Cincom Smalltalk™ Object Memory Management

Andrés Valloud

Arrangements

Eden

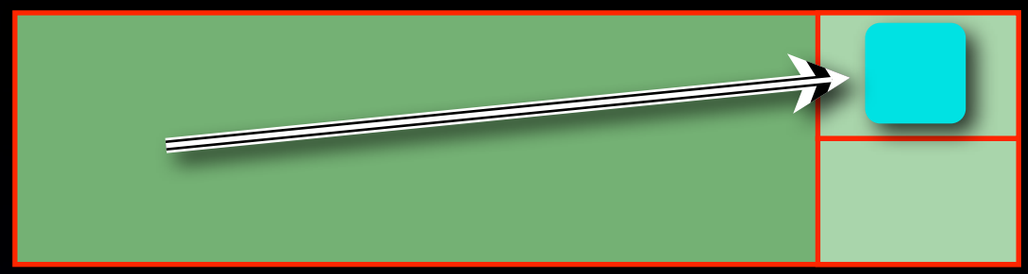
Instance creation

Eden

Instance creation

Eden

Scavenge 1



Eden
SA+SB

Scavenge 2

Eden
SA+SB

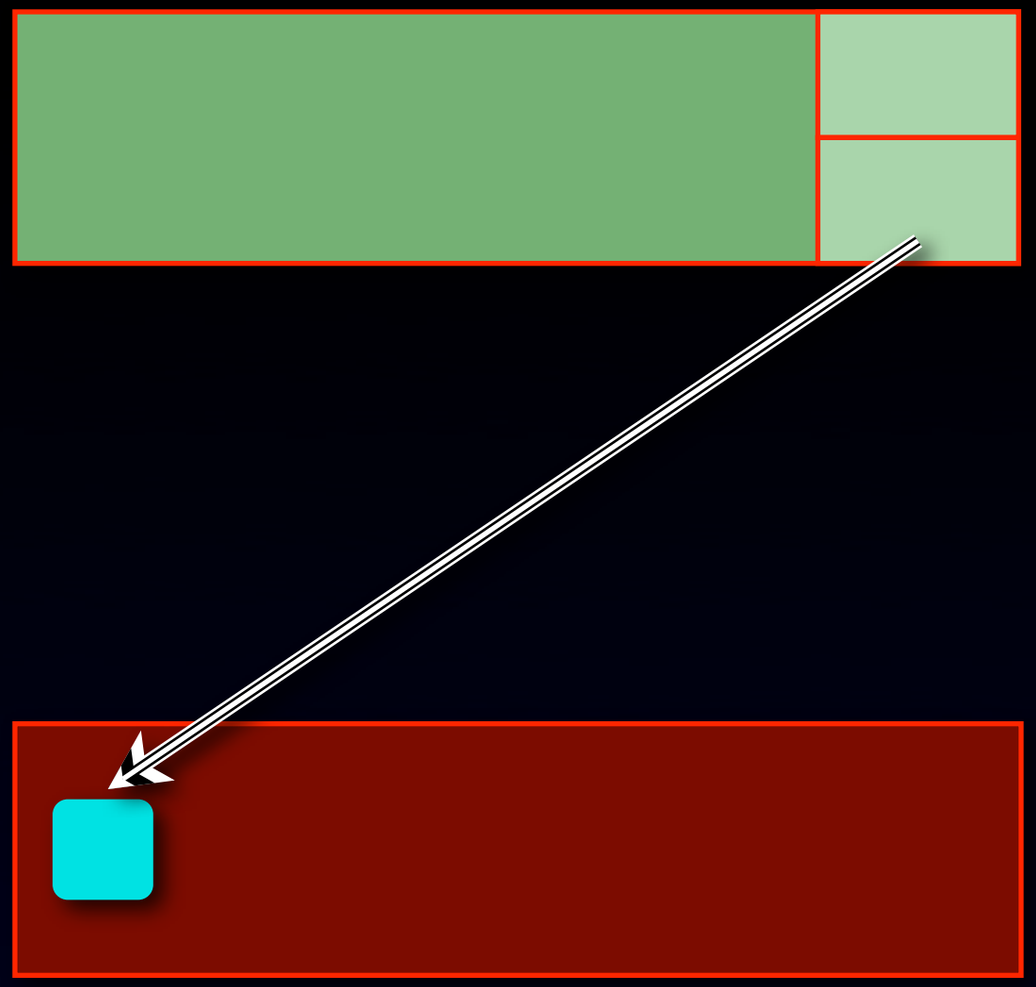
Scavenge 3

Eden
SA+SB

Scavenge 4

Eden
SA+SB

Tenured



Eden
SA+SB

Old

Eden
SA+SB

IGC / GC

Old

Eden
SA+SB

Perm save

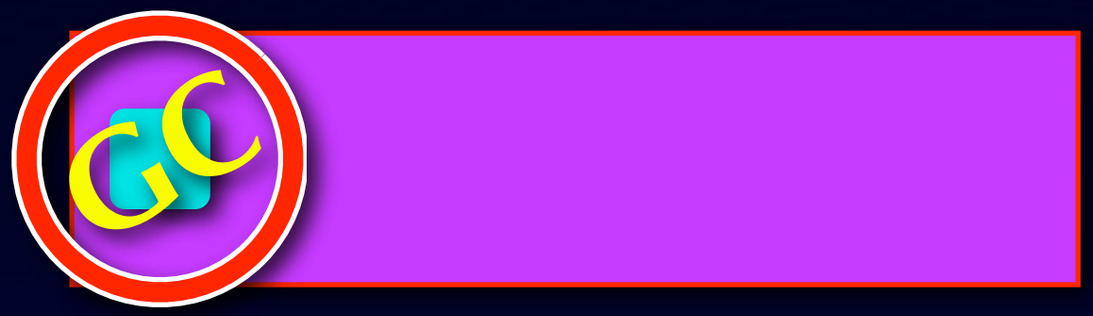
Old

Perm

Eden
SA+SB

Global GC

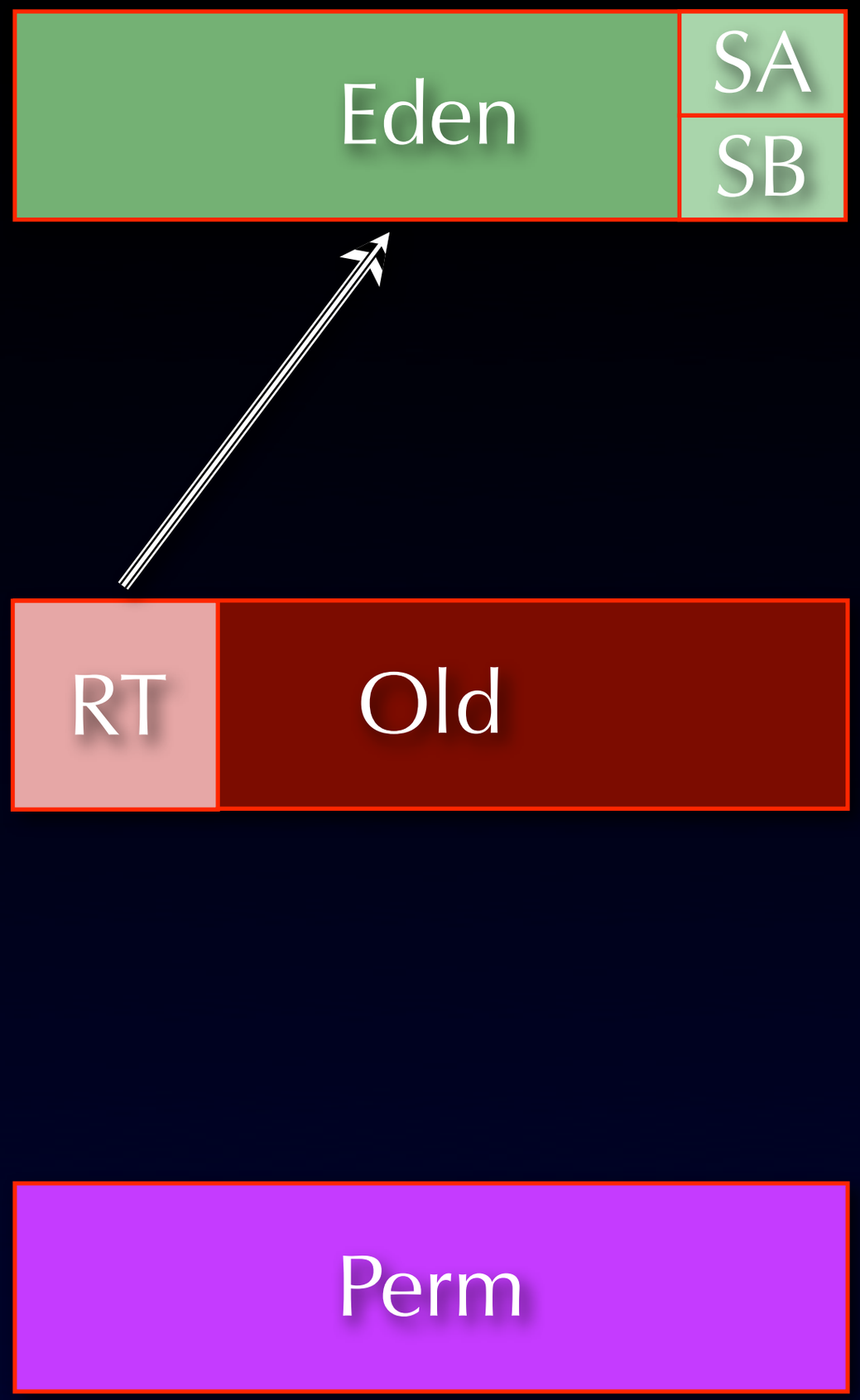
Old



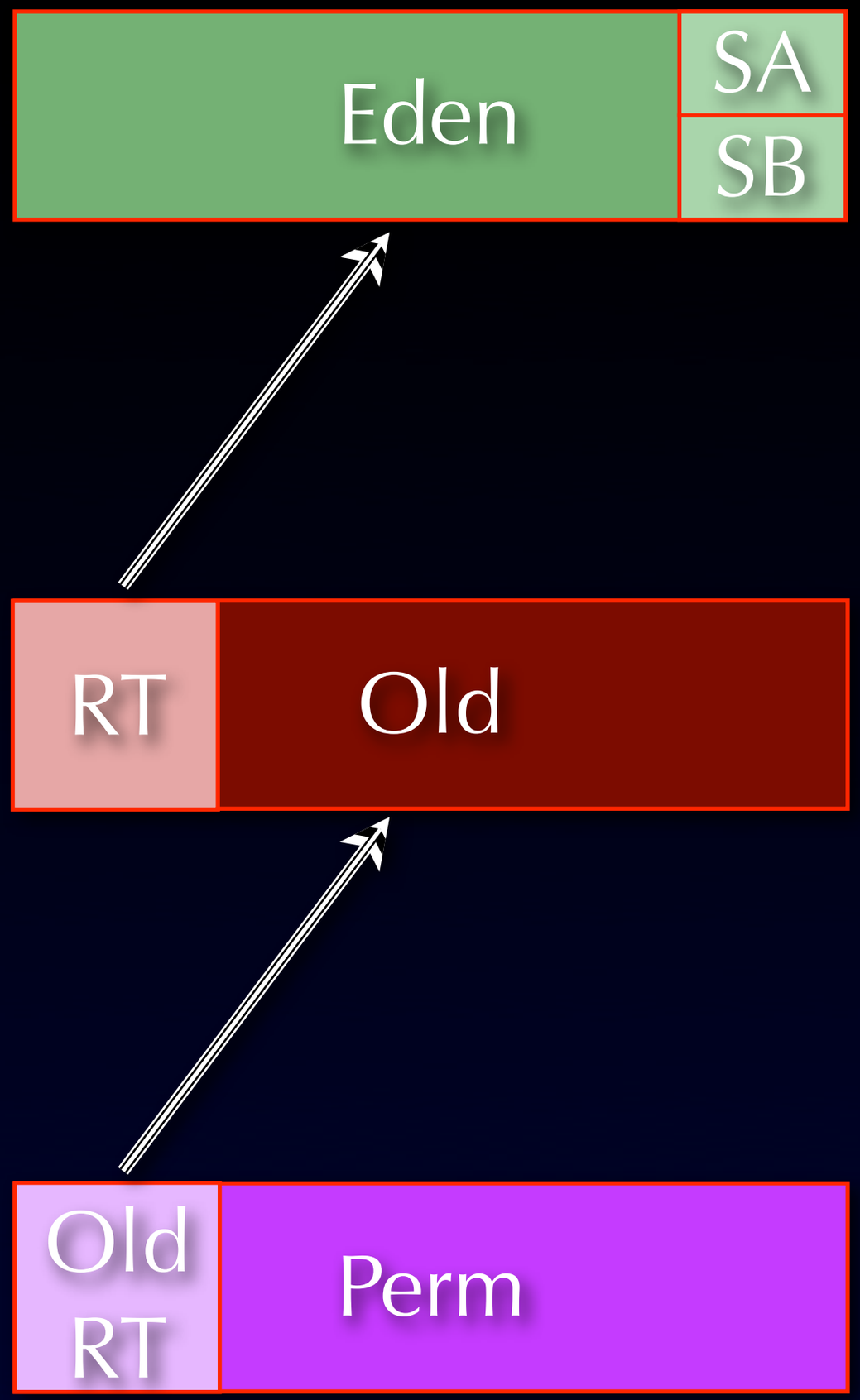
Perm

Fast scavenge?

Scavenge finds live objects faster



Scavenge finds live objects faster



IGC / GC find live objects faster

Responsibilities


Eden
SA
SB


RT
Old


Old
RT
Perm

VM scavenges new space

VM scavenges new space

VI must ensure success

VM scavenges new space

VI must ensure success

VM signals low memory semaphore

VM scavenges new space

VI must ensure success

VM signals low memory semaphore

VI manages old space (GC, grow, shrink)

VM scavenges new space

VI must ensure success

VM signals low memory semaphore

VI manages old space (GC, grow, shrink)

VM (few) fixed actions

VI memory policy classes

Memory policies

Worst case scavenge

Worst case scavenge

Eden and SA or SB full

Worst case scavenge

Eden and SA or SB full

RT grows

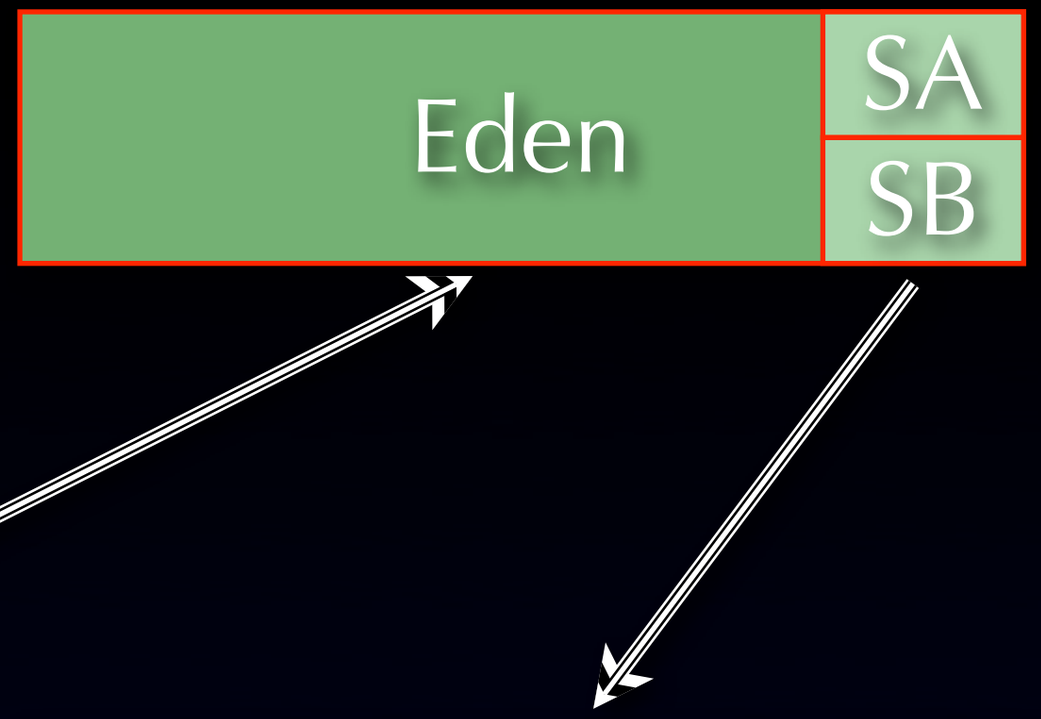
Worst case scavenge

Eden and SA or SB full

RT grows

Old RT grows

Worst case scavenge



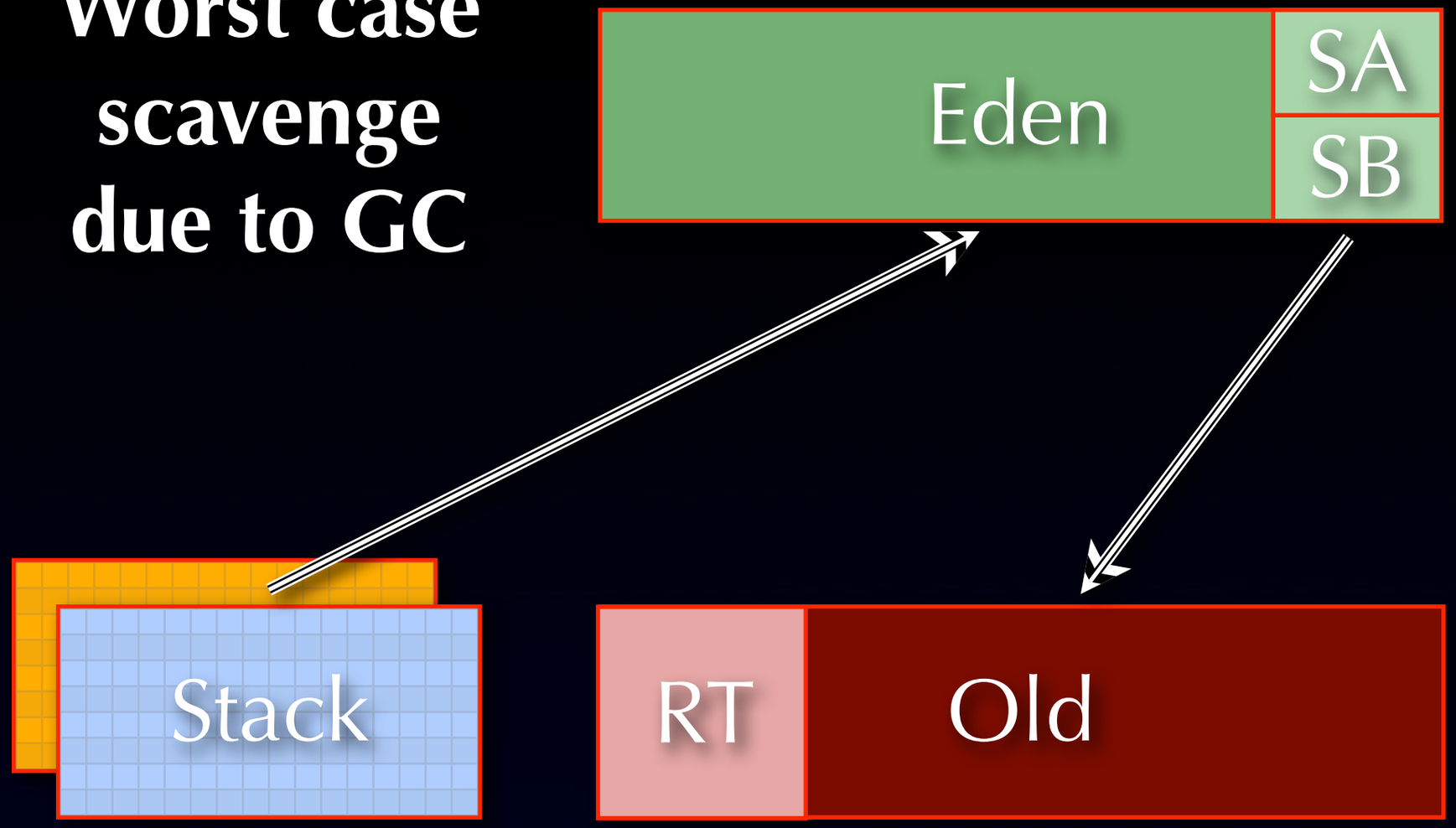
Eden and SA or SB full

RT grows

Large space scavenged

Old RT grows

Worst case scavenge due to GC



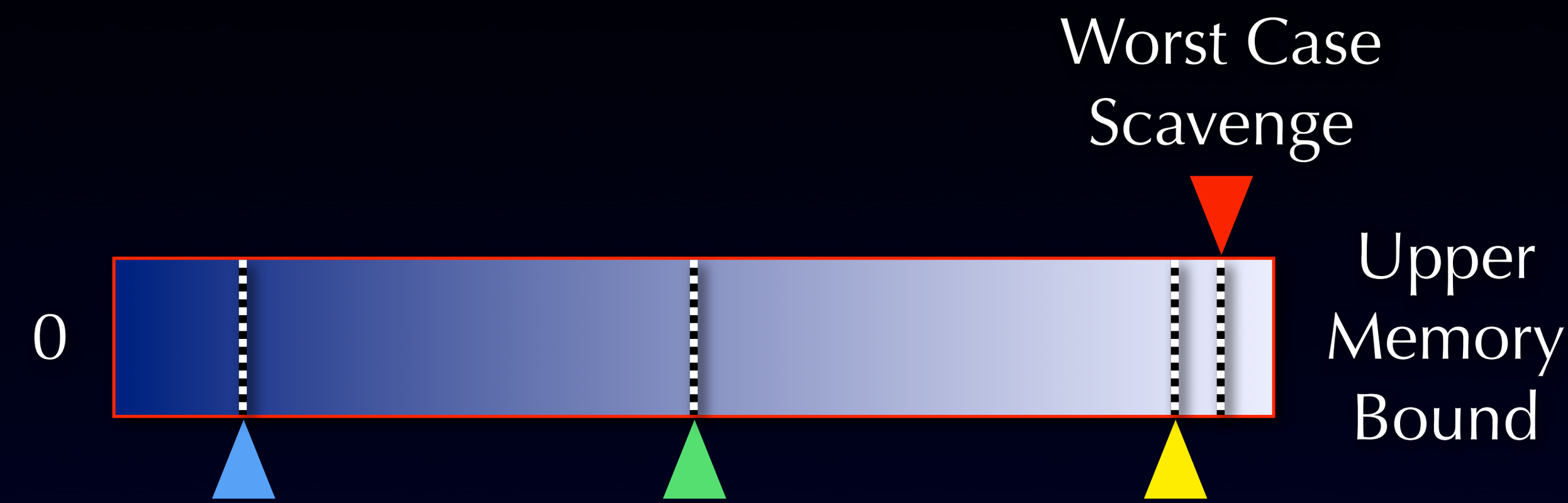
Eden and SA or SB full

RT grows

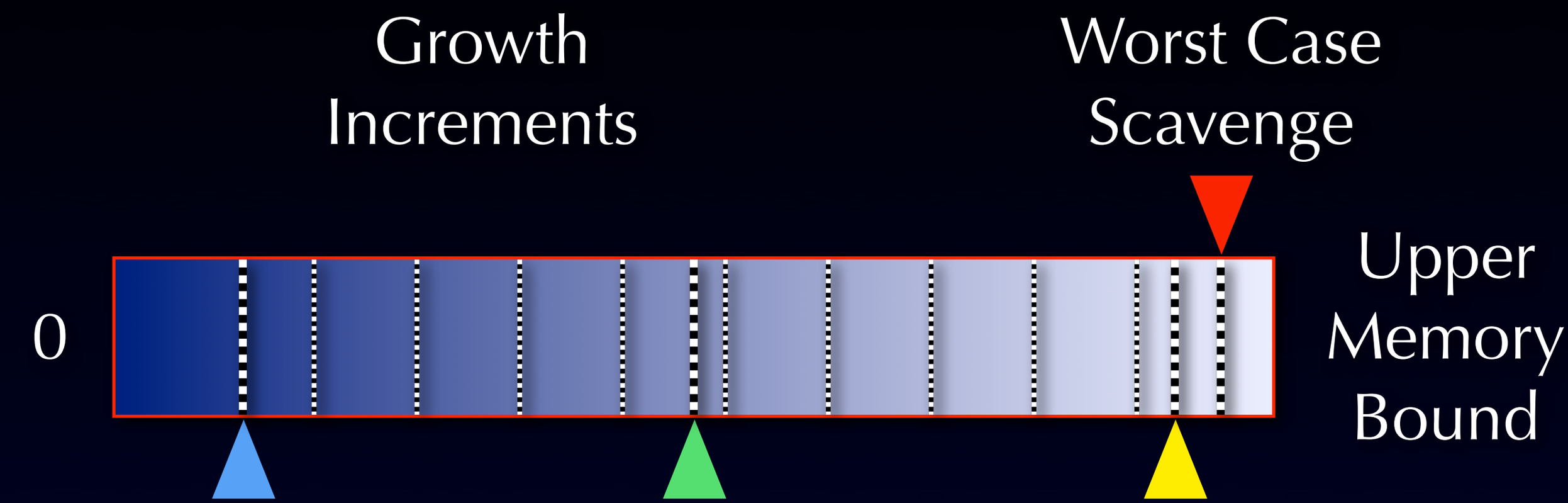
Full stack zone flushed

Old RT grows

The picture


Worst Case
Scavenge
Upper
Memory
Bound
0
Image
Size
Growth
Regime
Bound
Emergency
Threshold

The picture


Growth
Increments
Worst Case
Scavenge
Upper
Memory
Bound
0
Image
Size
Growth
Regime
Bound
Emergency
Threshold

Improvements since Cincom® VisualWorks® 7.7

Fixed since VW 7.7

```
ObjectMemory>>igcState
  ...
  aborting ifTrue: [^#aborting].
  ...
```

```
MemoryPolicy>>dealWithIGC
  ...
  memoryStatus igcState = #aborted
    ifTrue: [“recover from IGC abort”].
  ...
```

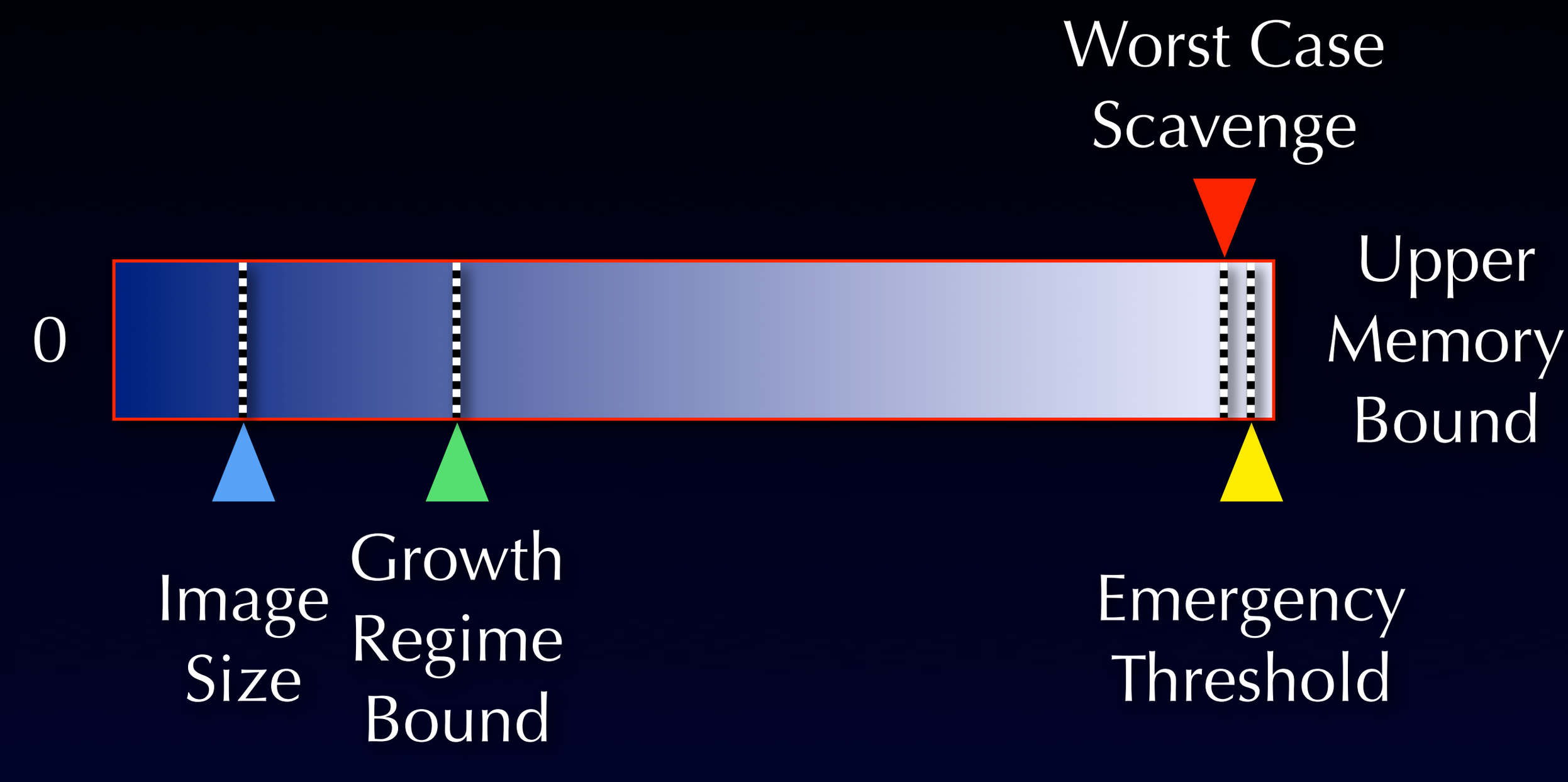
Fixed since VW 7.7

```
ObjectMemory>>igcState
  ...
  aborting ifTrue: [^#aborting].
  ...
```

IGC does not recover from stack overflow

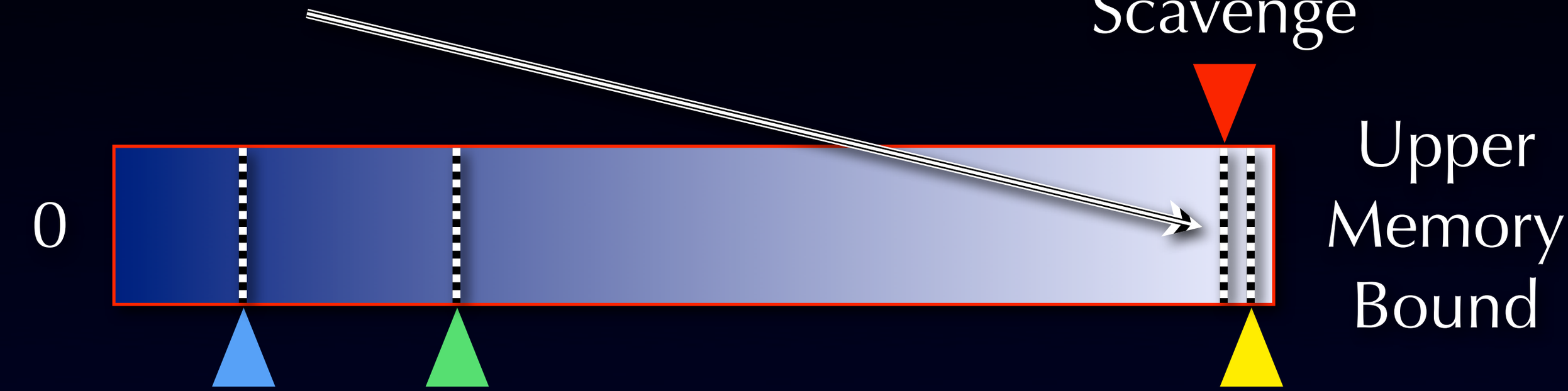
```
MemoryPolicy>>dealWithIGC
  ...
  memoryStatus igcState = #aborted
    ifTrue: [“recover from IGC abort”].
  ...
```

Fixed since VW 7.7



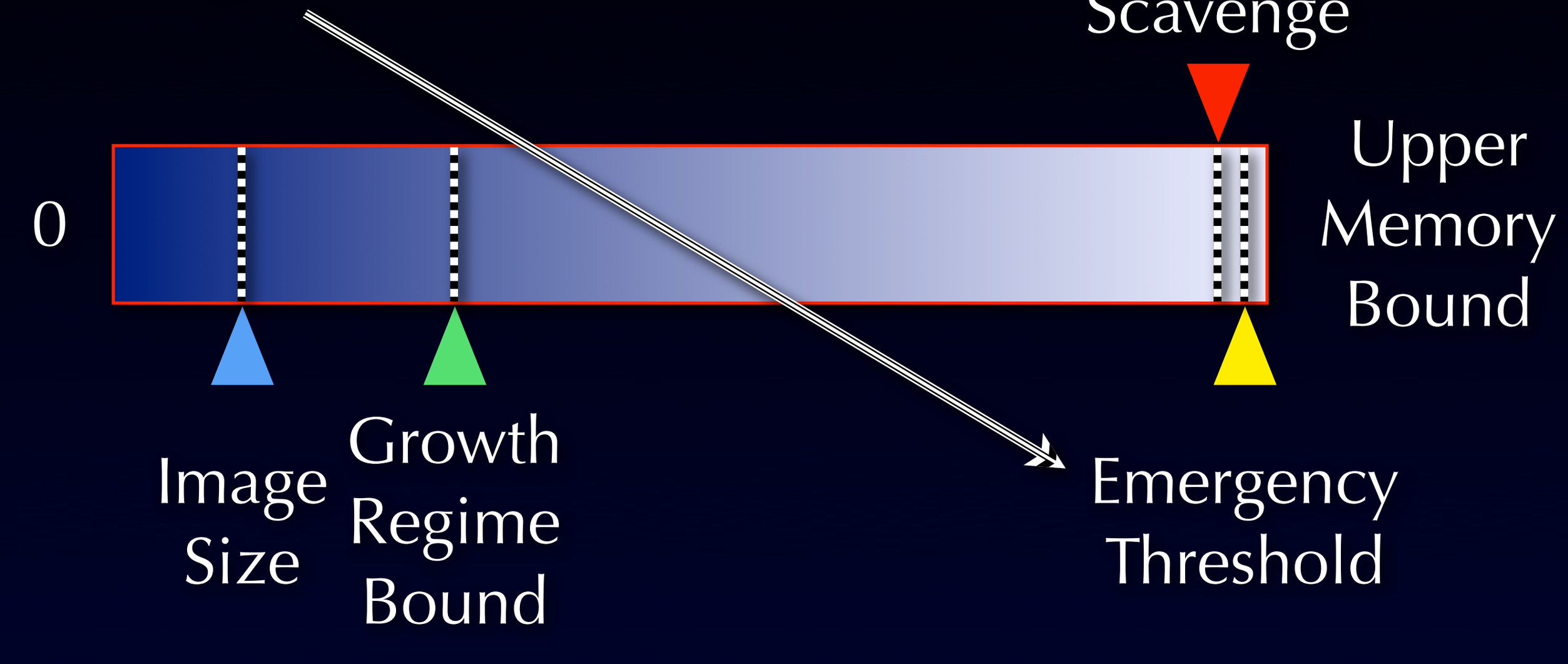
Fixed since VW 7.7

VM crash due to scavenge failure



Fixed since VW 7.7

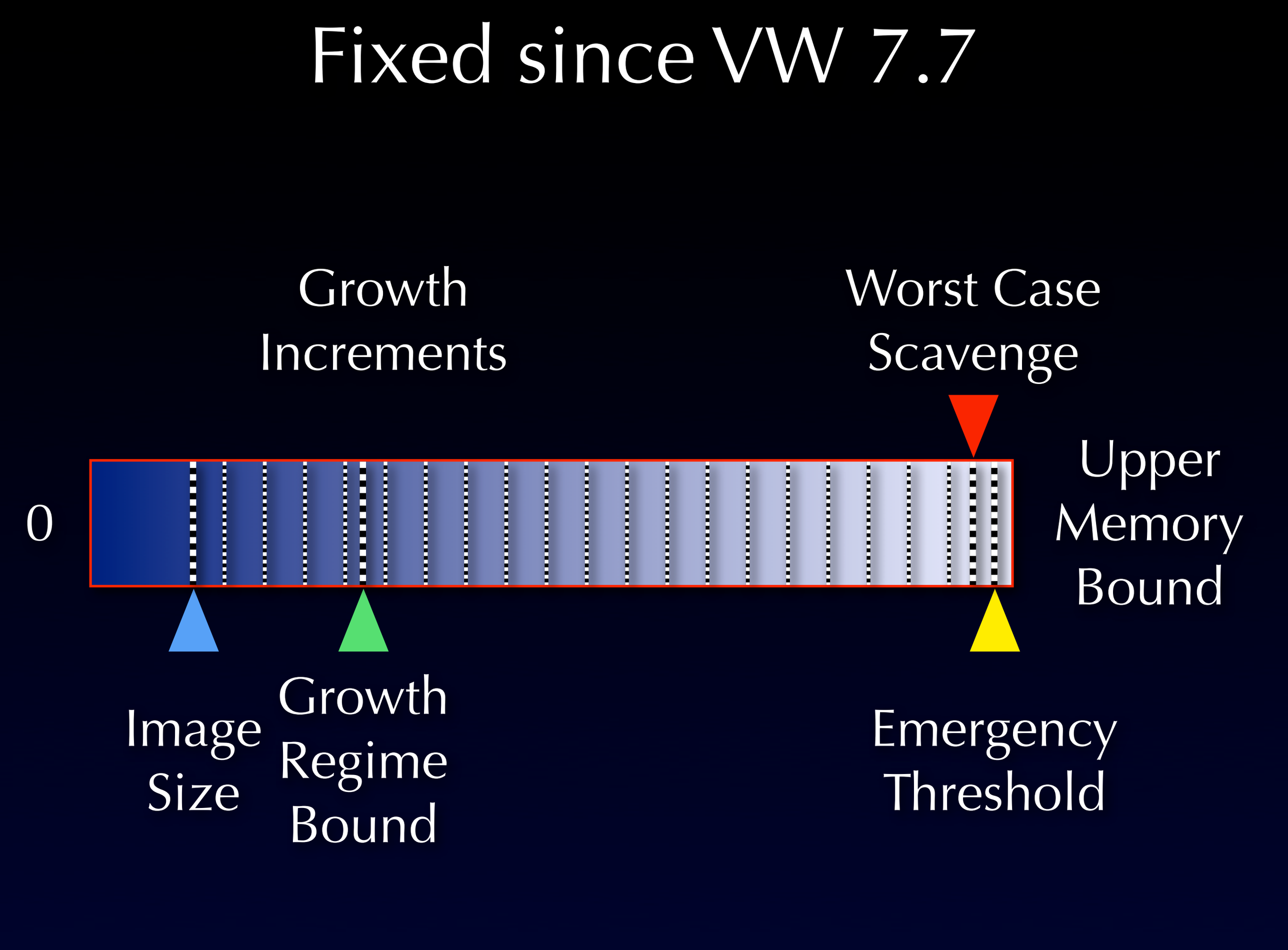
IGC as last resort, bogus emergencies



Fixed since VW 7.7

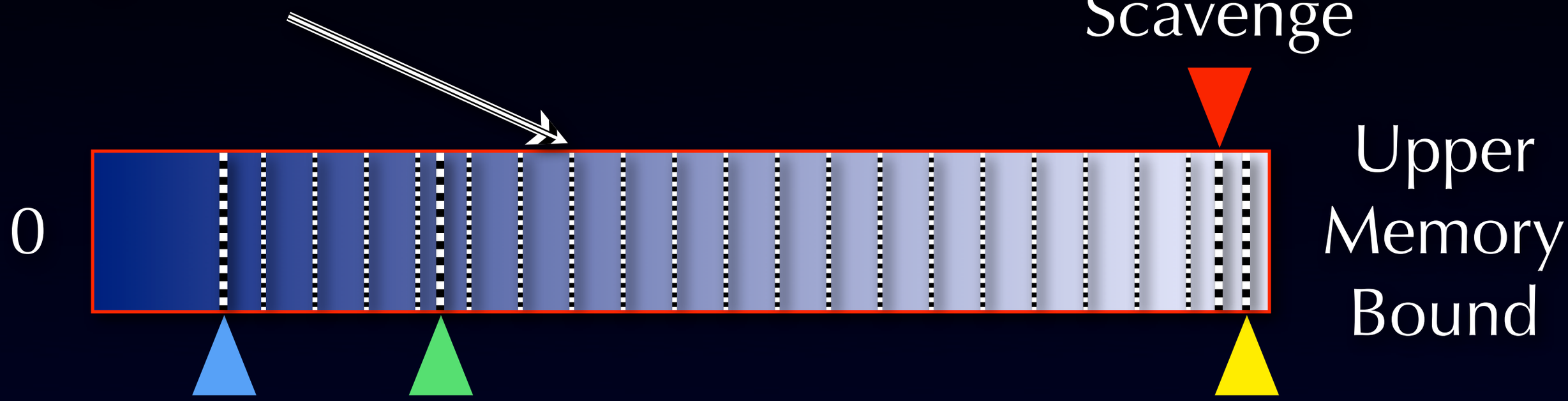
Too low for modern applications

Fixed since VW 7.7



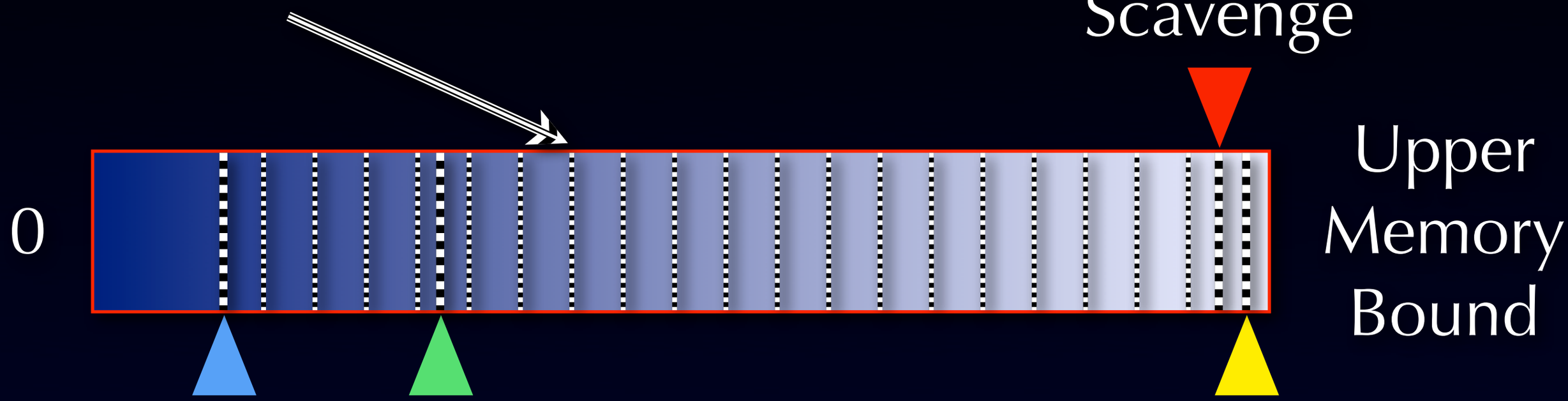
Fixed since VW 7.7

Tiny growth increment, fragmentation



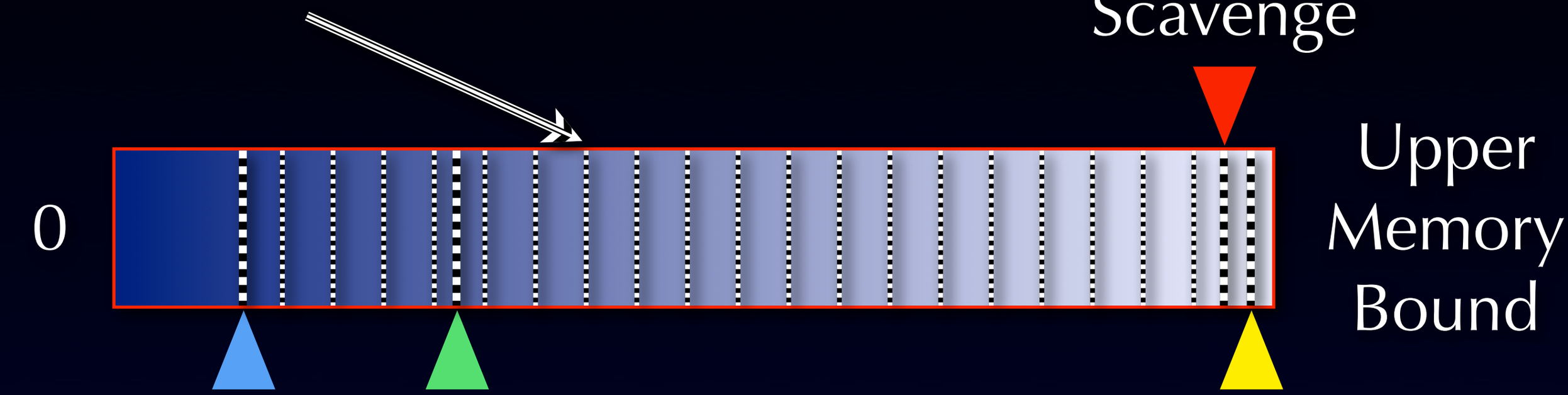
Fixed since VW 7.7

Tiny growth increment, excessive GC



Fixed since VW 7.7

Low free memory limit, even more GC



New technology

TESTS

N TESTS

- MemoryPolicyChecker

N TESTS sy

- MemoryPolicyChecker
- MemoryPolicyTuner

N TESTS ty

- MemoryPolicyChecker
- MemoryPolicyTuner
- MemoryPolicyStressTest

Memory Monitor

Space		Bytes used	Segs
Eden	10%	152,748	1
Survivor	14%	43,008	2
Old	3%	924,772	2

Space		Bytes used	Segs
Large	42%	425,340	1
Fixed	0%	20	1
Perm	100%	34,765,856	1

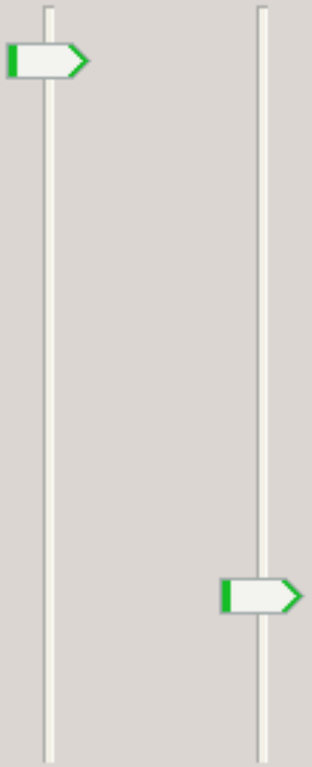
Scavenges: 484 last: 0 peak: 1

Used memory: 36,311,744 min: 34,765,856 peak: 37,662,916

GCs:	0	GCCs:	0
IGC state:	resting	IGCs:	10
Marked bytes:	735,192	Marked objects:	21,203
Marked weak bytes:	58,664	Marked weak objects:	159
Nilled bytes:	57,948	Nilled objects:	159
Reclaimed bytes:	717,935	Reclaimed objects:	37,291
Swept objects:	58,498	Unmarked objects:	0
Sweep alloc. bytes:	106,212	Sweep alloc. objects:	3,112
Stack spills:	342	JIT cache spills:	8
IGC stack overflows:	0	IGC mark aborts:	0
Mark stack overflows:	0		
Weak list overflows:	0		

IGC GC GGC FNS MAP Log

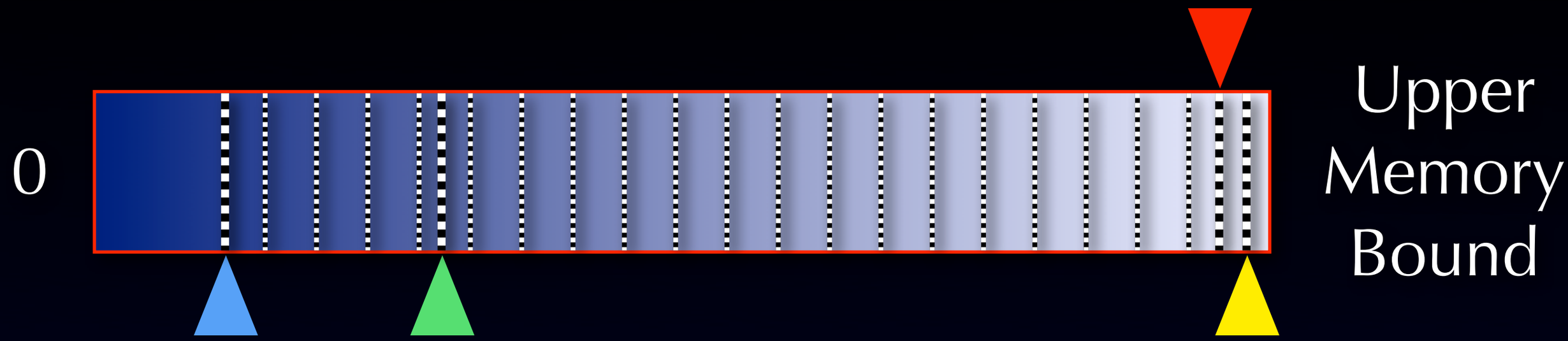
Delay: Priority:

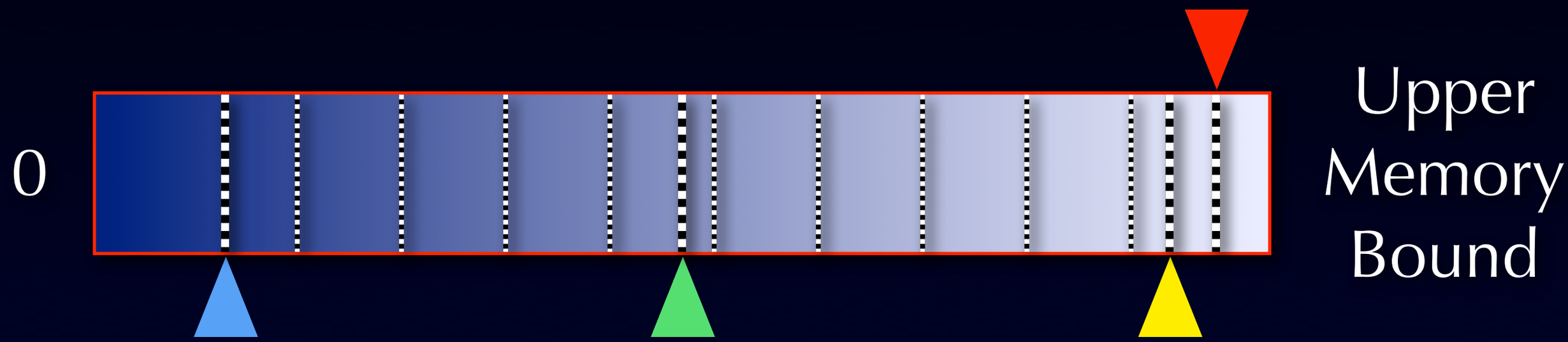


0.75 71

Before and after

Before





After

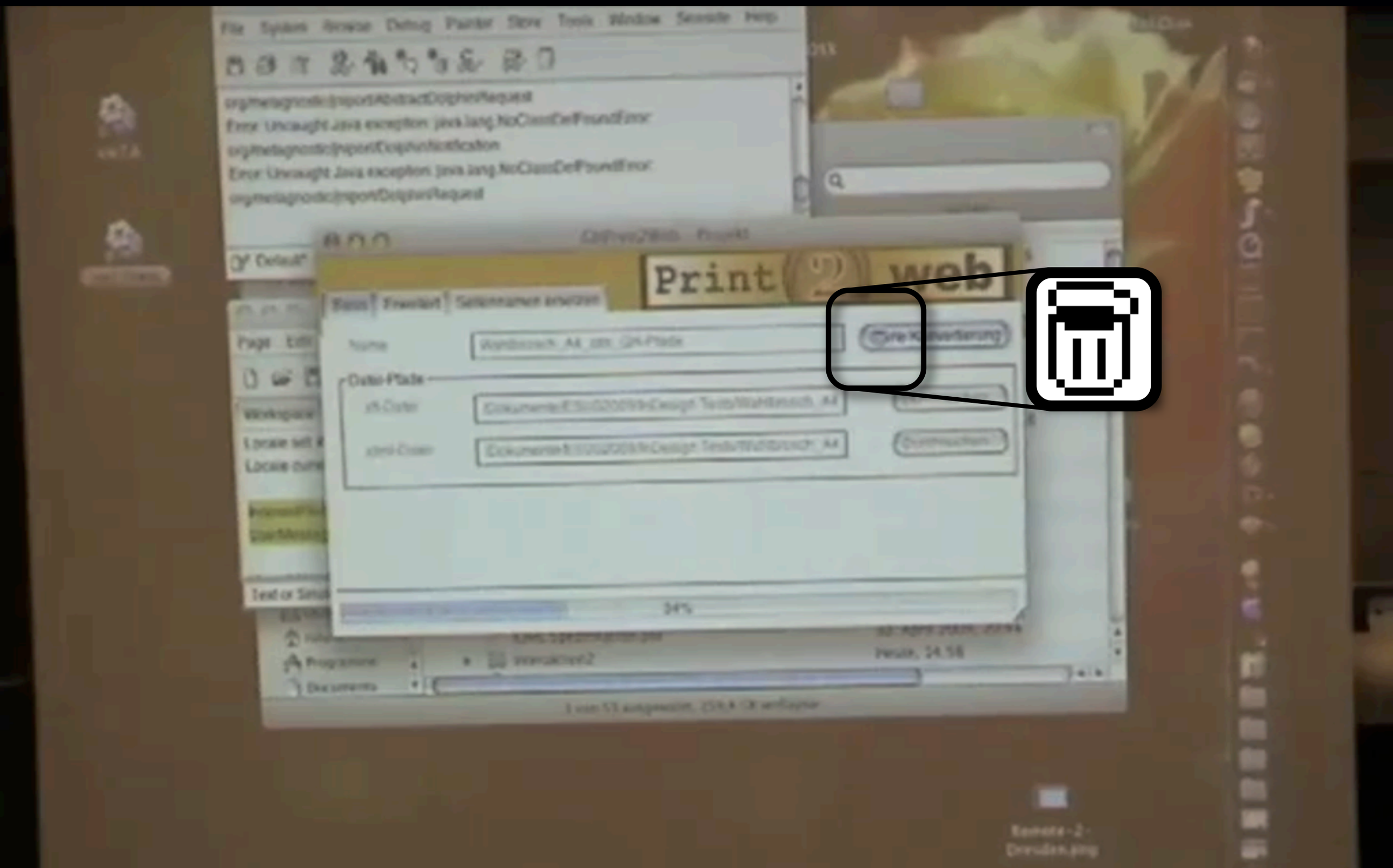
VW 7.7 legacy MemoryPolicy (with fixes)	~1 day, if they work
VW 7.7.1 without sizesAtStartup tuning	~160 minutes
VW 7.7.1 default	~73 minutes

A: pointer garbage
B: byte garbage
C: point creation

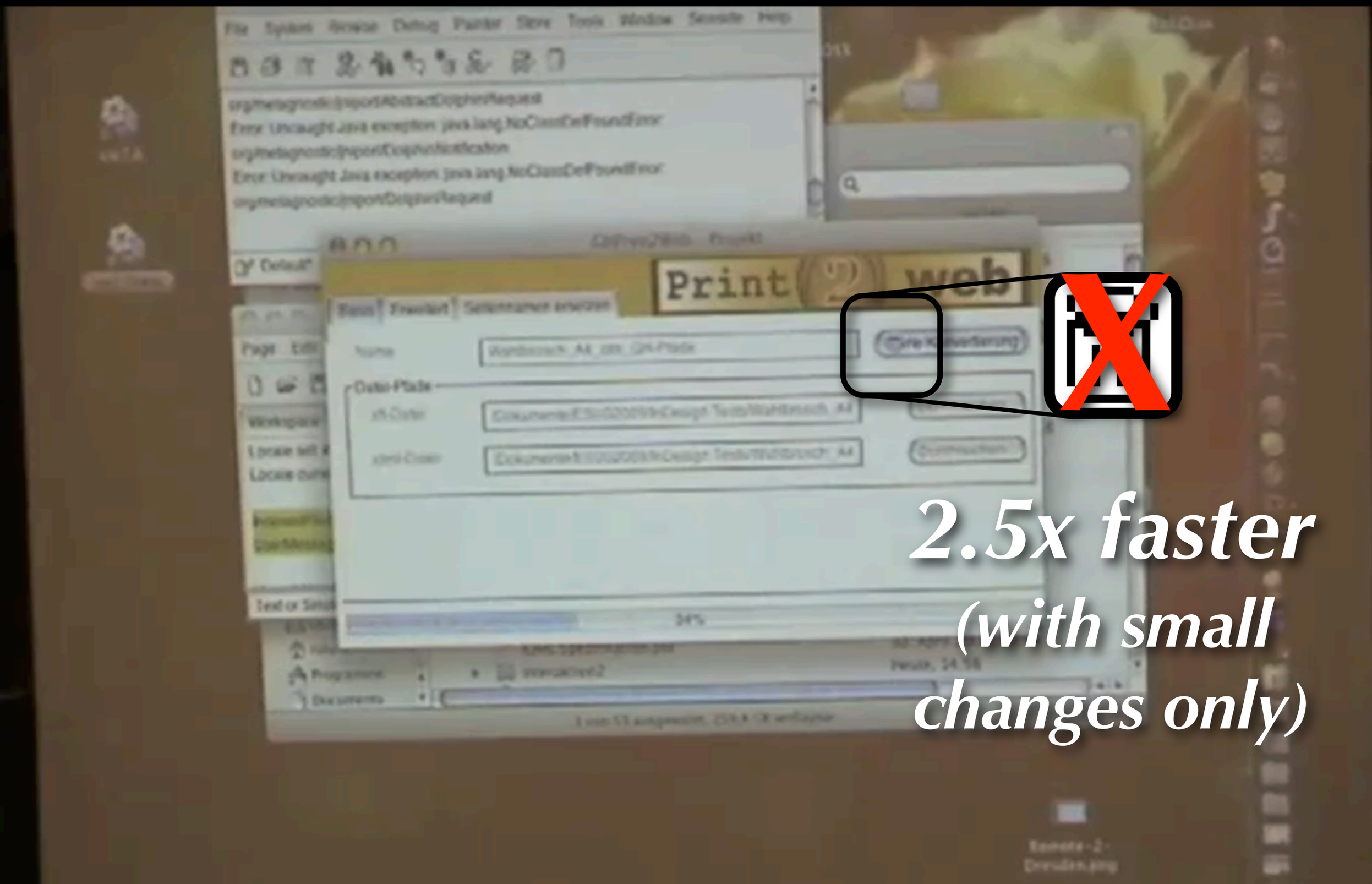
Segmented container
D: byte allocation
E: pointer allocation

Large container
F: byte allocation
G: pointer allocation

Print 2 web, ESUG 2009



Print 2 web, ESUG 2009



VisualWorks 7.8 and beyond

VisualWorks 7.8+

- Fixed space allocation fixed
- Improved weak / ephemeron support
- IGC performance improvements
- GC / IGC mark stack overflow prevention
- GC moves large objects back into large space
- New -m[1..7] VM switches

VisualWorks 7.8+

- Fixed space allocation fi
- Impro
- C
- New -m[1..7] VM switches

DONE

(November 2010)

jects back into large space

VisualWorks 7.9

VisualWorks 7.9

- Up to 40% faster GC

VisualWorks 7.9

- Up to 40% faster GC
- Improved IGC uses less memory

VisualWorks 7.9

- Up to 40% faster GC
- Improved IGC uses less memory
- Adaptive time based memory policy IGC driver

VisualWorks 7.9

- Up to 40% faster GC
- Improved IGC uses less memory
- Adaptive time based memory policy IGC driver
- Memory policy IGC smart abort

VisualWorks 7.9

- Up to 40% faster GC
- Improved IGC uses less memory
- Adaptive time based memory policy IGC driver
- Memory policy IGC smart abort
- Memory policy IGC switch

VisualWorks 7.9

- Up to 40% faster GC
- Improved IGC uses less memory
- Adaptive time based memory policy IGC driver
- Memory policy IGC smart abort
- Memory policy IGC switch
- Updates to the OldRT, become:, fixed space...

VisualWorks 7.9

- Up to 40% faster GC
- Improved I/O
- [illegible]er
- [illegible]
- M[illegible]itch
- Updates to the OldRT, become:, fixed space...

DONE

(August 2011)

VisualWorks 7.9 bis

VisualWorks 7.9 bis

- Rewritten, more efficient scavenger

VisualWorks 7.9 bis

- Rewritten, more efficient scavenger
  - RT optimizations

VisualWorks 7.9 bis

- Rewritten, more efficient scavenger
  - RT optimizations
  - New 64 bit class table management

VisualWorks 7.9 bis

- Rewritten, more efficient scavenger
  - RT optimizations
  - New 64 bit class table management
  - ~1000 LOC deleted

VisualWorks 7.9 bis

- Rewritten, more efficient scavenger
  - RT optimizations
  - New 64 bit class table management
  - ~1000 LOC deleted
- 2000+ new tests

VisualWorks 7.9 bis

- Rewritten, more efficient scavenger
  - RT optimizations
  - New 64 bit class table management
  - ~1000 LOC deleted
- 2000+ new tests
- Fixes to fixed space, perm space ephemerons, memcpy(), memory policy, etc...

VisualWorks 7.9 bis

- Rewritten, more efficient sca
  - RT optim
  -
- 2(
- Fix s to fixed space, perm space ephemerons, memcpy(), memory policy, etc...

DONE

(January 2012)

Questions?

Star Team (Smalltalk Strategic Resources)

sfortman@cincom.com Smalltalk Director

athomas@cincom.com Smalltalk Product Manager

jjordan@cincom.com Smalltalk Marketing Manager

http://www.cincomsmalltalk.com