

I

I E w p k p i

j w r o c u j k l p i e o

Tgxkuk p

-
- 以
- Gf gp
 - wtxlx t

- 以
- Vgpwtgf

r t dno hvjg rf l Eu

gtknl E

Rctcnl E

RP - EO

YO

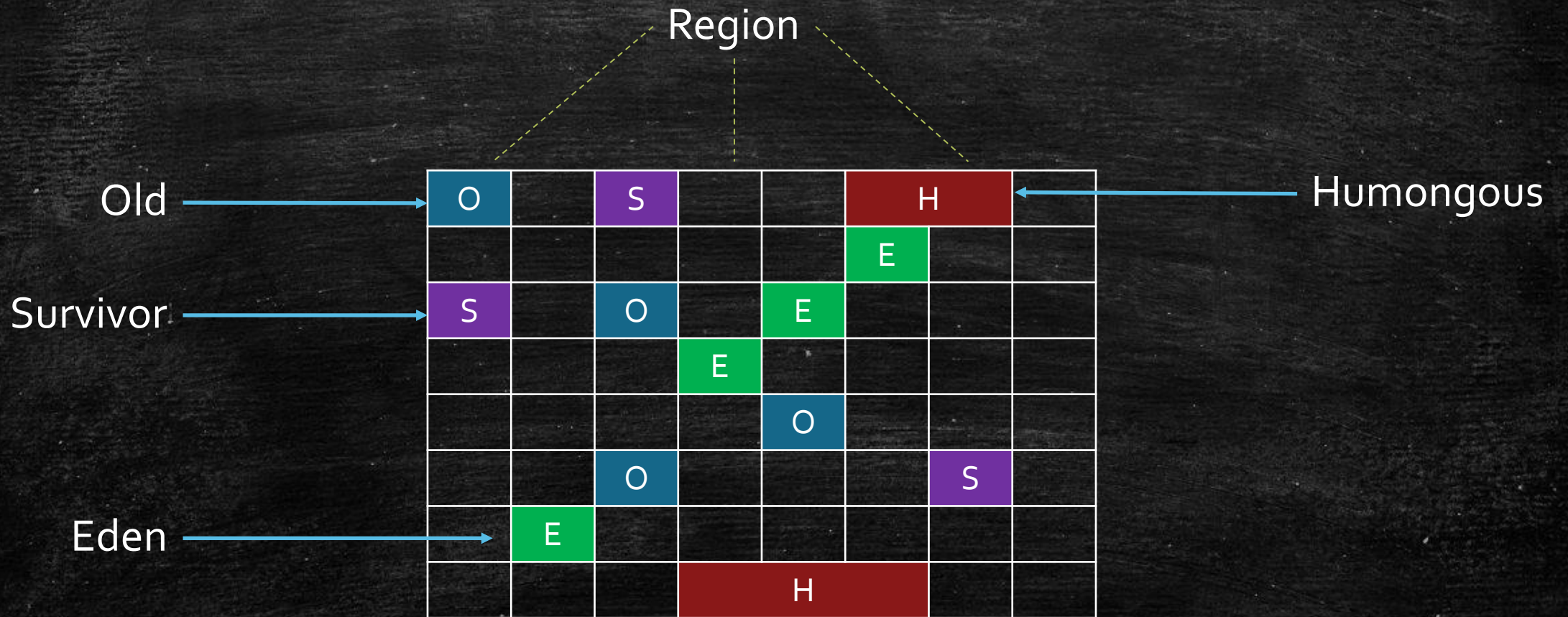




I E

I E

I



—
G₁

G₁

—

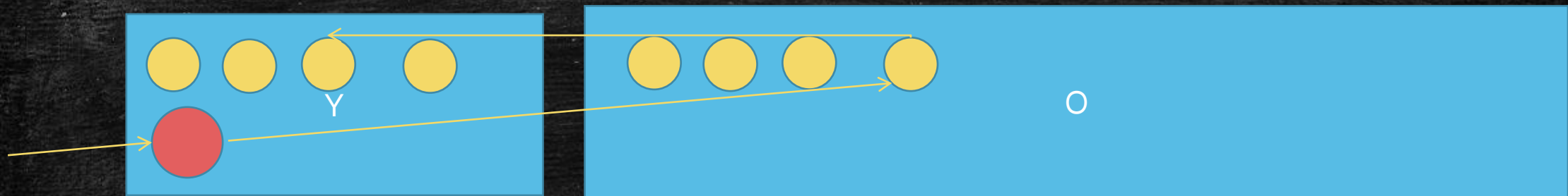
G₁

1M

32M

2

存 card table



存

CSet = Collection Set

CSet
CSet
CSet

GC
Eden

1%

survivor

RSet = RememberedSet



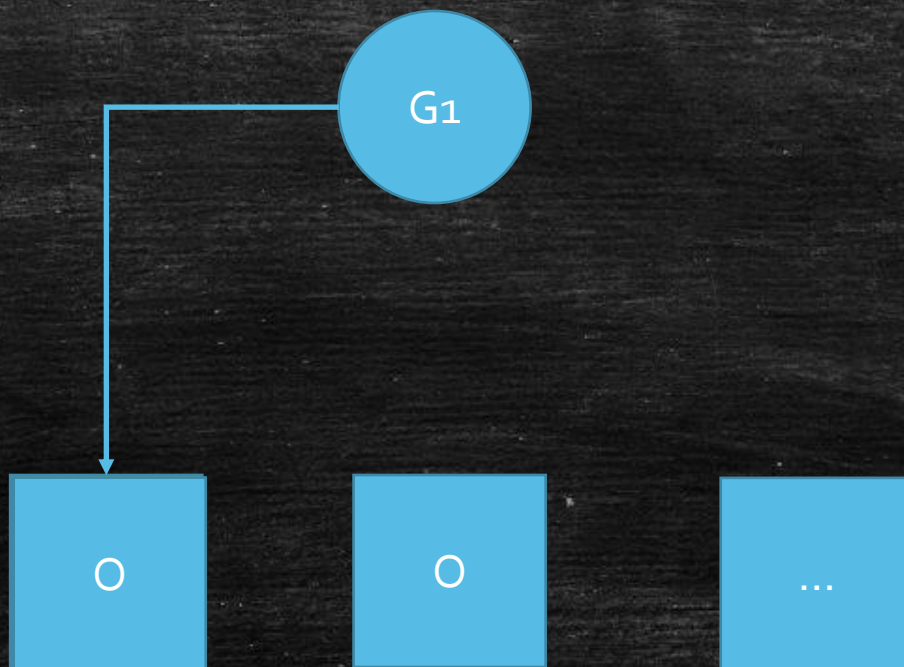
XO

比

uguuk p dcugf I E

I

G



Y j l

- er w

- cr r l E

-

- ZZ Oczi E Rc wug Oknu

- VY

- Tgi k p

- 会

Tgi k p

比 Tgi k p服

j gcf r Tgi k p er r

-

- ZZ I J gcr Tgi k p k g

```
// Minimum region size; we won't go lower than that.  
// We might want to decrease this in the future, to deal with small  
// heaps a bit more efficiently.  
#define MIN_REGION_SIZE ( 1024 * 1024 )  
  
// Maximum region size; we don't go higher than that. There's a good  
// reason for having an upper bound. We don't want regions to get too  
// large, otherwise cleanup's effectiveness would decrease as there  
// will be fewer opportunities to find totally empty regions after  
cleanup.  
#define MAX_REGION_SIZE ( 32 * 1024 * 1024 )  
  
round this // The automatic region size calculation will try to have a  
// many regions in the heap (based on the min heap size).  
#define TARGET_REGION_NUMBER 2048
```


T gv

教

T gv

比

T gv

I E

P

lxgt

wngv

以

/

-

-

|

j wo pi wu d gev

tgi k p



I E

I E

- G gp

-

H E

- rf

- ugo i e +

I Okzgf I E

EO

ZZ plcvpi J gcr eewr ce Rgtegpv

-

-

Okzgf I E

Okzgf I E

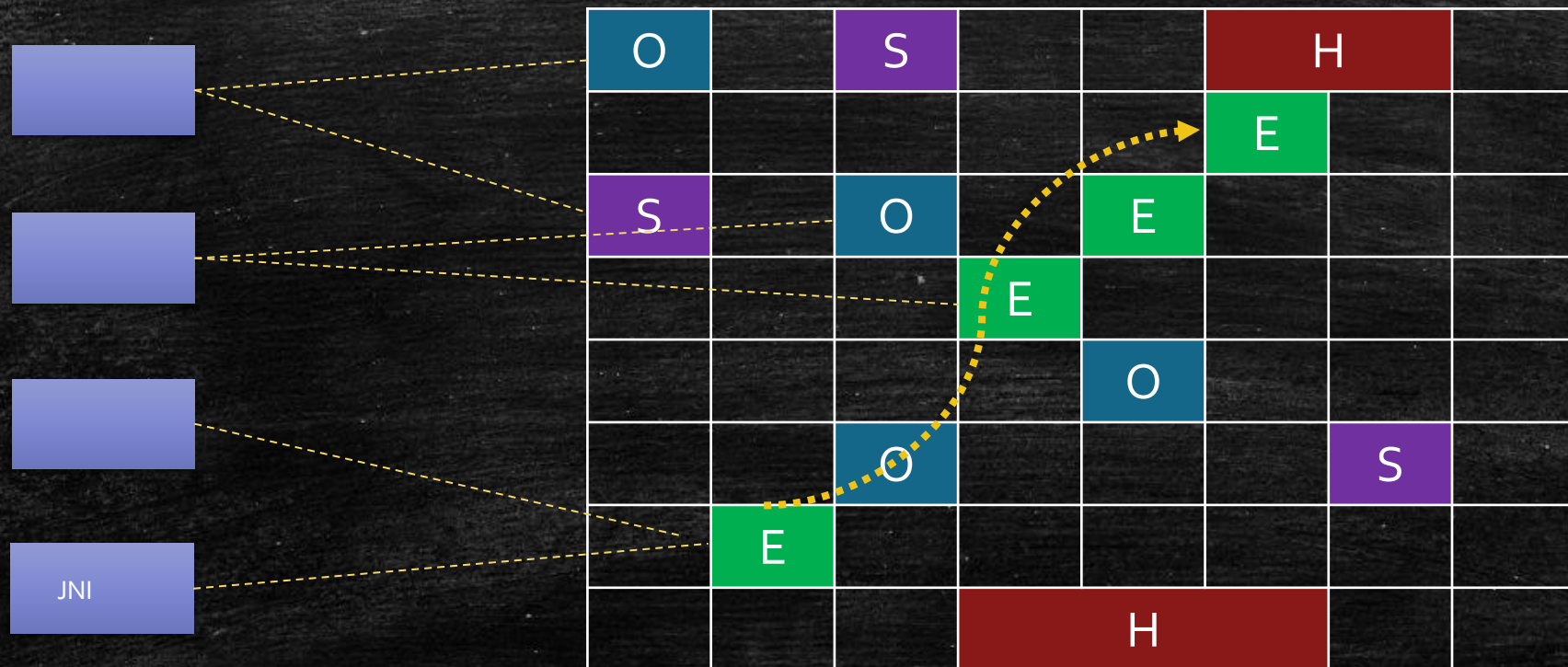
VY

有

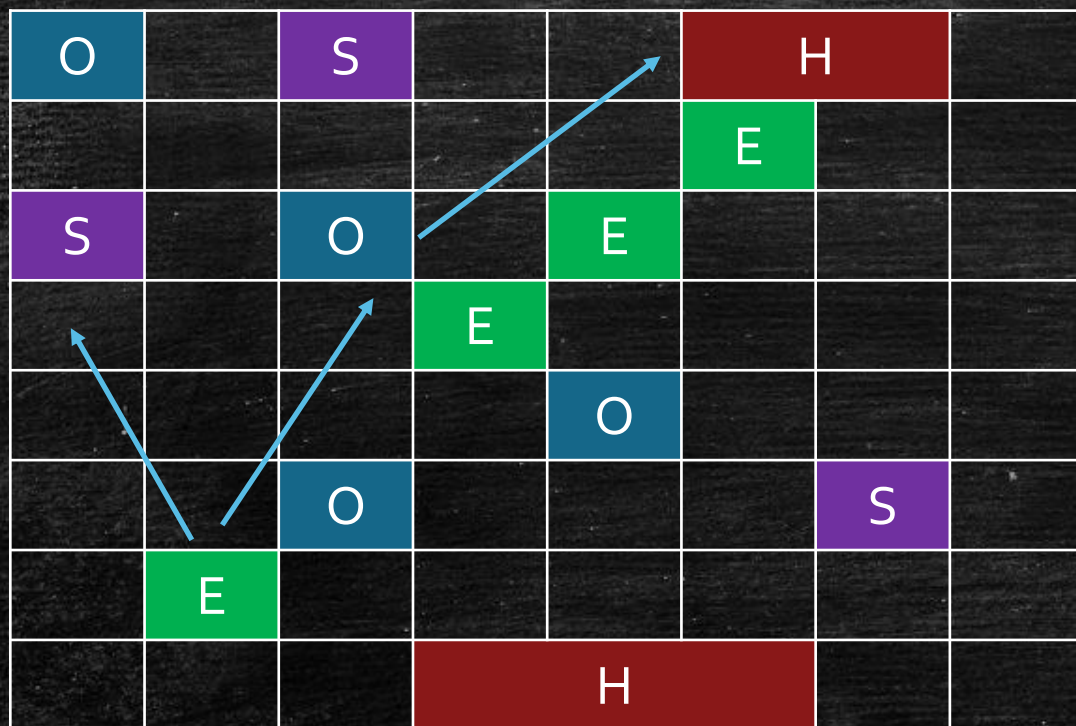
VY

VY

GC roots



有



| | | | | | | |
|---|---|---|---|---|---|---|
| O | | S | | | H | |
| | | | | | E | |
| S | | O | | E | | |
| | | | E | | | |
| | | | | O | | |
| | | O | | | | S |
| | E | | | | | |
| | | | H | | | |



G1 Full GC

java 10

FullGC

FullGC

y jcvkul

y j l

j y l y tma

|

1. 10ms
- 2.
3. M T

1. concurrent
2. region-based
3. compacting
4. NUMA-aware
5. colored pointers
6. load barriers

I E