Linked Weak Reference Arrays

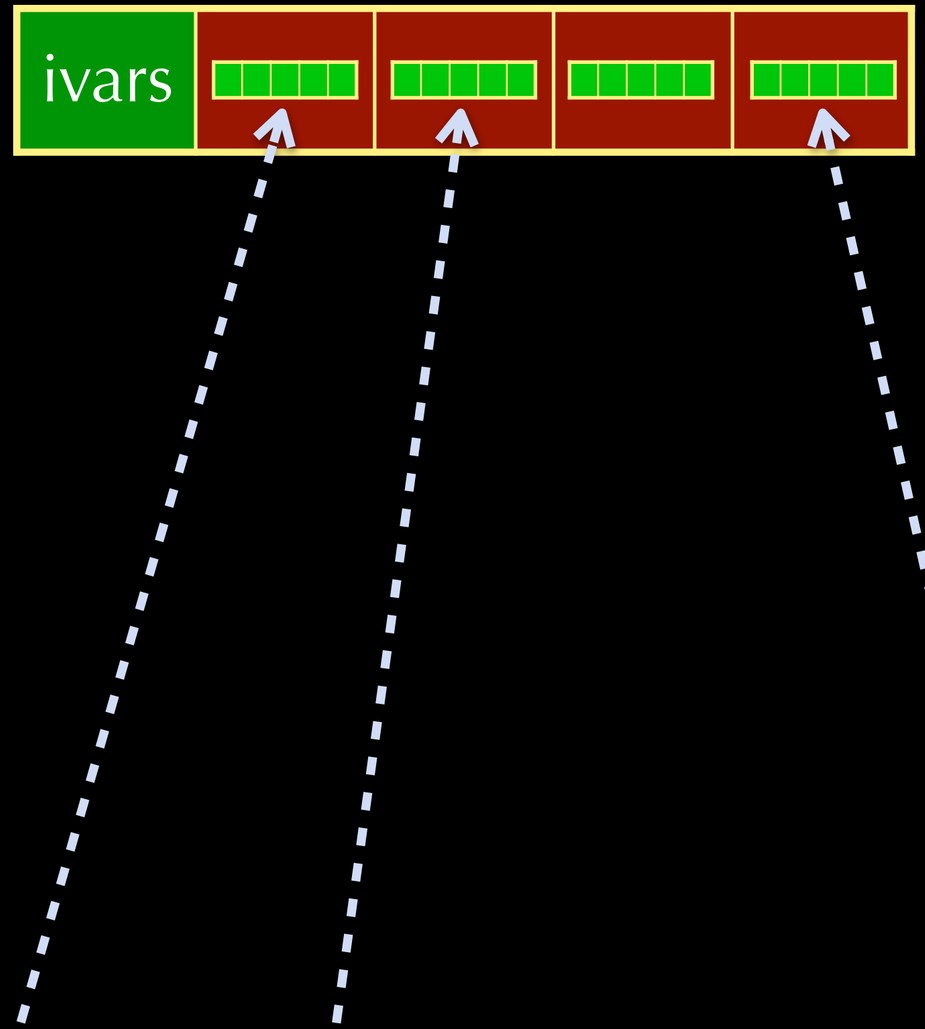
A Hybrid Approach to Efficient Bulk Finalization

Andrés Valloud

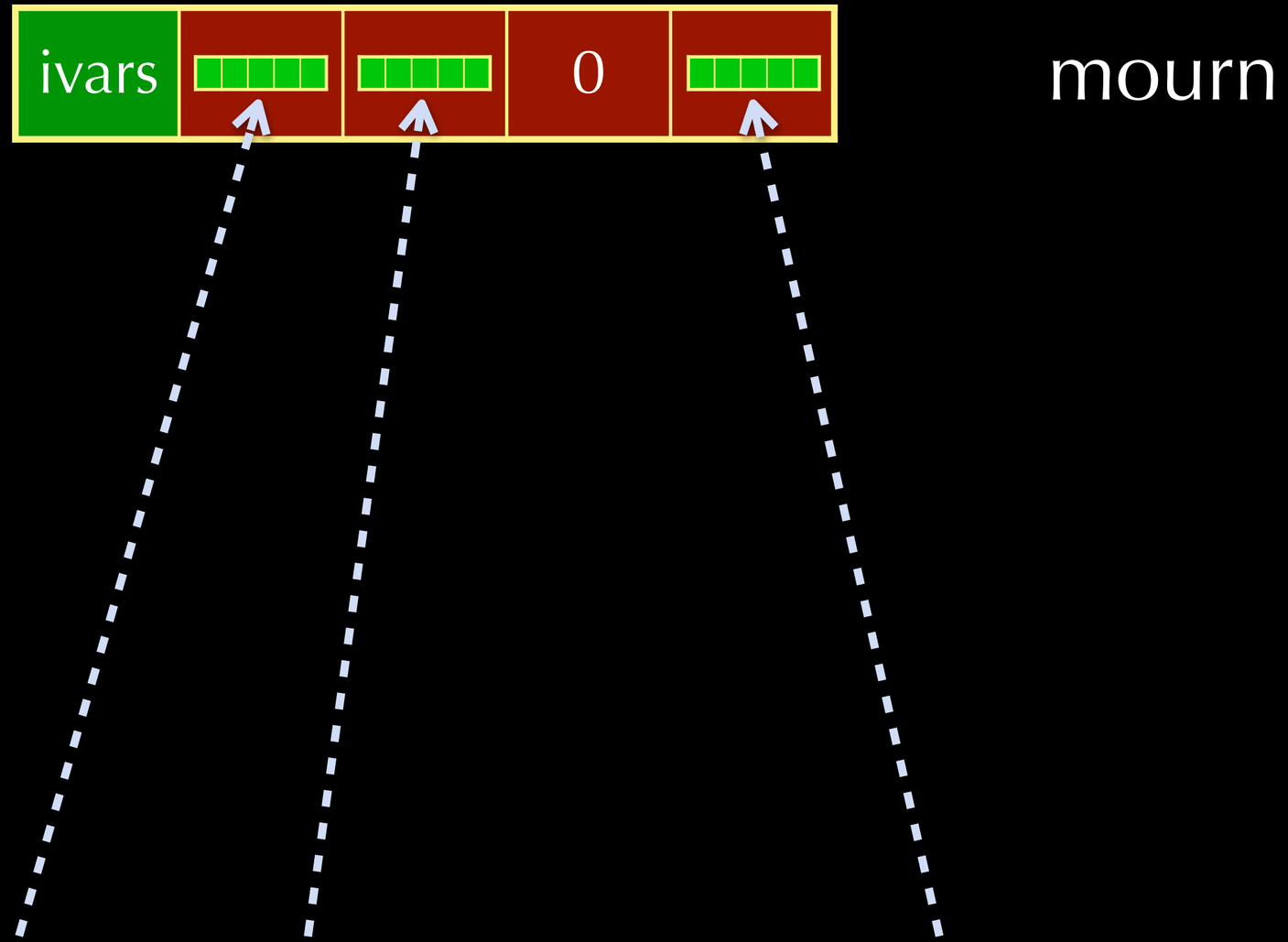
IWST 2015

Weak arrays

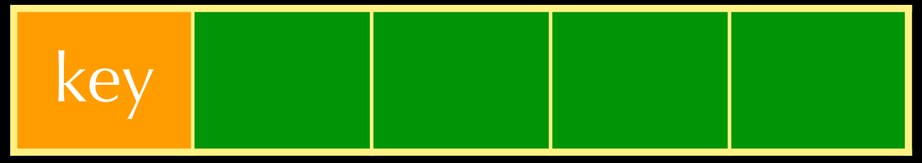
Weak arrays



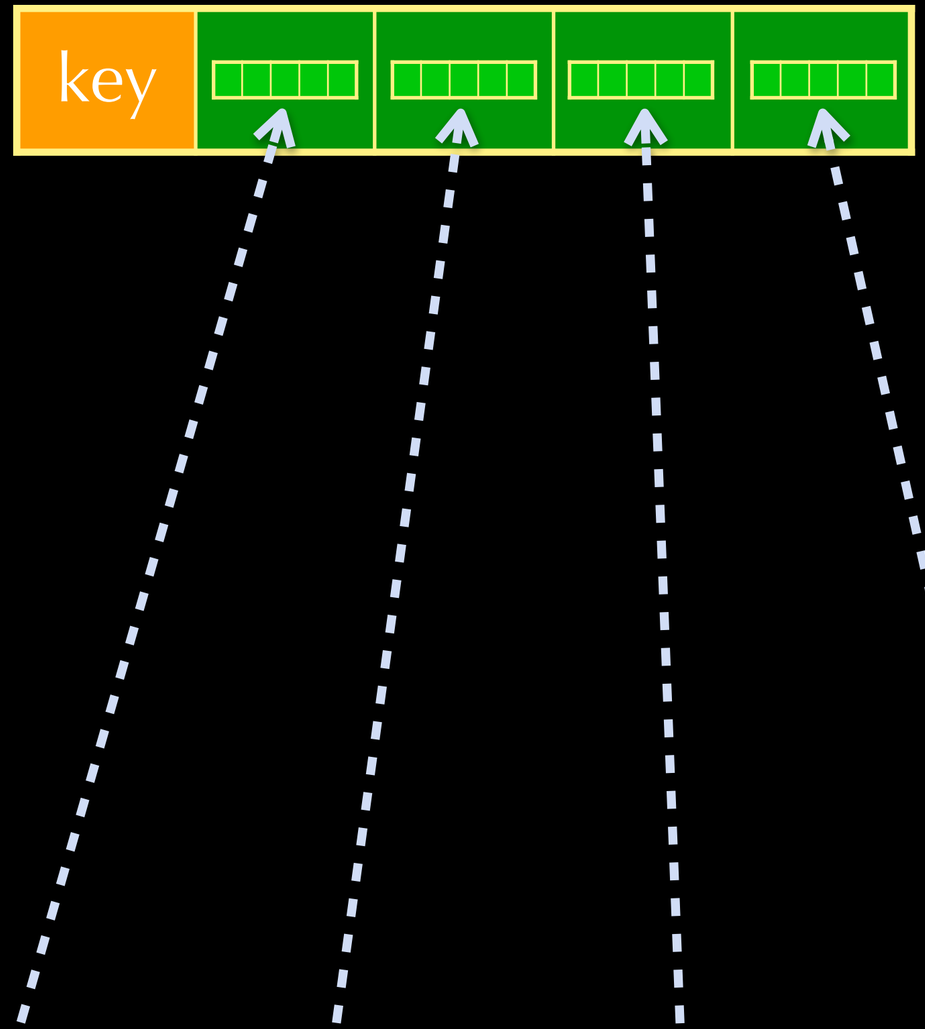
Weak arrays



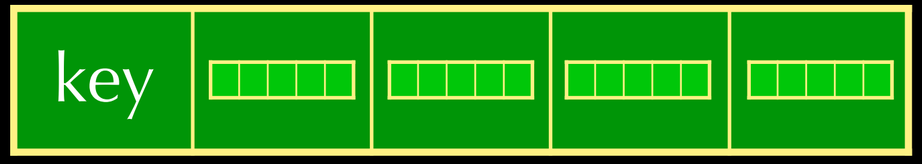
Ephemerons



Ephemerons

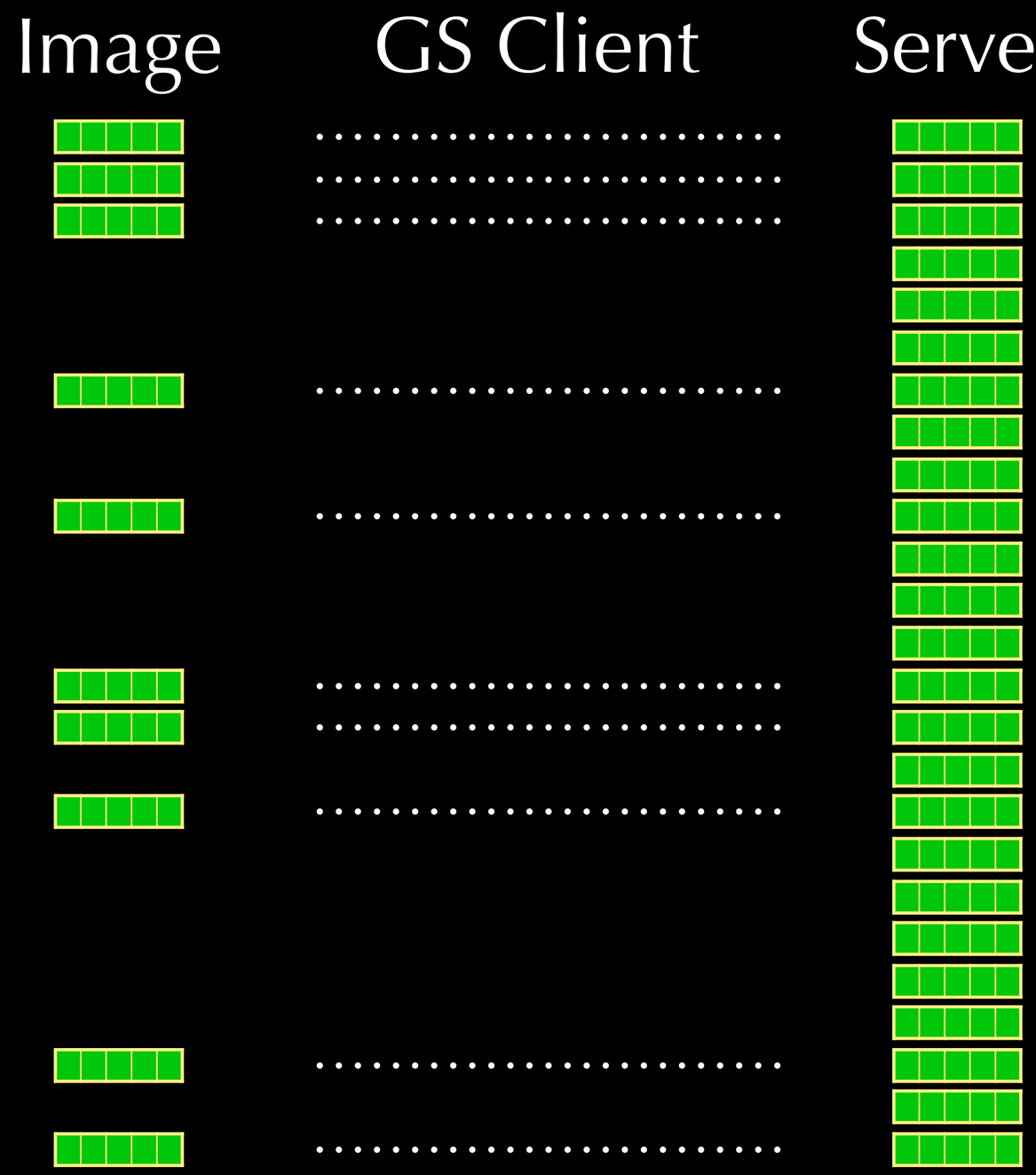


Ephemerons



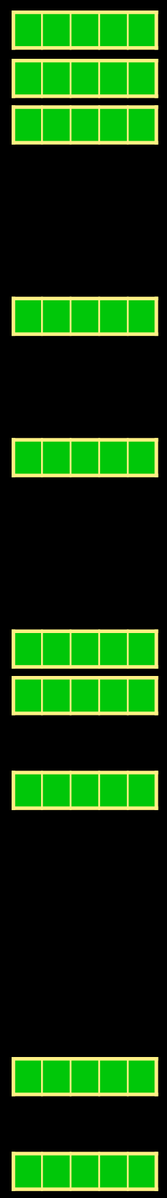
mourn

GemStone/S

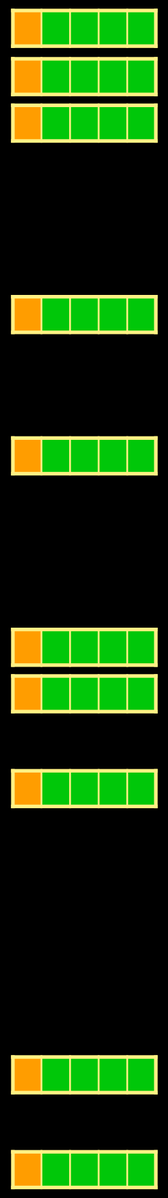


GemStone/S

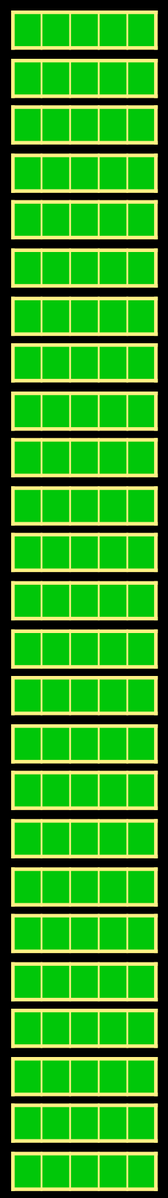
Image



Ephemerons

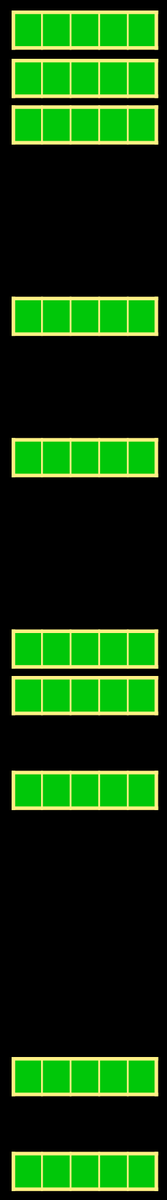


Server

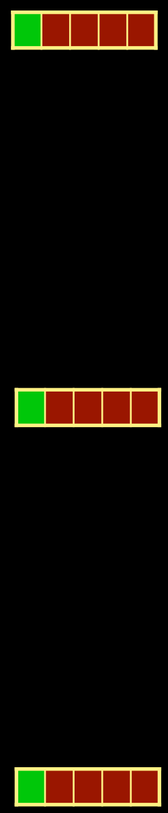


GemStone/S

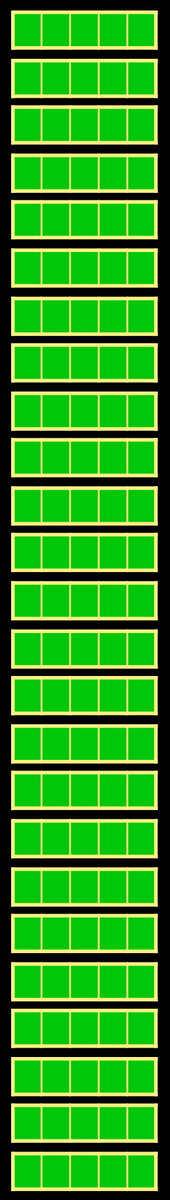
Image



Weak arrays



Server

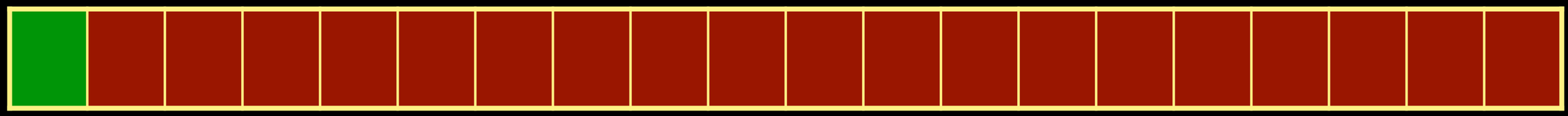


Observation overhead

	Ephemerons	Weak arrays	Evaluation
Space required per client object	at least one extra object header	negligible	weak arrays much more compact
VM cost per client object	one extra finalization queue slot	negligible	weak arrays much more efficient in bulk
Finalization cost per client object	ephemerons finalized only as needed	scanning for tombstones	weak arrays induce linear search

No scanning, deployed

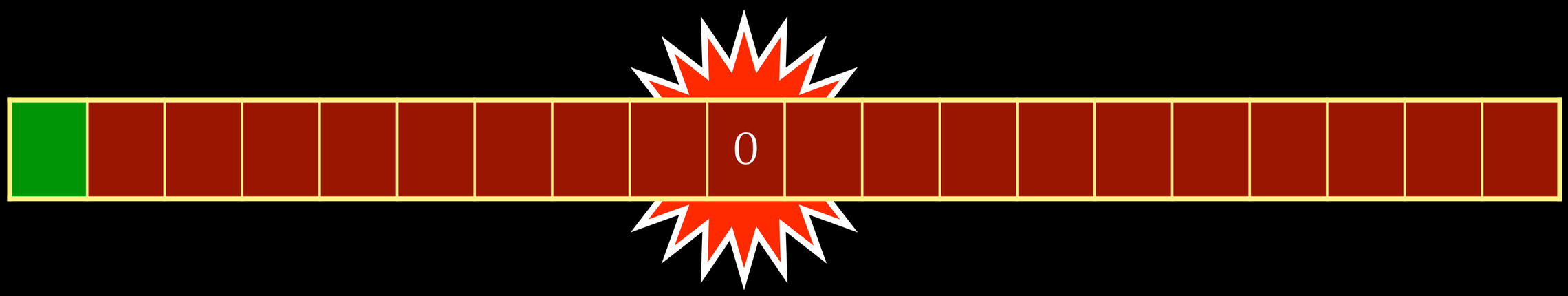
Begin interaction



End interaction

No scanning, deployed

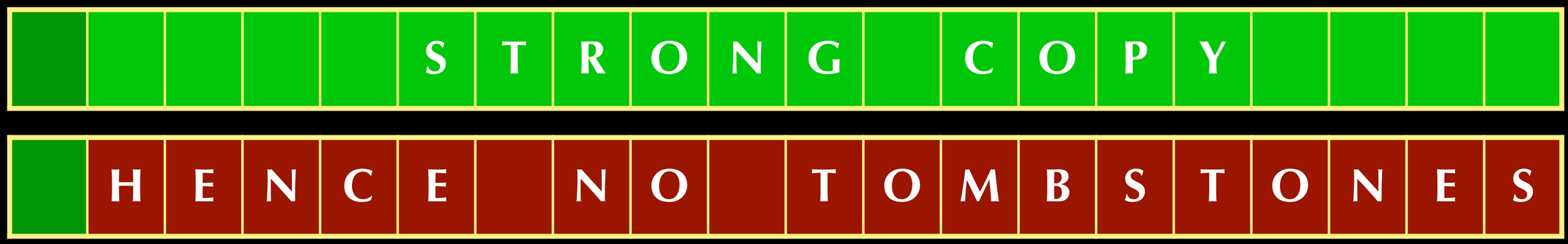
Begin interaction



End interaction

No scanning, deployed

Begin interaction

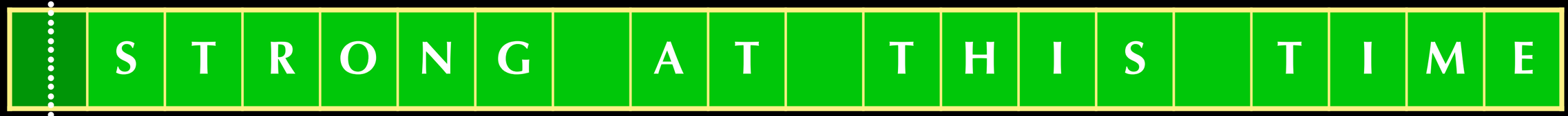


End interaction

No scanning, deployed

Begin interaction

isWeakContainer: false

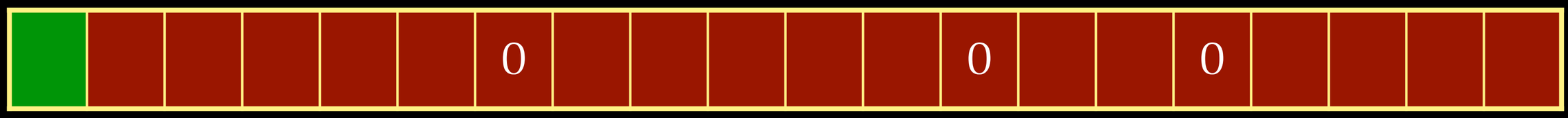


isWeakContainer: true

End interaction

No scanning, proposed

Begin mourning

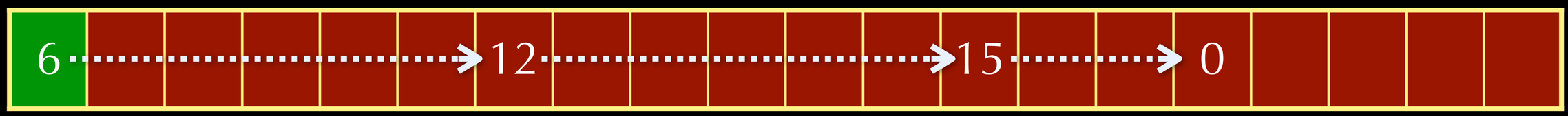



```
WeakArray>>mourn
    1 to: self size do:
      [:eachIndex |
       (self at: eachIndex) == 0
         ifTrue: [self mournAt: eachIndex]
      ]
```

End mourning

No scanning, proposed

Begin mourning

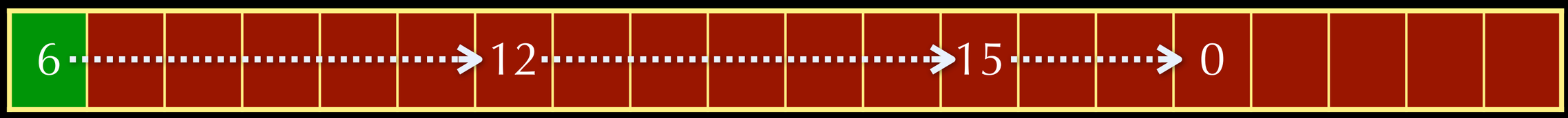


LinkedWeakArray

End mourning

No scanning, proposed

Begin mourning




```
LinkedWeakArray>>mourn
    | nextIndex |
    nextIndex := self firstTombstoneIndex. "6"
    [nextIndex == 0] whileFalse:
     [
      self mournAt: nextIndex.
      nextIndex := self at: nextIndex          "12, 15, 0"
     ]
```

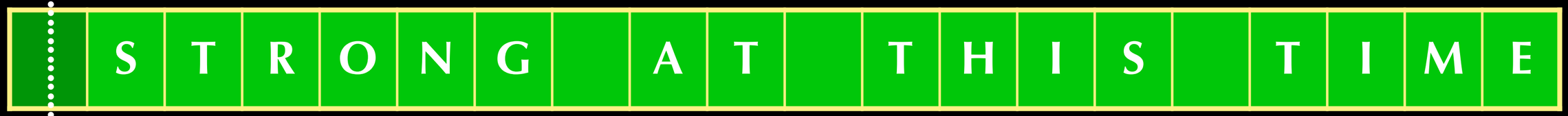
End mourning

No scanning, proposed

Begin GC

VM queues for finalization

setWeakContainer(false);



LinkedWeakArray>>mourn

isWeakContainer: true

End mourning

Mourning performance I

Quotient

11.0
10.4
9.7
9.1
8.4
7.8
7.1
6.5
5.8
5.2
4.5
3.9
3.2
2.6
1.9
1.3
0.6
0.0

1
2
3
5
7
10

Tombstoning rates

Mourning performance II

Log quotient

1E+06

1E+04

1E+02

1E+00

Linked weak array sizes

2^20

2^14

2^8

2^4

Tombstoning rates

574583

188277

77116

31585

12935

5296

2167

886

361

146

58

22

7

1

Questions