0.06 %	▶ typeRelatedToSomeType
ms	0.02 %	▶ someTypeRelatedToType
ms	0.00 %	▶ eachTypeRelatedToType
ms	4.21 %	▶ propertiesRelatedTo
ms	0.06 %	▶ isRelatedTo
ms	0.00 %	▶ typeArgumentsRelatedTo
ms	0.00 %	▶ compareProperties
ms	0.00 %	▶ eachPropertyRelatedTo

Be careful with optimizations!

- Don't “optimize” unless you must
- Measure first

Be careful with optimizations!

- V8 internals change
- Different in other engines