



[Node-internals]

V8 & GC





Hello!

Vincent Vallet

NodeJs & performance evangelist

@Vince_Vallet



1

Introduction

Garbage Collector What is this ?

Good point : With NodeJS, no need to manage memory !



=> **G**arbage **C**ollector (**GC**) = take care of memory !

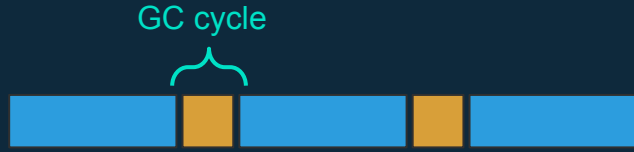


Type of GC

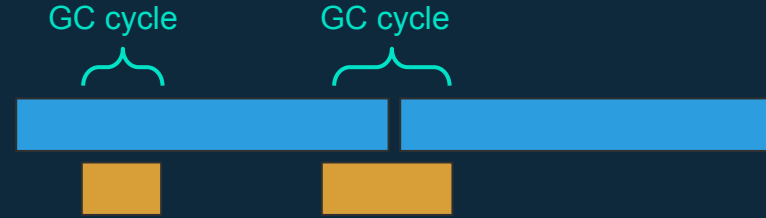
Type	Description
1	Scavenge (Minor GC)
2	Mark Sweep compact (Major GC)
4	Incremental marking / Lazy sweeping
8	Weak/Phantom callback processing
15	All



Stop the world



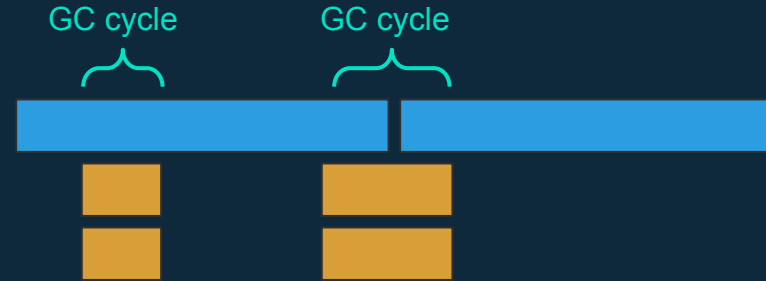
Concurrent



Incremental



Parallel + concurrent

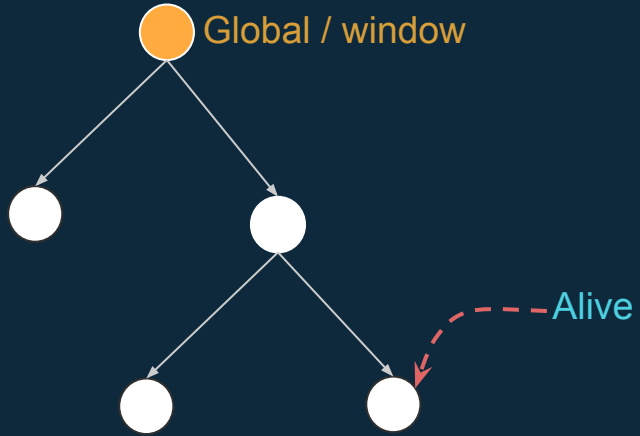




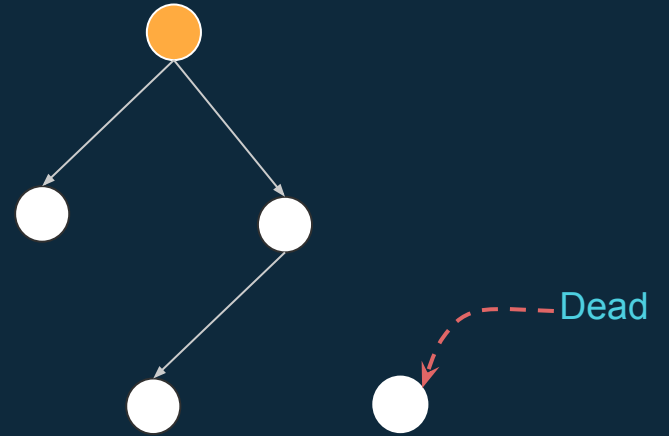
Dead or Alive ?

When an object is dead (not reachable) GC can remove it !

Cycle 1



Cycle N

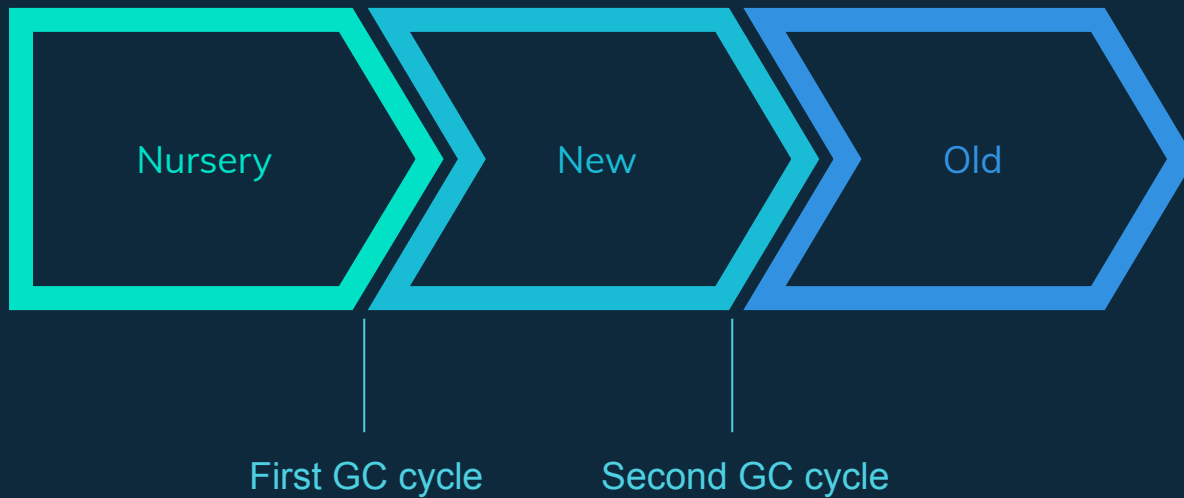


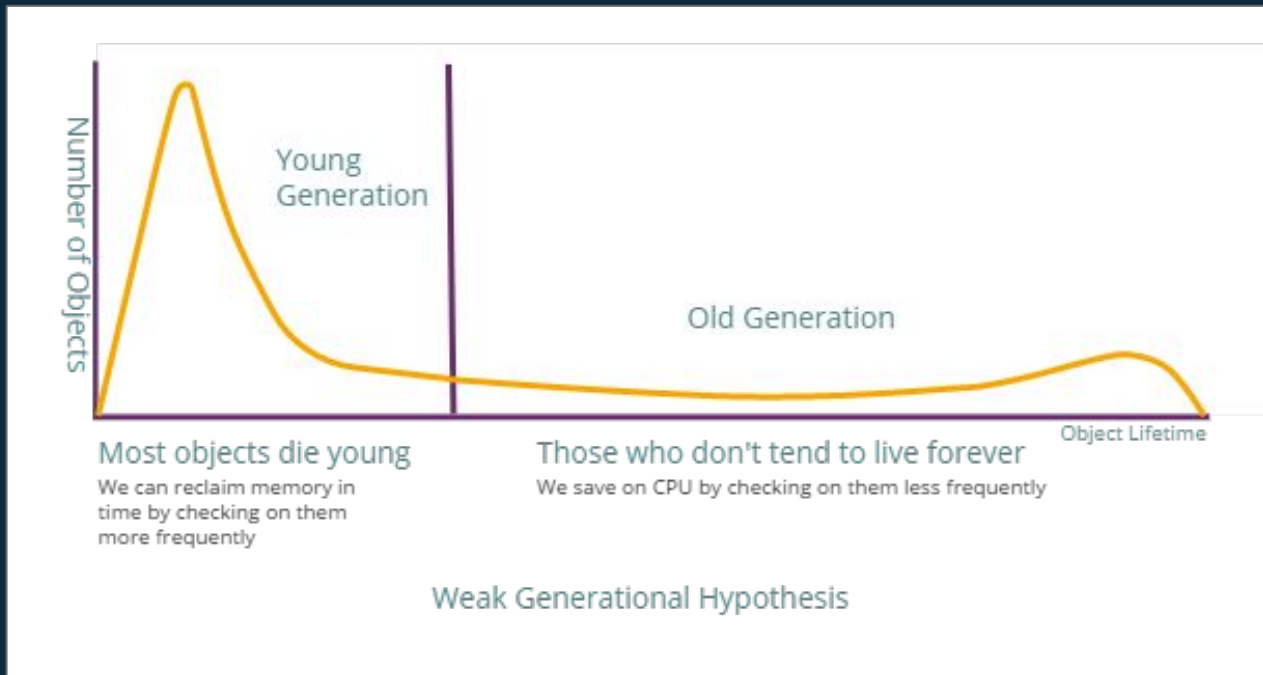
Time





Generations







Spaces

V8 uses spaces for a better memory organization !

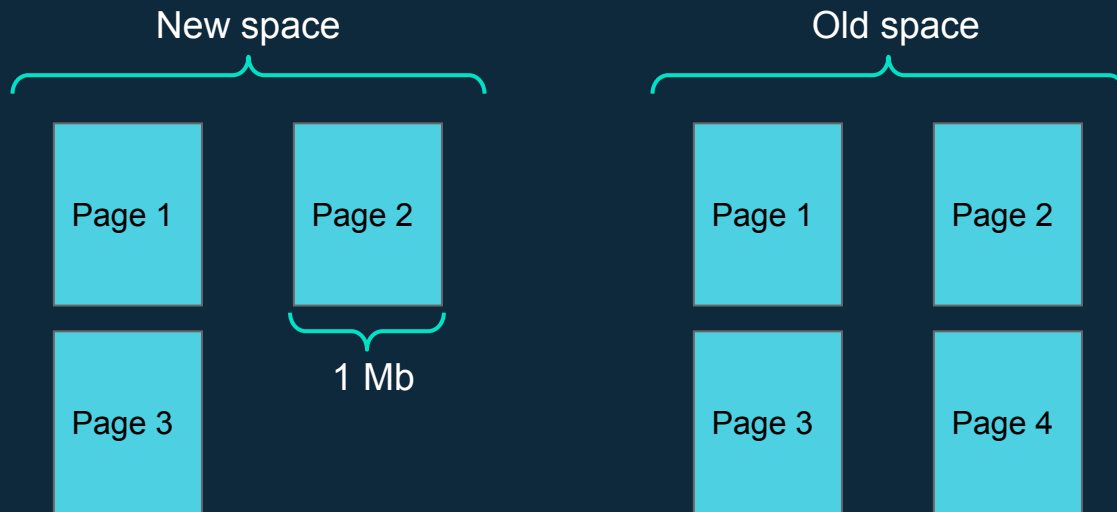


Spaces

Type	Description
New-space	Most objects are allocated here.
Old space	Moved here after surviving in new-space for a while.
Large space	This space contains objects which are larger than the size limits of other spaces. Large objects are never moved by the garbage collector.
Code space	Code objects, which contain JITed instructions, are allocated here.
Map space	Cells, PropertyCells, and Maps



Spaces & pages

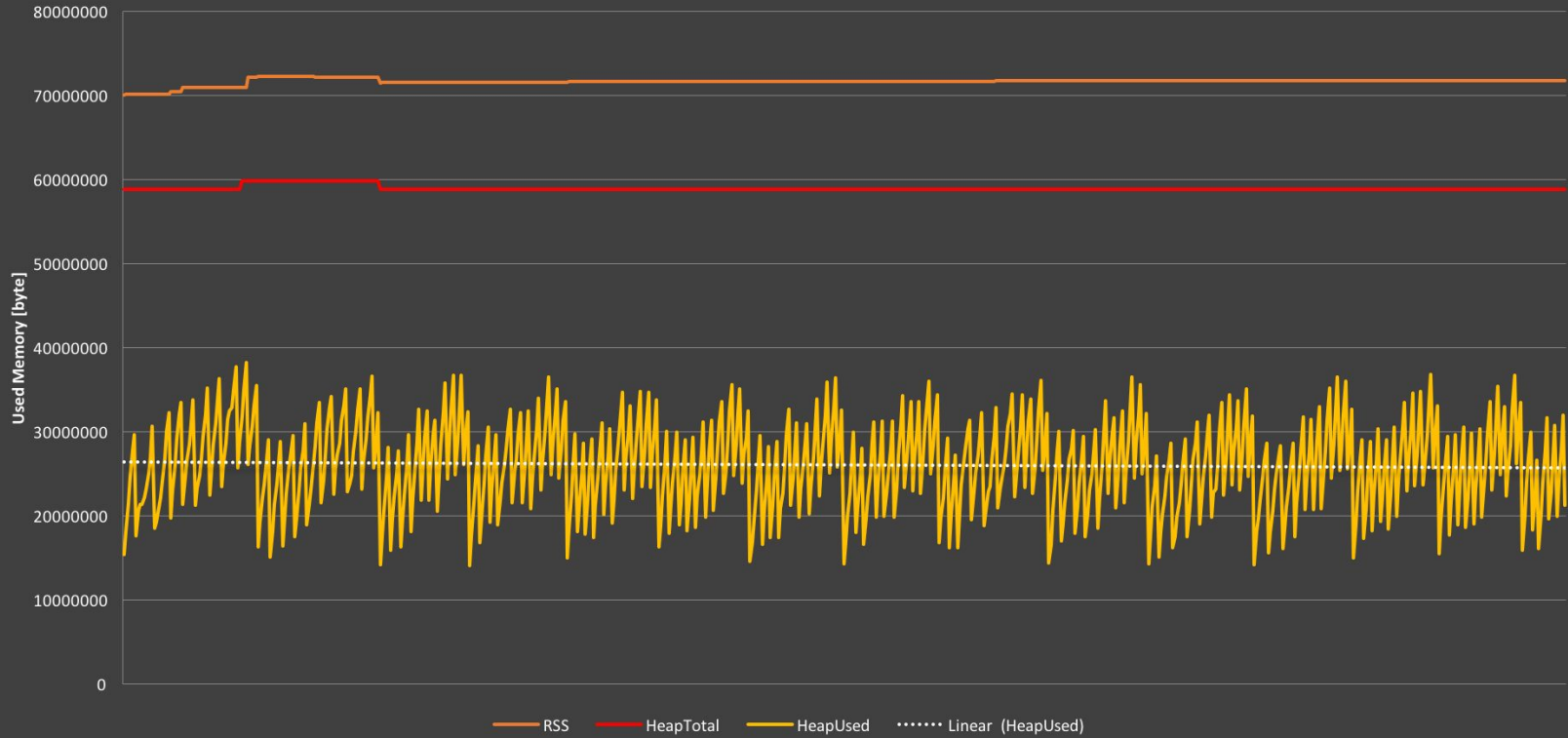




Memory Leak

Nightmare !

Node.js Memory Consumption over Time



Node.js Memory Consumption over Time



Resident Set

Code Segment

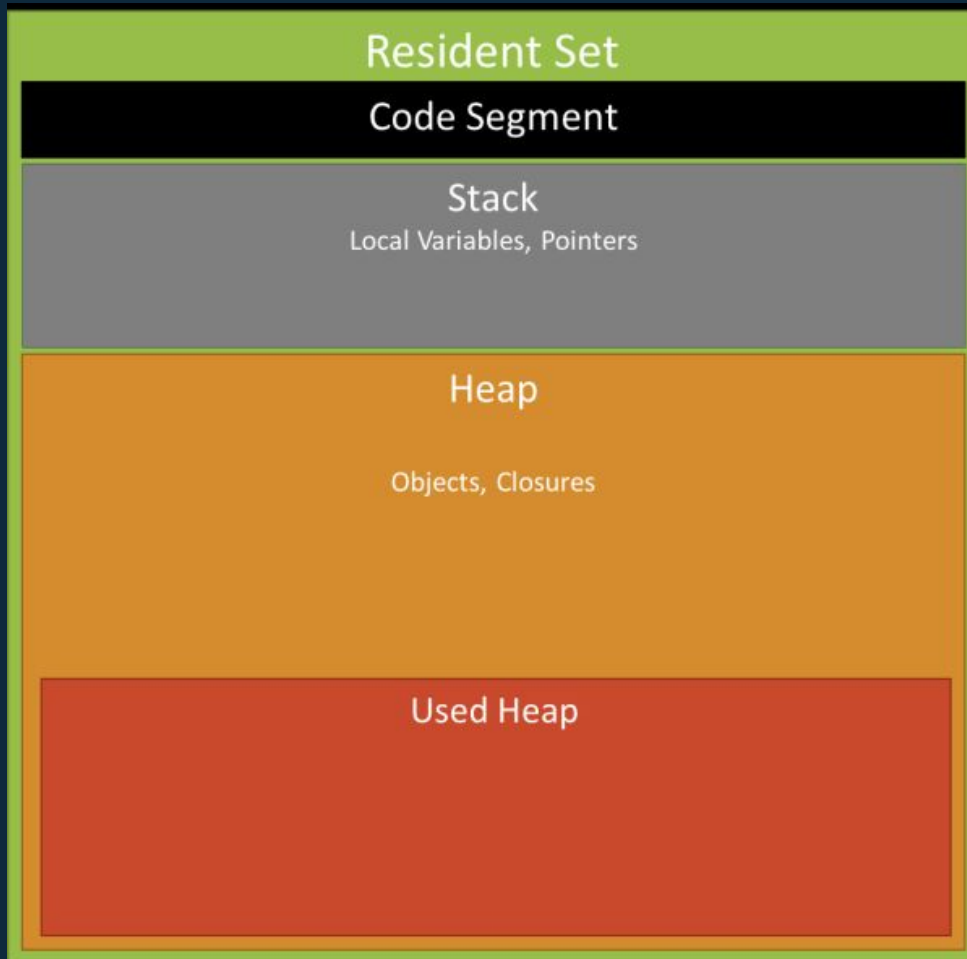
Stack

Local Variables, Pointers

Heap

Objects, Closures

Used Heap





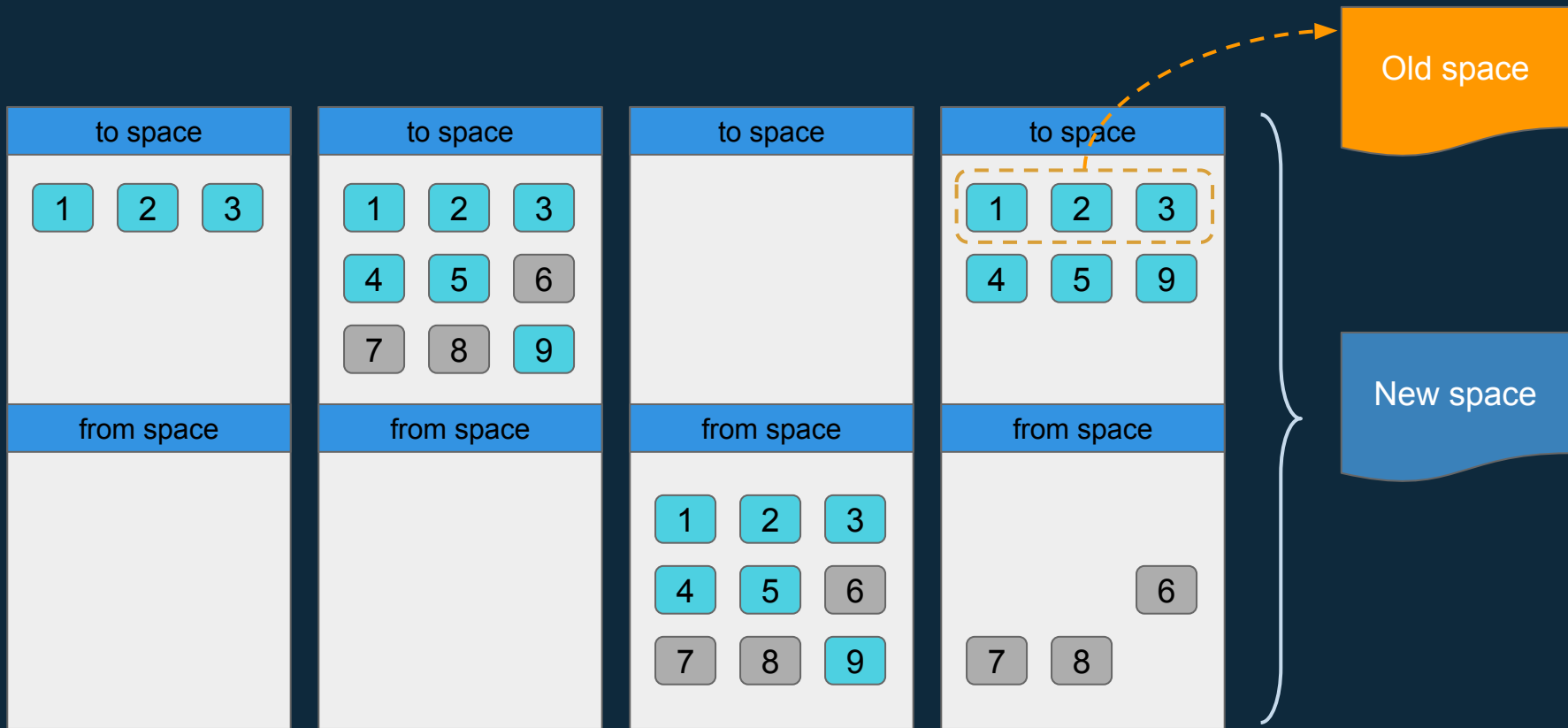
2

Algorithm

Scavenge vs Mark / Sweep / Compact
White -> gray -> back ...



Scavenge : Minor GC algorithm.
It's used only in new space.



Old space

New space



Major GC

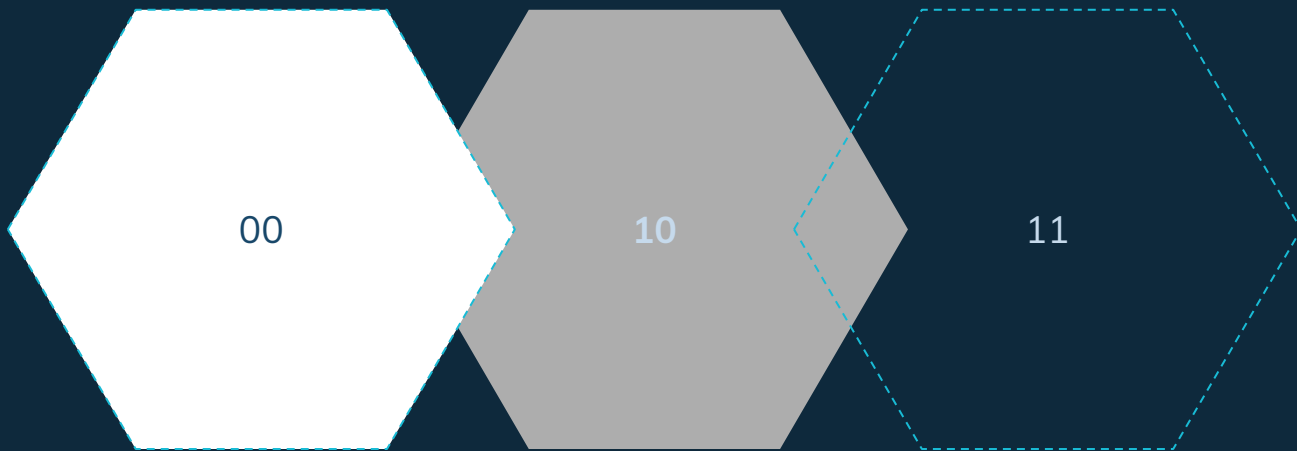




Mark : Recursive procedure of marking reachable objects !

V8 use the white/gray/black marking system.

White / Gray / Black

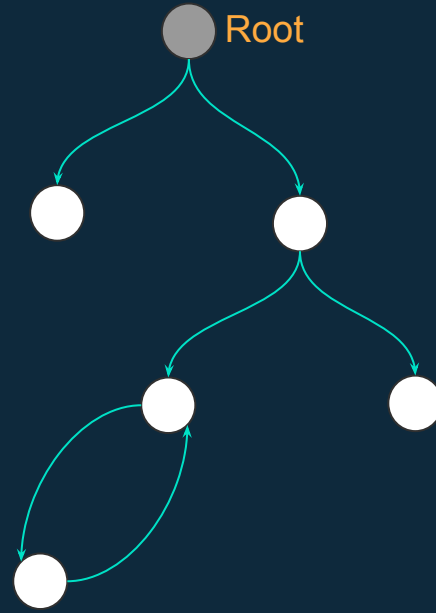
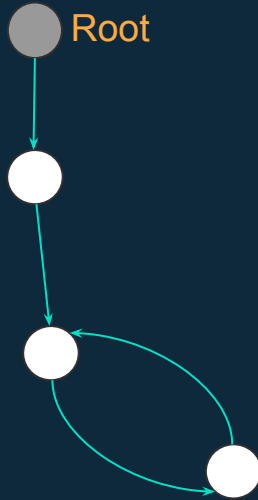
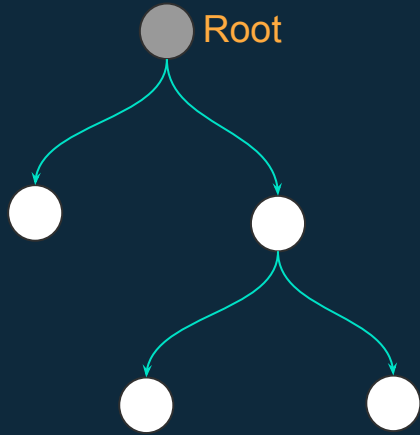


Not yet been discovered

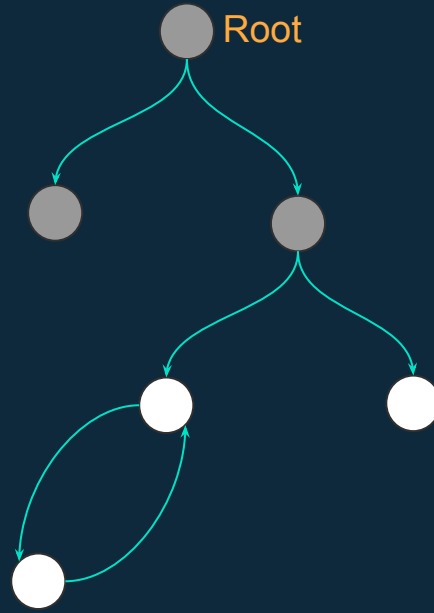
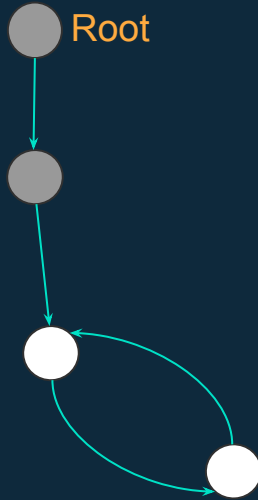
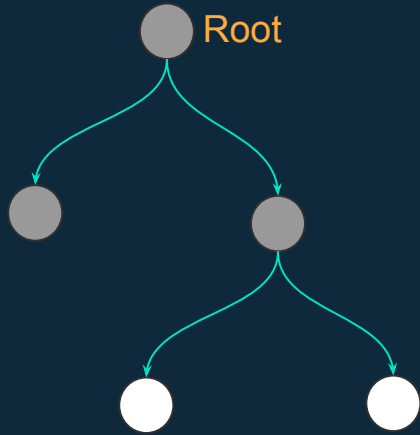
Discovered

Discovered and all of
its neighbors also.

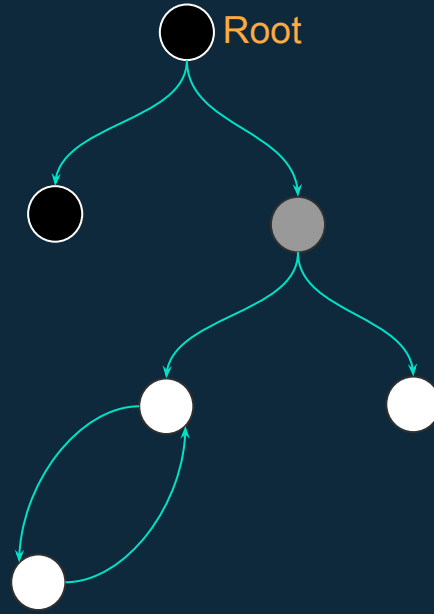
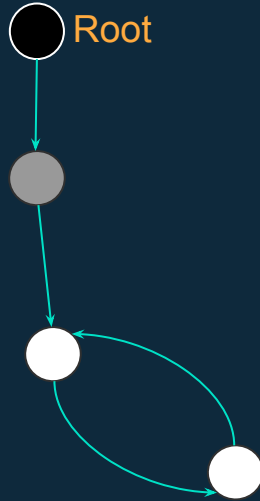
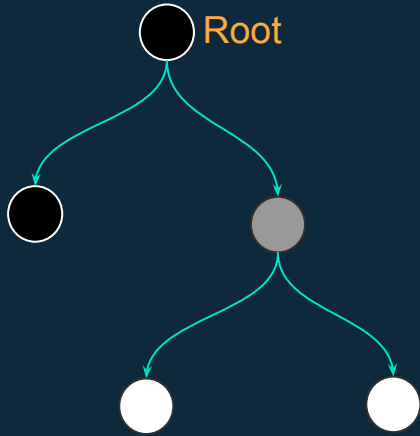
Define root nodes + mark them as discovered !



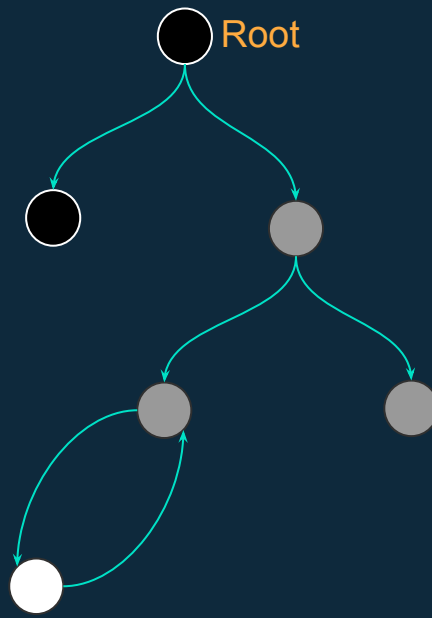
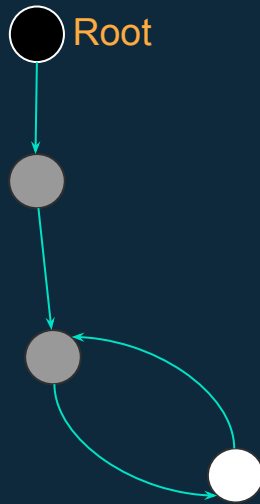
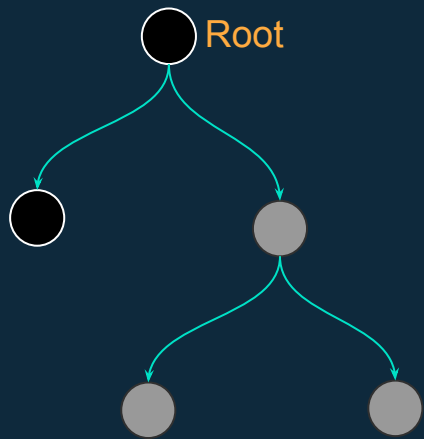
Direct neighbors discovery



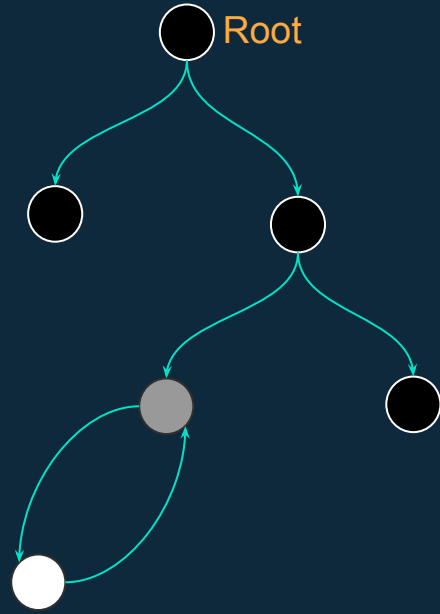
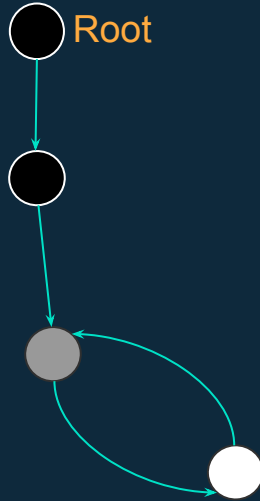
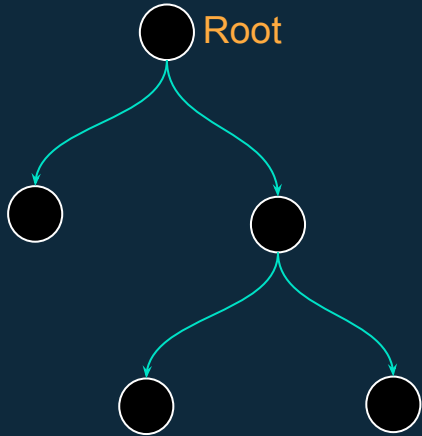
Mark node in black if they are discovered and all their neighbors also.



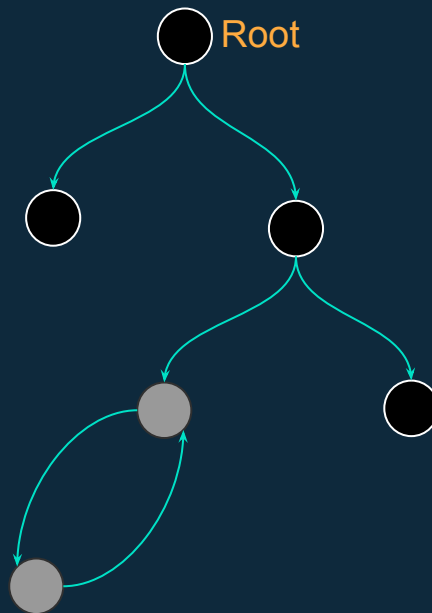
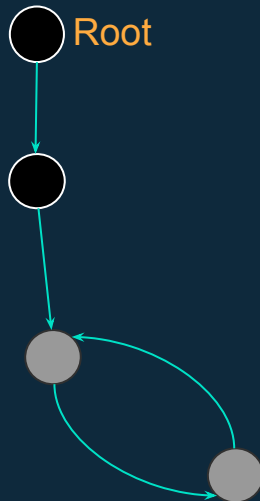
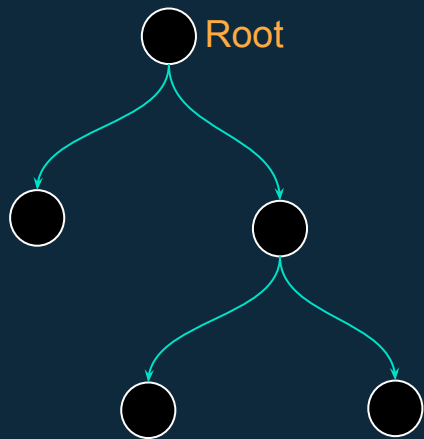
Mark node in black if they are discovered and all their neighbors also.



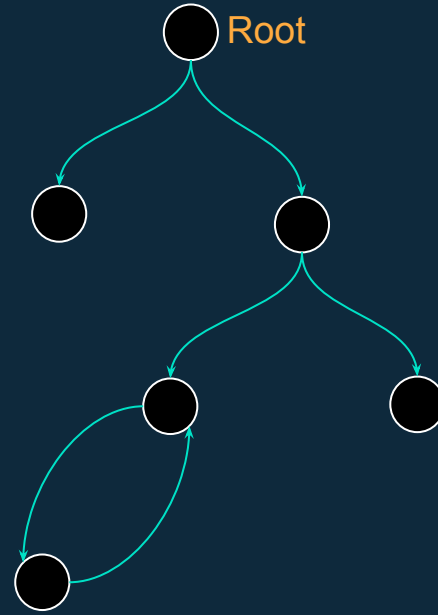
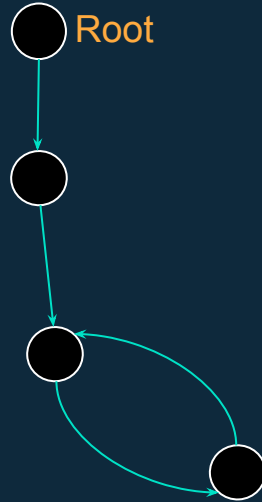
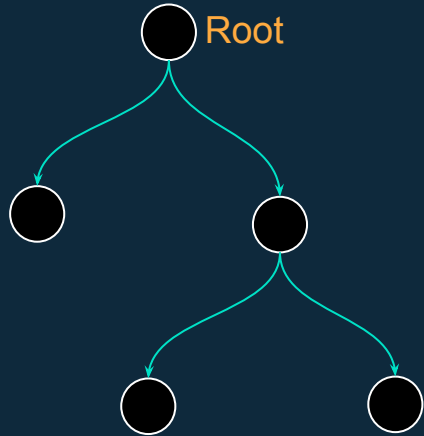
Mark node in black if they are discovered and all their neighbors also.



Mark node in black if they are discovered and all their neighbors also.



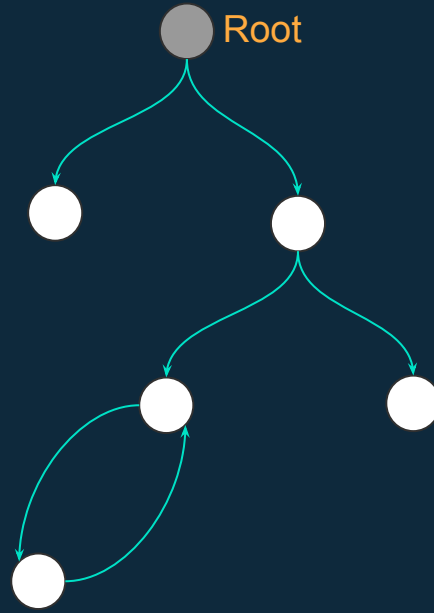
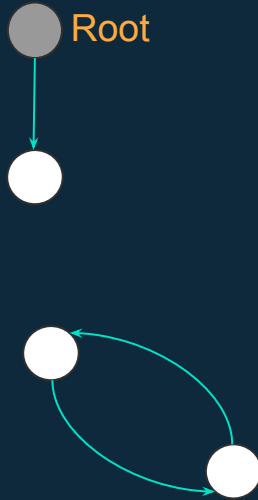
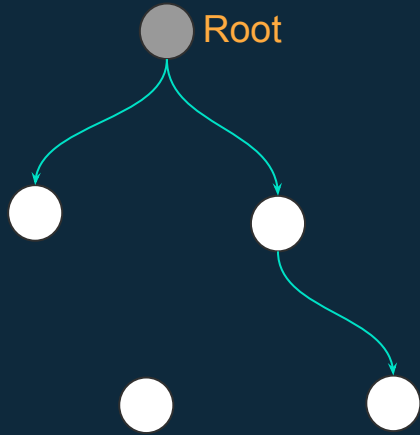
All reachable nodes are black : mark phase is done !!!!!



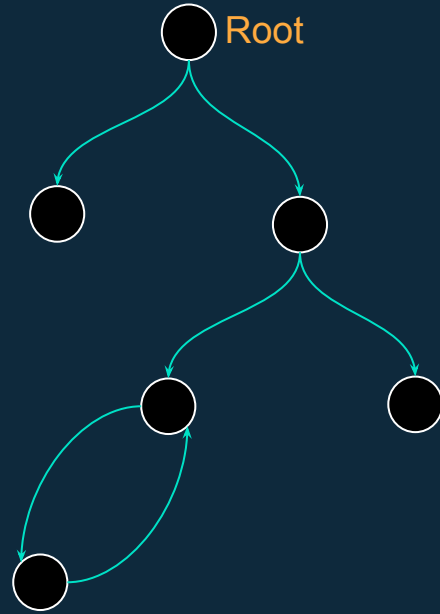
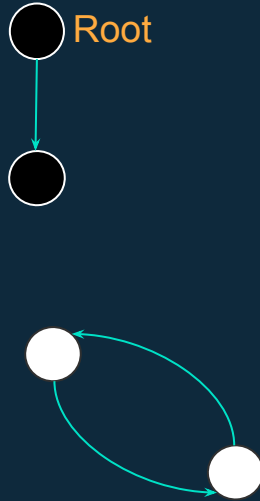
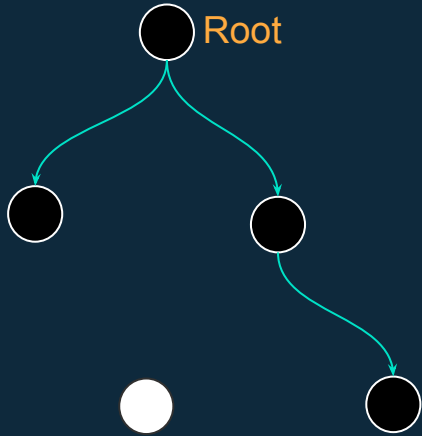


What about
non-reachable ?

Some nodes are unreachable !



Some nodes stay in white, they will be removed by sweep phase.





Sweep : Remove all unused (white) objects.

Can be done during another GC pause !



Compact : moving all marked – and thus alive – objects to the beginning of the memory region.



Sweep in action

Before



Sweep phase



After





Compact in action

Before Sweep



After Sweep



After Compact





3

Reduce marking pause

Parallel / concurrent



Stop the world





Incremental marking :

garbage collector splits up the marking work into smaller chunks



Incremental marking



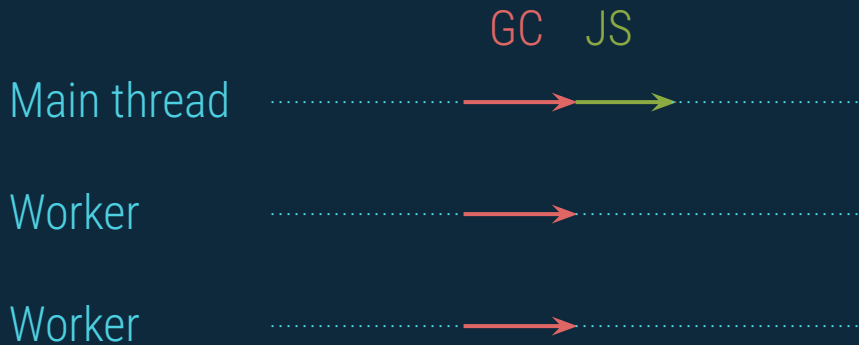


Lazy sweeping :

sweep pages on an as-needed basis
until all pages have been swept

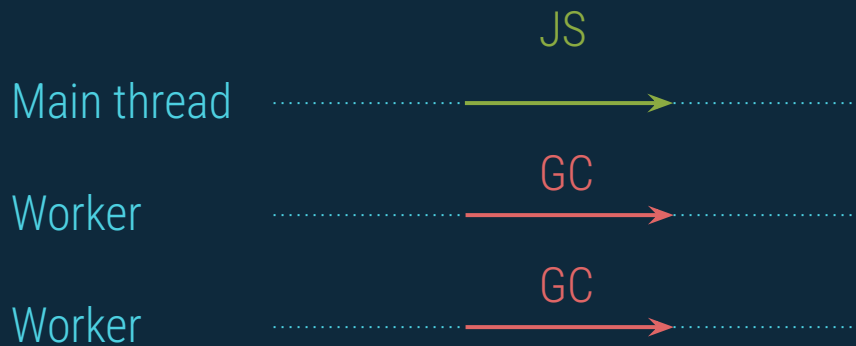


Concurrent



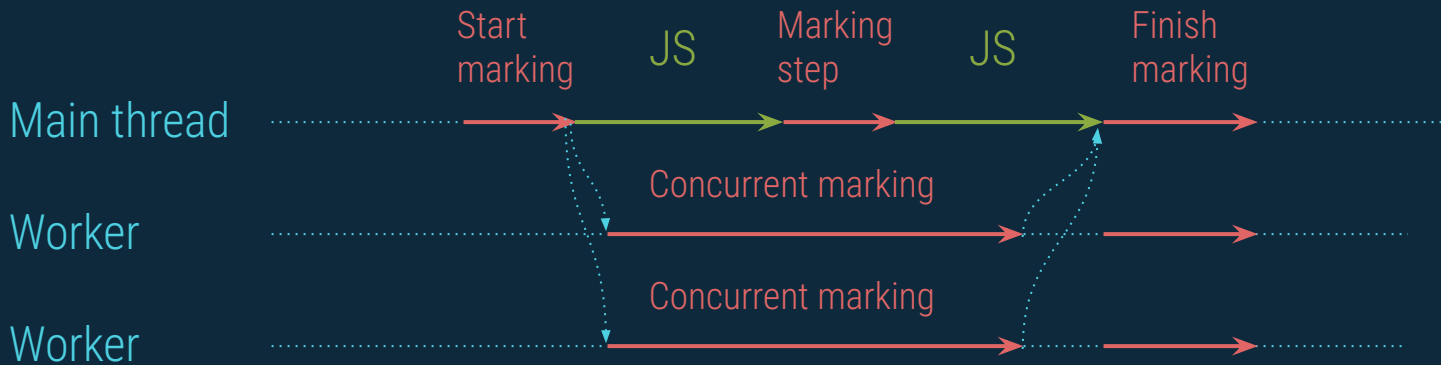


Parallel





All together





4

Monitoring tools

Modules : v8, gc stats

Options : --trace_gc



Demo

Module : V8

```
5 @link https://develop
6 * @package _s
7 */
8
9
10 if ( ! function_exists( 'incode_starter_setup' ) ) :
11 /**
12  * Sets up theme defaults and registers support for various WordPress features.
13  *
14  * Note that this function is hooked into the after_setup_theme hook, which
15  * runs before the init hook. The init hook is too late for some features, such
16  * as indicating support for post thumbnails.
17  */
18 function incode_starter_setup() {
19     /* Make theme available for translation.
20     * Translations can be filed in the 'languages/' directory of the theme.
21     * If you're building a theme based on incode_starter, you can't skip this.
22     * To change the default language to something other than 'en-US' you can:
23     * 1. Use the 'i18n' filter to change the default language.
24     * 2. Use the 'load_textdomain_locale' filter to change the locale for 'incode_starter'.
25     * 3. Use the 'load_default_textdomain_locale' filter to change the locale for all
26     *    textdomains.
27     * See https://codex.wordpress.org/Internationalization for more information.
28     */
29     $locale = apply_filters( 'load_textdomain_locale', 'en_US', 'incode_starter' );
30     load_textdomain( 'incode_starter', get_theme_root() . "/languages/$locale/incode_starter-$locale.mo" );
31     load_default_textdomain( 'incode_starter' );
32 }
33
```





Module v8

The v8 module exposes APIs that are specific to the version of V8 built into the Node.js binary

- `v8.getHeapSpaceStatistics()`
- `v8.getHeapStatistics()`

```
{  
  total_heap_size: 7326976,  
  total_heap_size_executable: 4194304,  
  total_physical_size: 7326976,  
  total_available_size: 1152656,  
  used_heap_size: 3476208,  
  heap_size_limit: 1535115264,  
  malloced_memory: 16384,  
  peak_malloced_memory: 1127496,  
  does_zap_garbage: 0  
}
```





Gc Stats

Exposes stats about V8 GC after it has been executed.

```
var gc = (require('gc-stats'))();

gc.on('stats', function (stats) {
  console.log('GC happened', stats);
});
```


```
GC happened {
  startTime: 9426055813976,
  endTime: 9426057735390,
  pause: 1921414,
  pauseMS: 1,
  gctype: 1,
  before: {
    ...
  },
  after: {
    ...
  },
  diff: {
    ...
  }
}
```



```
5 @link https://dev
6 * @package _s
7 */
8
9
10 if ( ! function_exists( 'incode_starter_setup' ) ) :
11 /**
12  * Sets up theme defaults and registers support for various WordPress fea
13  *
14  * Note that this function is hooked into the after_setup_theme hook, which
15  * runs before the init hook. The init hook is too late for some features, such
16  * as indicating support for post thumbnails.
17  */
18 function incode_starter_setup() {
19     /* Make theme available for translation.
20     * Translations can be filed in the "languages/" directory, which
21     * you can find in the "languages/" directory of the theme. To see more
22     * on changing "incode_starter" to the name of the theme, visit:
23     * https://codex.wordpress.org/Internationalization
24     */
25     load_theme_textdomain( 'incode_starter', get_template_directory() . '/languages' );
26 }
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Demo
--trace_gc





V8 options

- `--trace_gc` : print one trace line following each garbage collection
- `--trace_gc_verbose` : print details following each garbage collection
- `--trace_gc_nvp` : print one detailed trace line in name=value format after each garbage collection
- `--expose_gc` : expose gc extension



GC pause / time spent in external memory

PID

Start GC type

PID	Start	GC type	GC pause / time spent in external memory	Reason of GC
[29347:0x3c18b10]	2254 ms	Scavenge	114.8 (144.3) -> 113.0 (152.8) MB, 15.5 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2274 ms	Scavenge	122.4 (152.8) -> 121.9 (159.3) MB, 15.3 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2298 ms	Scavenge	128.3 (159.3) -> 127.7 (168.8) MB, 22.6 / 0.1 ms	allocation failure
[29347:0x3c18b10]	2318 ms	Scavenge	137.0 (168.8) -> 136.6 (175.3) MB, 15.4 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2344 ms	Scavenge	143.0 (175.3) -> 142.3 (184.8) MB, 22.2 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2360 ms	Scavenge	151.7 (184.8) -> 151.3 (191.3) MB, 13.2 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2378 ms	Scavenge	157.6 (191.3) -> 157.1 (201.3) MB, 17.2 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2393 ms	Scavenge	166.3 (201.3) -> 165.8 (208.3) MB, 12.4 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2414 ms	Scavenge	172.3 (208.3) -> 171.7 (217.8) MB, 16.4 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2430 ms	Scavenge	180.9 (217.8) -> 180.3 (225.3) MB, 11.1 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2470 ms	Scavenge	187.0 (225.3) -> 186.2 (234.3) MB, 14.6 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2510 ms	Mark-sweep	195.0 (234.3) -> 67.2 (103.3) MB, 3.9 / 0.0 ms (+ 53.5 ms in 188 ms since start of marking 96 ms)	via stack guard GC in old space requested
[29347:0x3c18b10]	2542 ms	Scavenge	74.2 (103.3) -> 73.7 (108.3) MB, 26.3 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2558 ms	Scavenge	82.3 (108.3) -> 81.8 (116.3) MB, 9.7 / 0.0 ms	allocation failure
[29347:0x3c18b10]	2584 ms	Scavenge	88.8 (116.3) -> 88.1 (125.3) MB, 23.0 / 0.0 ms	allocation failure

Reason of GC

Size of all objects



Thanks!

Any questions?

You can find me at:

- ◇ @Vince_Vallet
- ◇ wallet77@gmail.com



Credits

- ◇ V8 GC overview
- ◇ Concurrent marking
- ◇ GC algorithm
- ◇ V8 options