ORACLE

Java™

# ZGC

## The Next Generation Low-Latency Garbage Collector

**Per Liden (@perliden)**
Consulting Member of Technical Staff
Java Platform Group, Oracle

# Agenda

1. What is ZGC?
2. The Design of ZGC
3. Performance
4. Using ZGC
5. Roadmap

# Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

# A Scalable Low-Latency Garbage Collector

# Goals

Max GC pause time

**10ms**

Multi-terabyte heaps

**TB**

Max throughput reduction

**15%**

# Goals

Easy to tune!

# GC Landscape

Oracle supported garbage collectors

| GC | Optimized For |
|---|---|
| Serial | Memory Footprint |
| Parallel | Throughput |
| G1 | Throughput/Latency Balance |
| ZGC | Low Latency |

# ZGC at a Glance

Concurrent
Tracing
Compacting
Single generation

Region-based
NUMA-aware
Load barriers
Colored pointers

ZGC pause times **do not** increase with the heap or live-set size

# ZGC pause times **do** increase with the root-set size

(Number of Java Threads)

# What's Concurrent?

| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | | | | | |
| Compaction | | | | | |
| Reference Processing | | | | | |
| Relocation Set Selection | | | | | |
| StringTable Cleaning | | | | | |
| JNI WeakRef Cleaning | | | | | |
| JNI GlobalRefs Scanning | | | | | |
| Class Unloading | | | | | |
| Thread Stack Scanning | | | | | |

# What's Concurrent?

| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | - | - | | | |
| Compaction | - | - | | | |
| Reference Processing | - | - | | | |
| Relocation Set Selection | - | - | | | |
| StringTable Cleaning | - | - | | | |
| JNI WeakRef Cleaning | - | - | | | |
| JNI GlobalRefs Scanning | - | - | | | |
| Class Unloading | - | - | | | |
| Thread Stack Scanning | - | - | | | |

# What's Concurrent?

| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | - | - | ✓ | ✓ | |
| Compaction | - | - | - | - | |
| Reference Processing | - | - | - | - | |
| Relocation Set Selection | - | - | - | - | |
| StringTable Cleaning | - | - | - | - | |
| JNI WeakRef Cleaning | - | - | - | - | |
| JNI GlobalRefs Scanning | - | - | - | - | |
| Class Unloading | - | - | - | - | |
| Thread Stack Scanning | - | - | - | - | |

*) Old Gen only

# What's Concurrent?

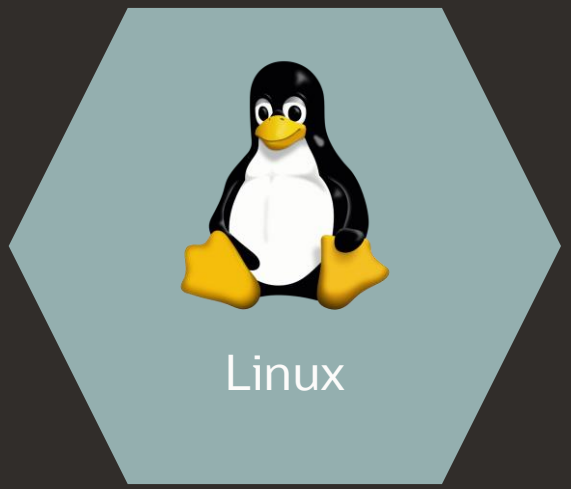
| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | - | - | ✓ * | ✓ * | |
| Compaction | - | - | - | - | |
| Reference Processing | - | - | - | - | |
| Relocation Set Selection | - | - | - | - | |
| StringTable Cleaning | - | - | - | - | |
| JNI WeakRef Cleaning | - | - | - | - | |
| JNI GlobalRefs Scanning | - | - | - | - | |
| Class Unloading | - | - | - | - | |
| Thread Stack Scanning | - | - | - | - | |

# What's Concurrent?

*) Old Gen only
**) Not yet

| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | - | - | ✓ * | ✓ * | ✓ |
| Compaction | - | - | - | - | ✓ |
| Reference Processing | - | - | - | - | ✓ |
| Relocation Set Selection | - | - | - | - | ✓ |
| StringTable Cleaning | - | - | - | - | ✓ |
| JNI WeakRef Cleaning | - | - | - | - | ✓ |
| JNI GlobalRefs Scanning | - | - | - | - | ✓ |
| Class Unloading | - | - | - | - | ✓ |
| Thread Stack Scanning | - | - | - | - | - ** |

# Supported on All Commonly Used Platforms




x86 (64-bit)
Arm (64-bit)


x86 (64-bit)


x86 (64-bit)

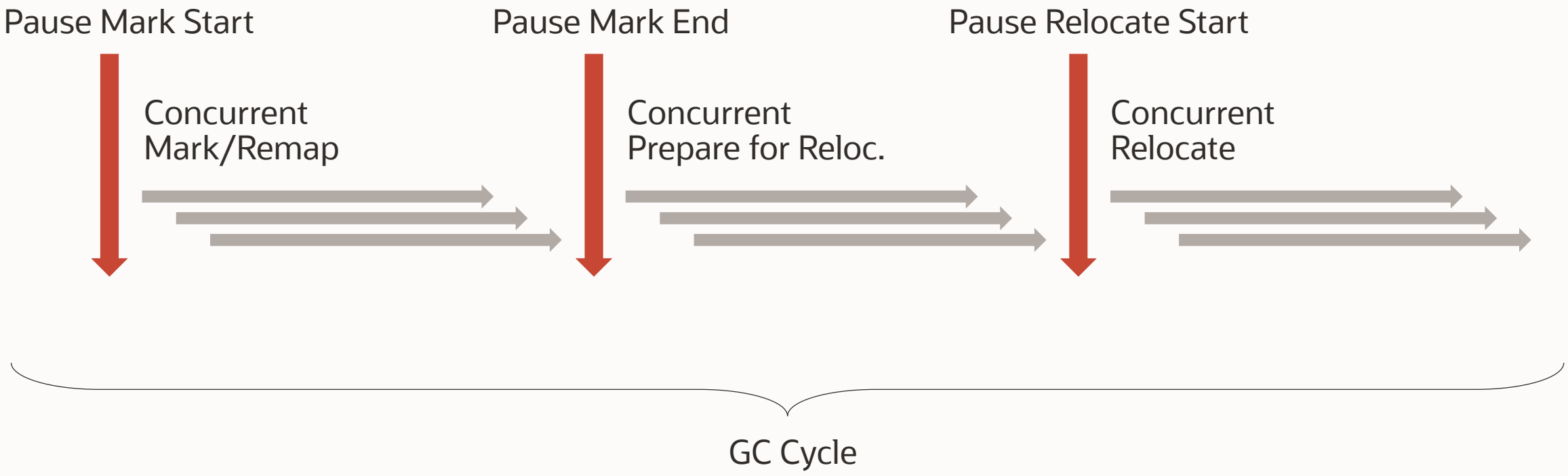
# Production Ready in JDK 15

Production Ready!

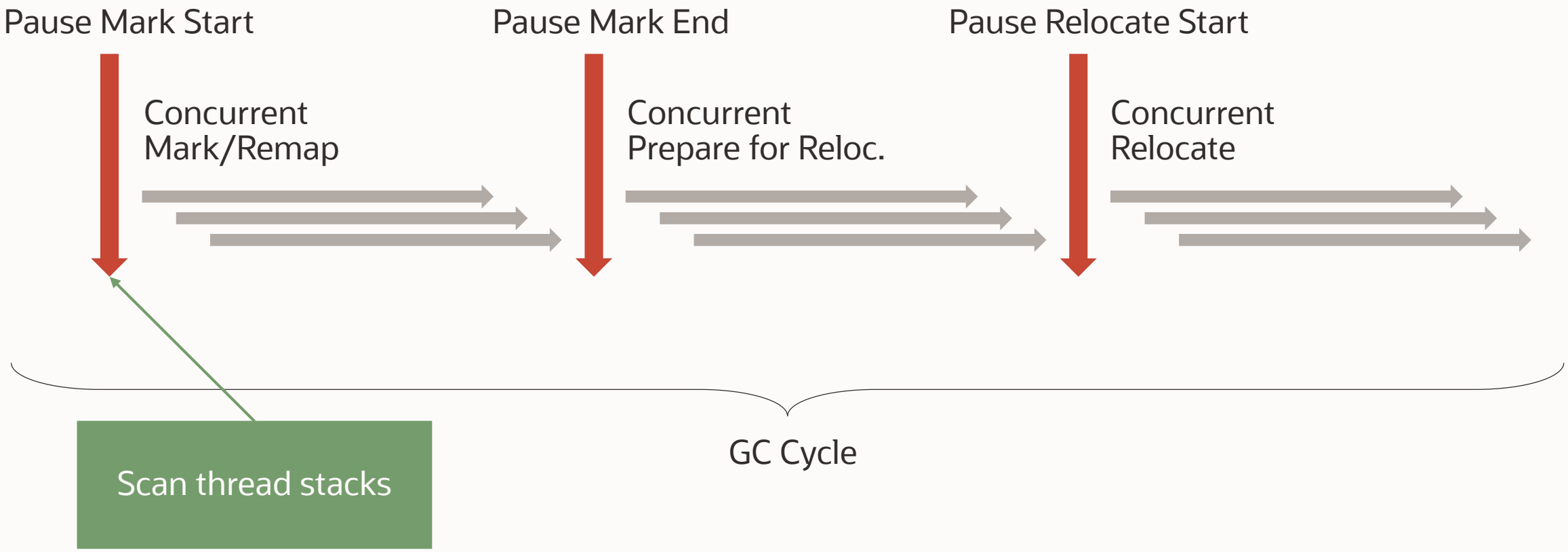
| JDK 11 | JDK 12 | JDK 13 | JDK 14 | JDK 15 |
|---|---|---|---|---|
| • Initial release<br>• Experimental<br>• Linux/x64 | • Concurrent class unloading<br>• Pause time reductions<br>• … | • Linux/aarch64<br>• Uncommit unused memory<br>• Soft max heap size<br>• … | • Windows/x64<br>• macOS/x64<br>• Tiny heaps<br>• JFR leak profiler<br>• Stability improvements<br>• … | • Compressed class pointers<br>• Class Data Sharing<br>• NUMA improvements<br>• Incremental uncommit<br>• … |

# The Design of ZGC

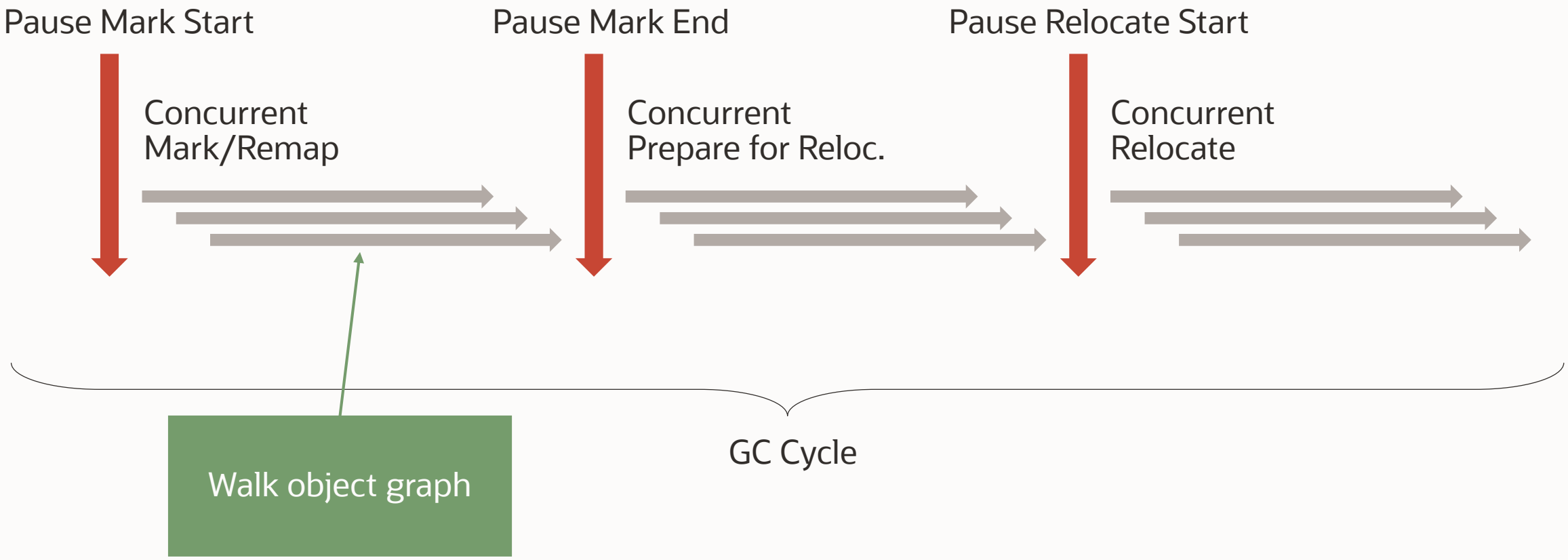
# ZGC Phases



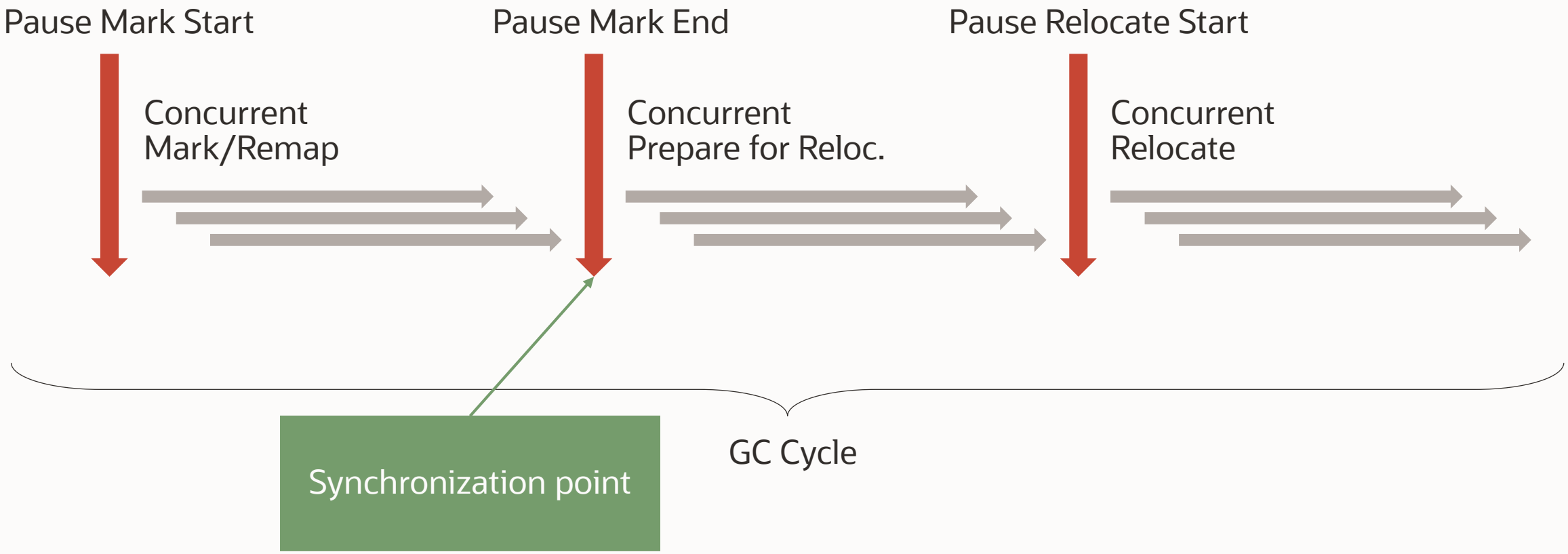
# ZGC Phases



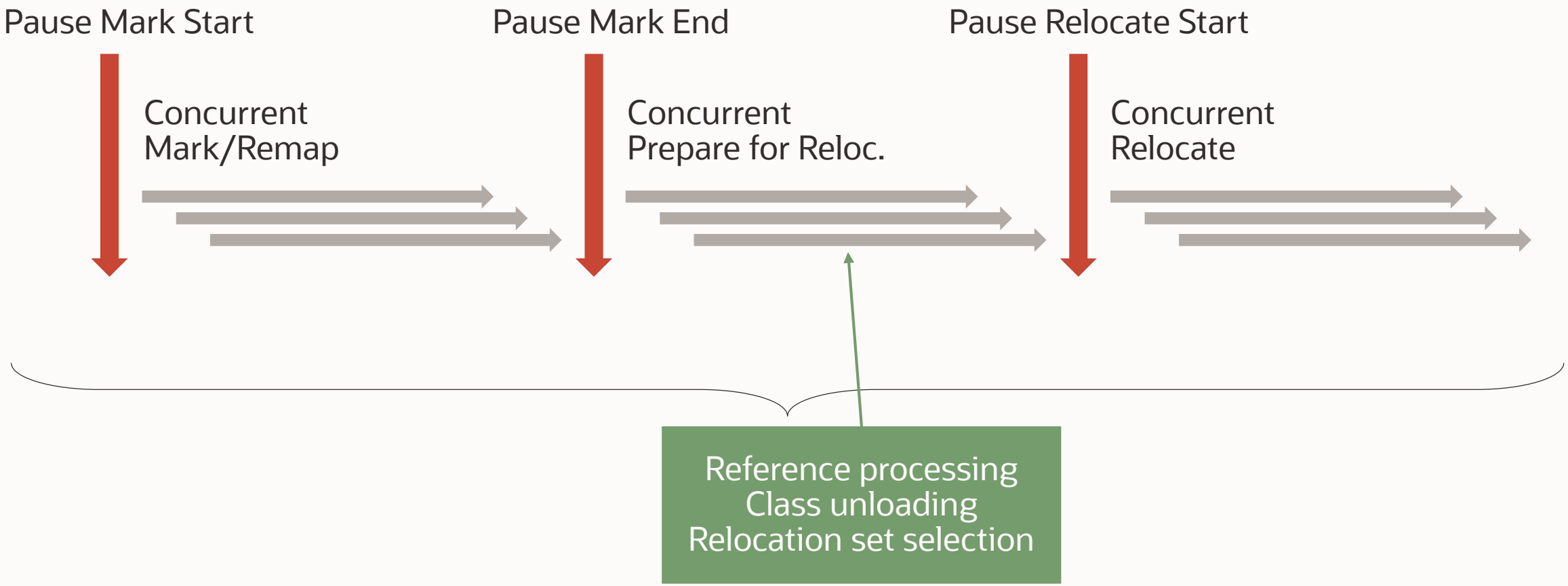
# ZGC Phases



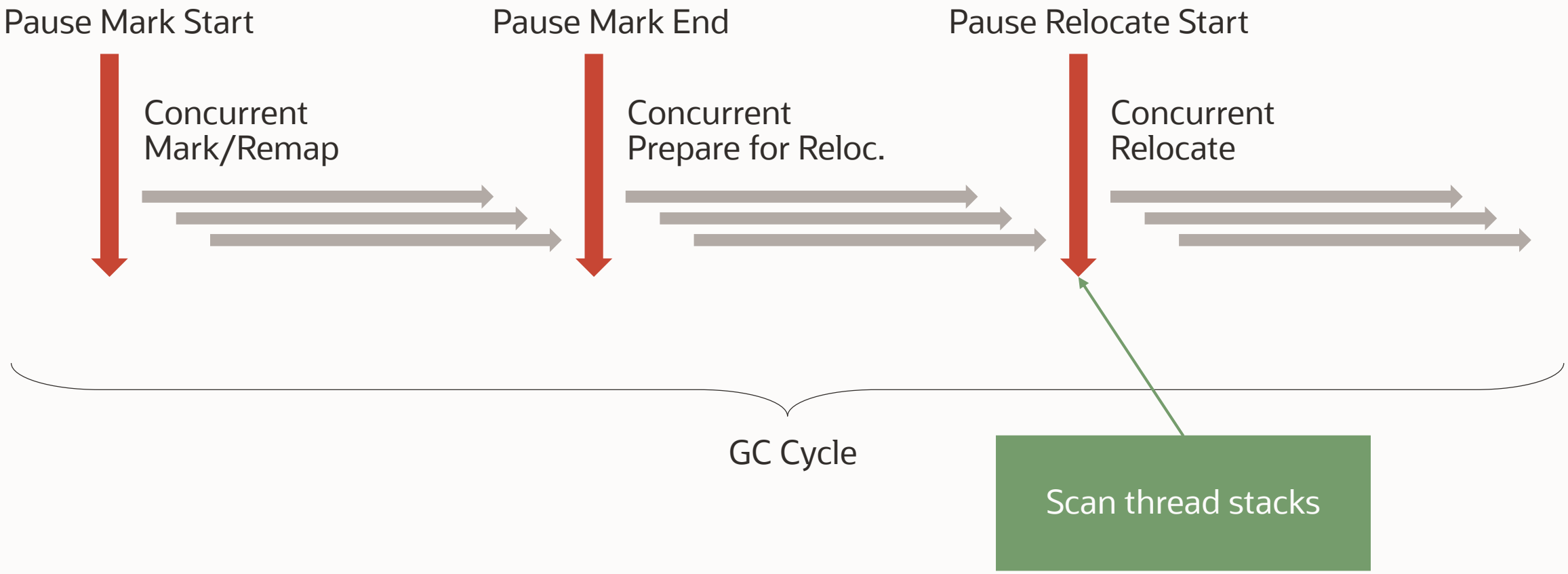
# ZGC Phases



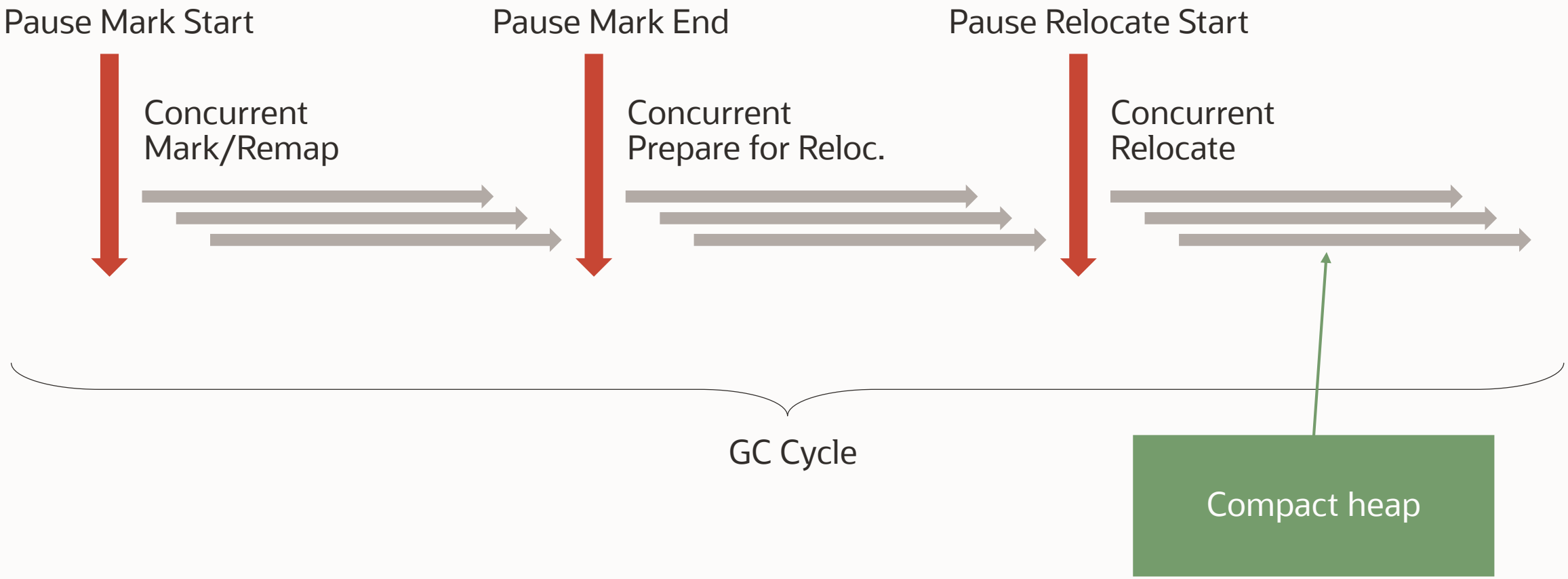
# ZGC Phases



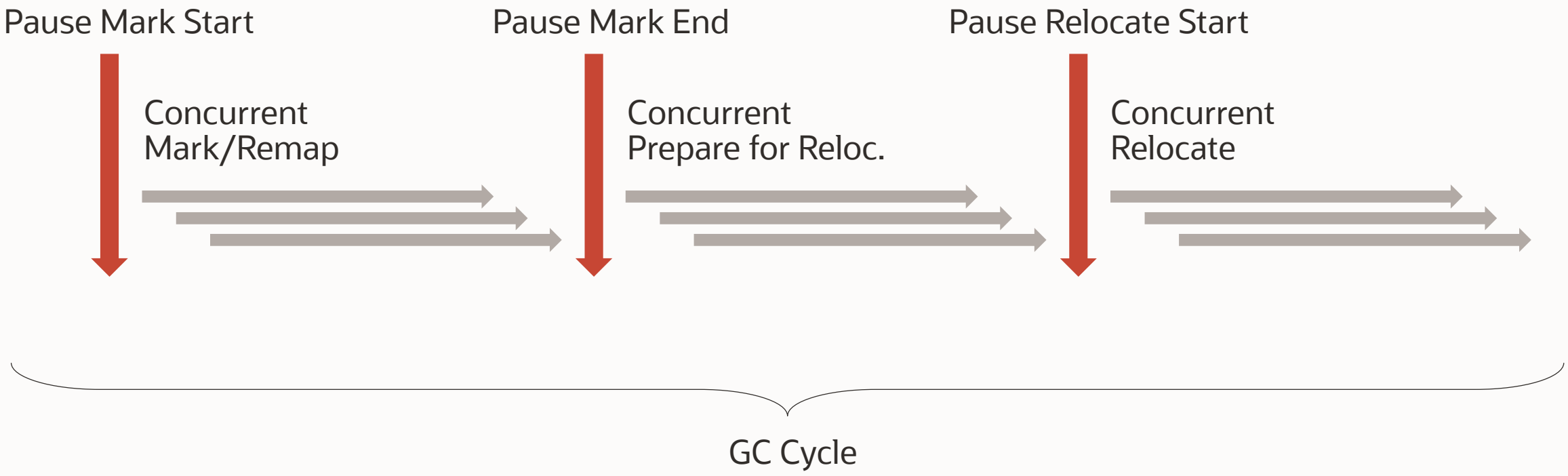
# ZGC Phases



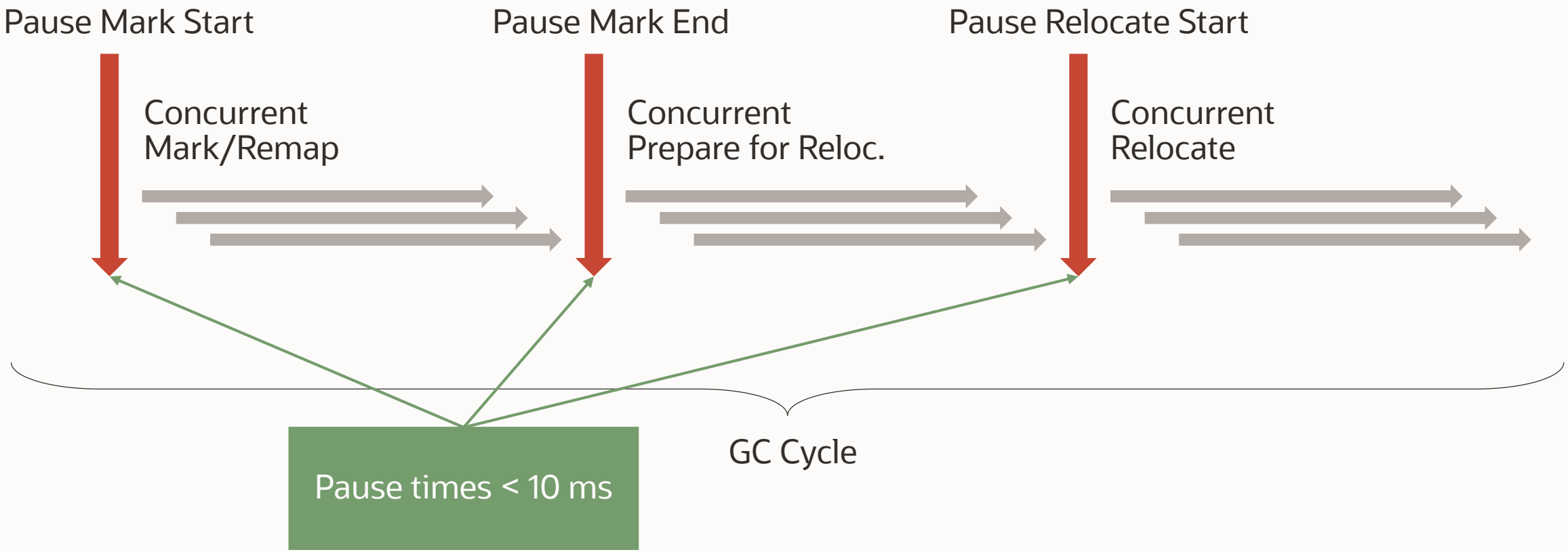
# ZGC Phases



# ZGC Phases

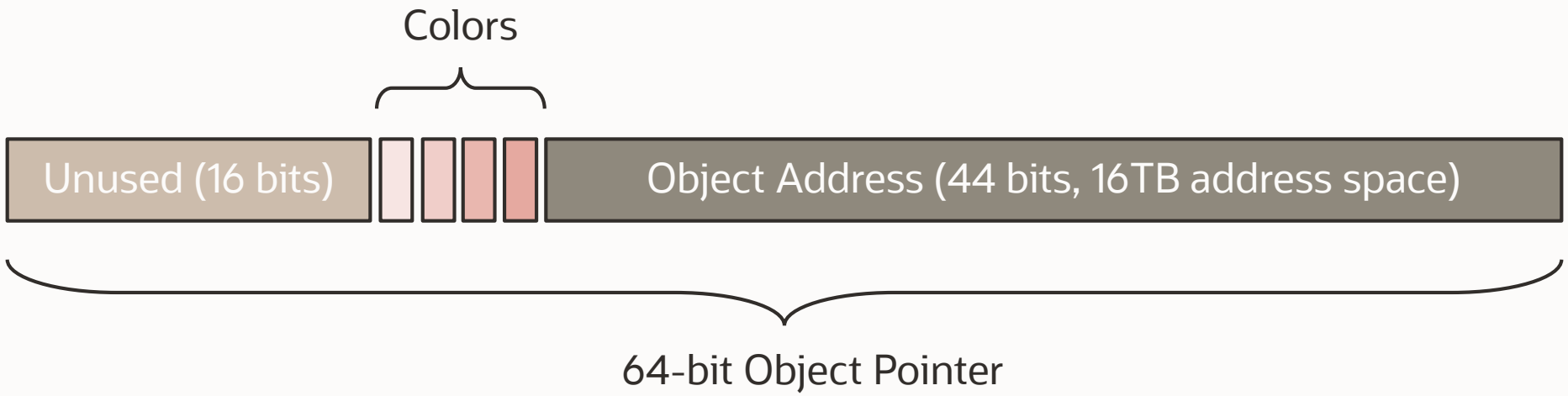


# ZGC Phases



# Colored Pointers

- Core design concept in ZGC
- **Metadata** stored in unused bits in 64-bit pointers



# Load Barrier

A small piece of code injected by the JIT in strategic places

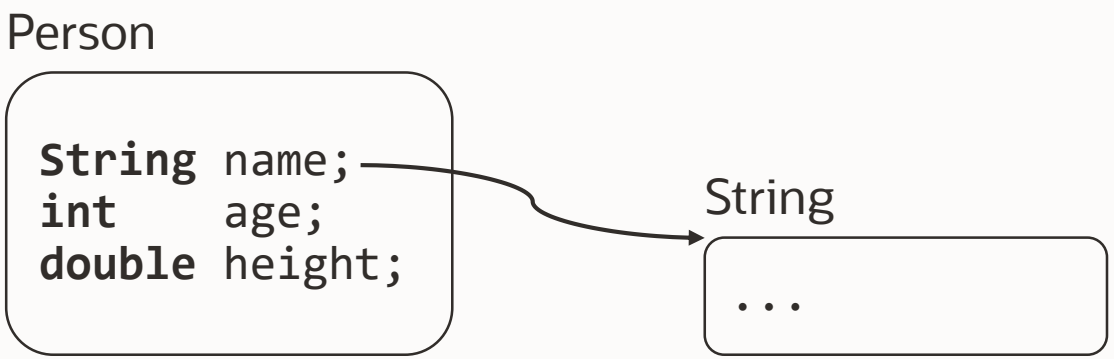
- When **loading an object reference from the heap**

Checks if the loaded object reference has a **bad** color

- If so, take **action** and **heal** it

# Load Barrier

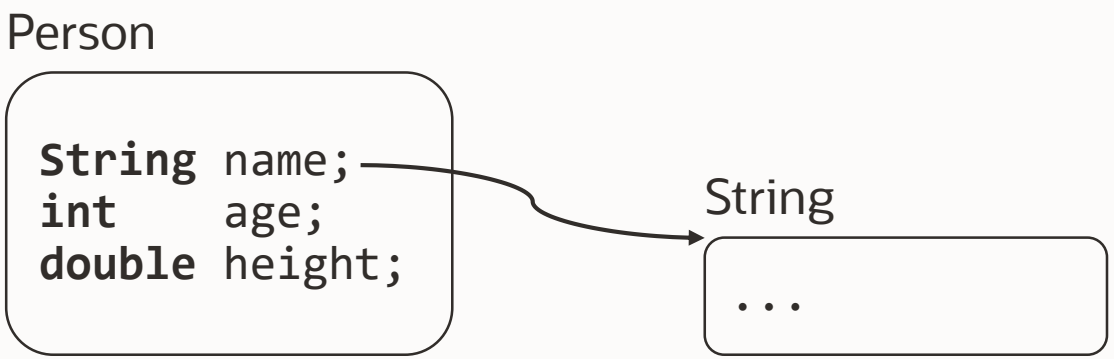
```
String n = person.name;          // Loading an object reference from heap
```



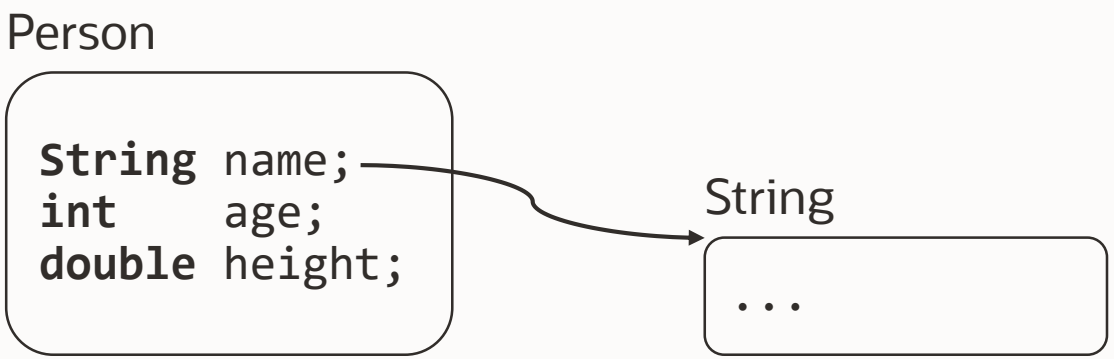
# Load Barrier

```
String n = person.name;          // Loading an object reference from heap
<load barrier needed here>
```



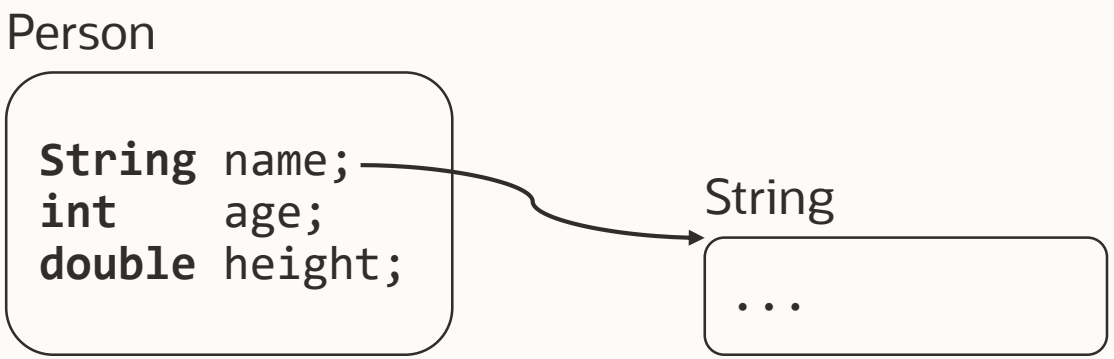
# Load Barrier

```
String n = person.name;          // Loading an object reference from heap
<load barrier needed here>
String p = n;                    // No barrier, not a load from heap
n.isEmpty();                     // No barrier, not a load from heap
int age = person.age;            // No barrier, not an object reference
```



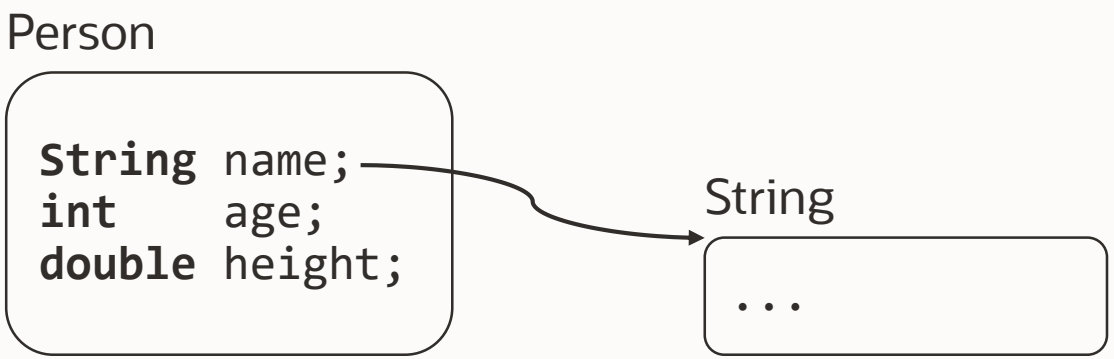
# Load Barrier

```
String n = person.name;          // Loading an object reference from heap
<load barrier needed here>
```



# Load Barrier

```
String n = person.name;        // Loading an object reference from heap
if (n & bad_bit_mask) {
  slow_path(register_for(n), address_of(person.name));
}
```



# Load Barrier

```
mov     0x10(%rax), %rbx        // String n = person.name;
test    %rbx, 0x20(%r15)        // Bad color?
jnz     slow_path               // Yes -> Enter slow path and
                                // mark/relocate/remap, adjust
                                // 0x10(%rax) and %rbx
```

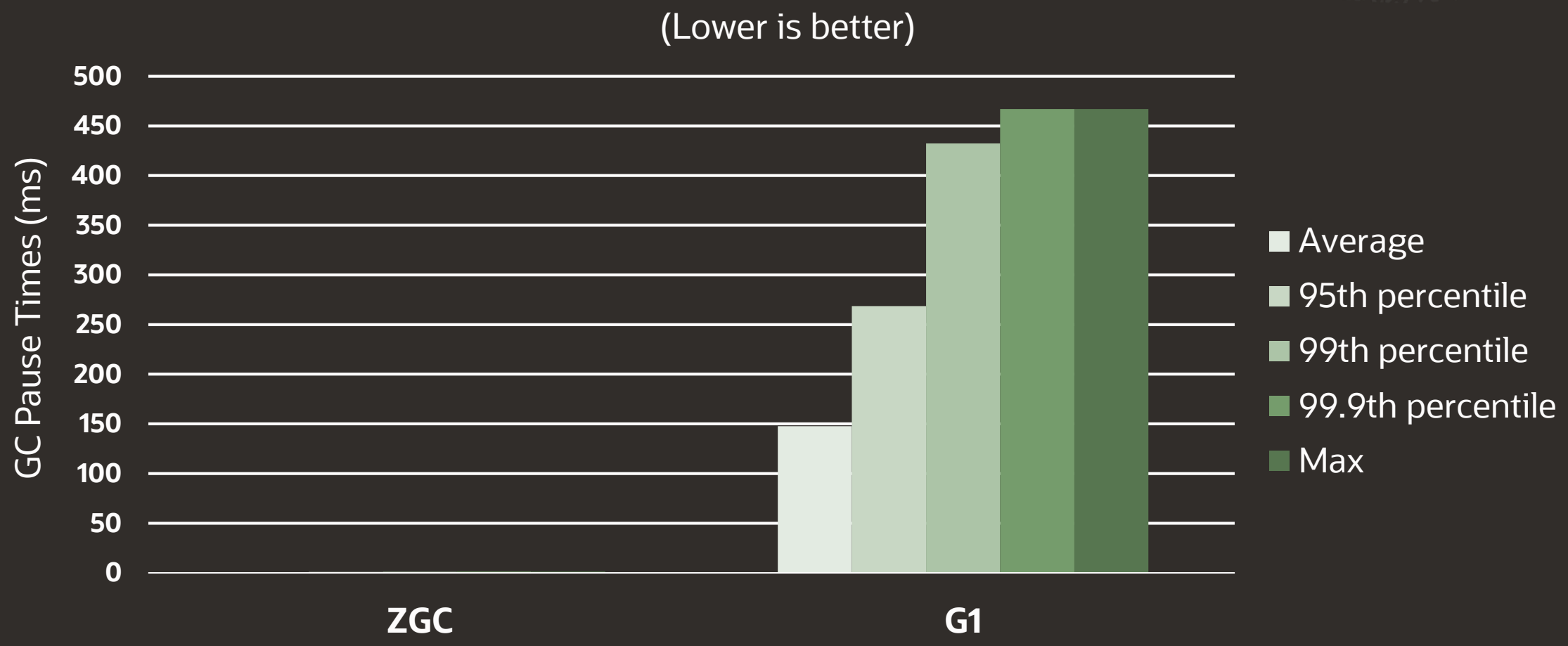
# Performance

# SPECjbb2015 – Benchmark Score

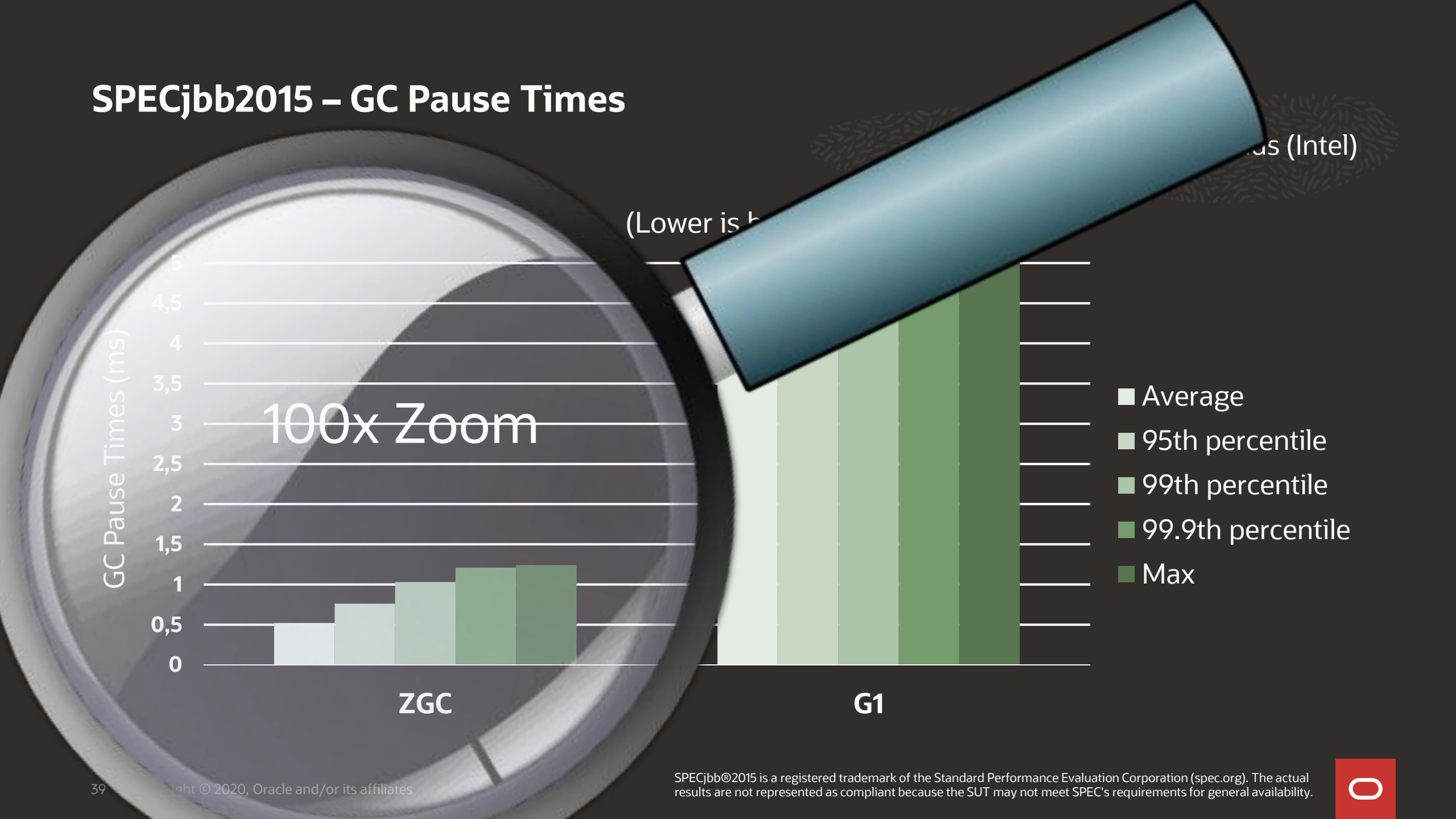
**128G** Heap
**32** Hyper-threads (Intel)

# SPECjbb2015 – GC Pause Times

**128G** Heap
**32** Hyper-threads (Intel)



# SPECjbb2015 – GC Pause Times





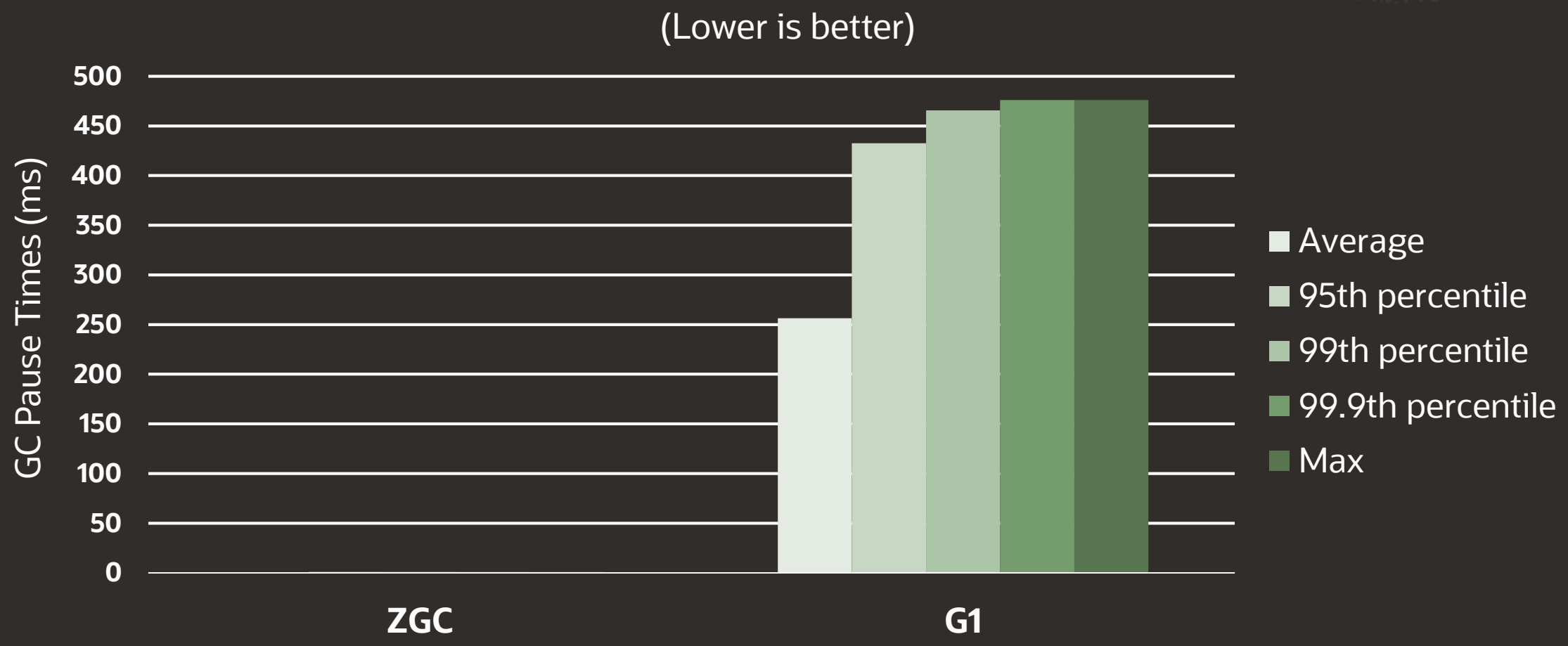


SPECjbb®2015 is a registered trademark of the Standard Performance Evaluation Corporation (spec.org). The actual results are not represented as compliant because the SUT may not meet SPEC's requirements for general availability.

# BigRamTester – GC Pause Times

Lots of heap fragmentation

**16G** Heap
**32** Hyper-threads (Intel)

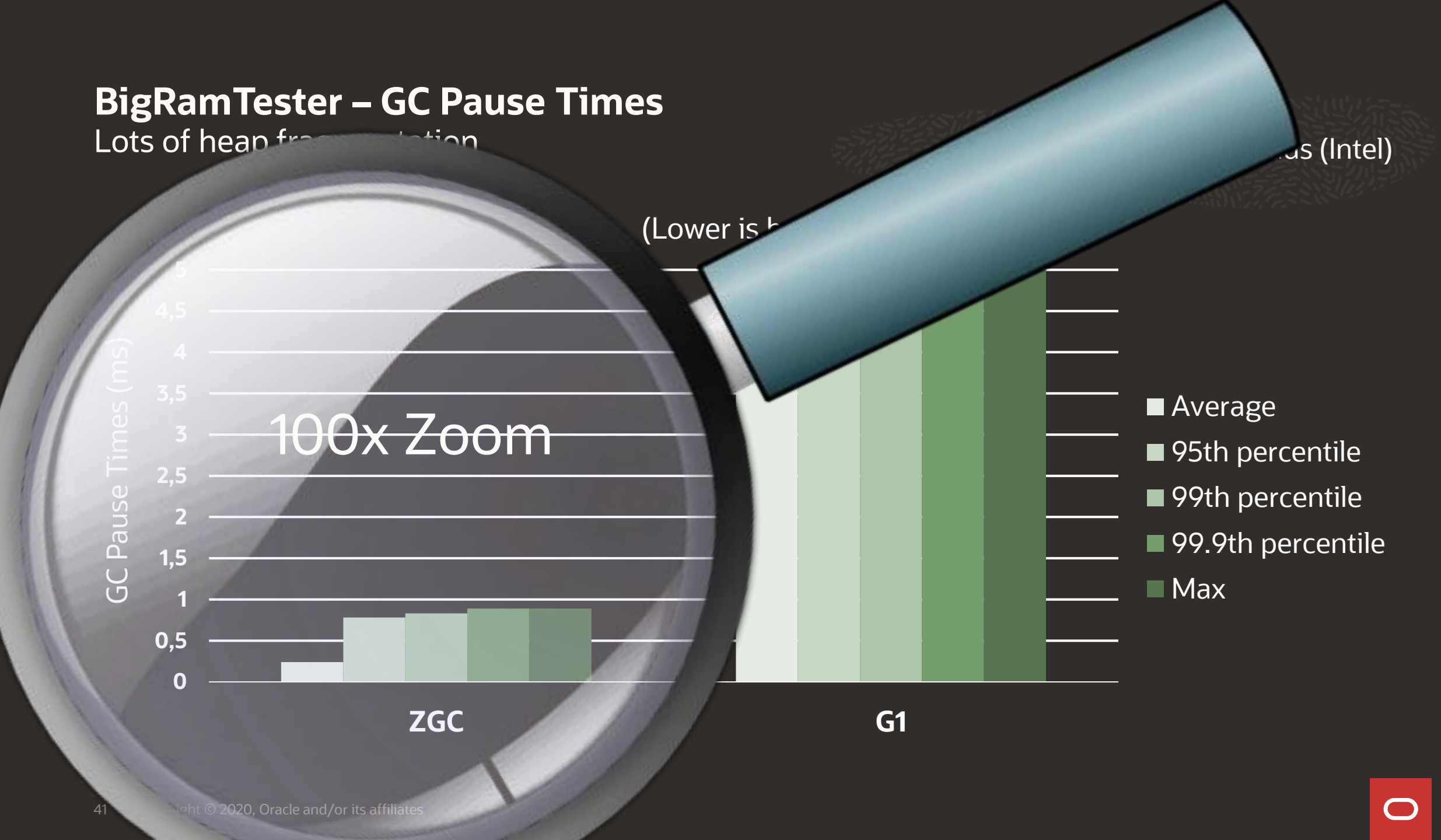


# BigRamTester – GC Pause Times

Lots of heap fr[illegible]tion

[illegible]ds (Intel)

(Lower is b[illegible]



# Using ZGC

# Recommendation

Use the latest JDK version!

(Lots of ZGC improvements since **JDK 11**, and they will continue to come...)

# Enable

```
-XX:+UseZGC
```

# Enable

```
-XX:+UnlockExperimentalVMOptions
-XX:+UseZGC
```

(JDK 14 and earlier)

# Tuning

# Tuning

## Set Max Heap Size

```
-Xmx<size>
```

# Tuning

*Maybe* Set Number of Concurrent GC Threads

```
-Xmx<size>
-XX:ConcGCThreads=<number>
```

# Tuning
That's it?

How much memory?

How much CPU-time?

```
-Xmx<size>
-XX:ConcGCThreads=<number>
```

# Logging

`-Xlog:gc` (basic)

`-Xlog:gc*` (detailed)

```
Garbage Collection (Proactive) 13426M(10%)->2492M(2%)
Garbage Collection (Allocation Rate) 87676M(67%)->19578M(15%)
Garbage Collection (Allocation Rate) 55302M(42%)->17646M(13%)
Garbage Collection (Allocation Rate) 61794M(47%)->26794M(20%)
Garbage Collection (Allocation Rate) 60856M(46%)->31926M(24%)
Garbage Collection (Allocation Rate) 52744M(40%)->38050M(29%)
Garbage Collection (Allocation Rate) 42542M(32%)->32204M(25%)
Garbage Collection (Allocation Rate) 49974M(38%)->8534M(7%)
Garbage Collection (System.gc()) 8534M(7%)->282M(0%)
Garbage Collection (Allocation Rate) 95454M(73%)->25660M(20%)
Garbage Collection (Allocation Rate) 42478M(32%)->23812M(18%)
Garbage Collection (Allocation Rate) 56714M(43%)->29090M(22%)
Garbage Collection (Allocation Rate) 62802M(48%)->28648M(22%)
Garbage Collection (Allocation Rate) 59748M(46%)->23770M(18%)
Garbage Collection (Allocation Rate) 74946M(57%)->23284M(18%)
Garbage Collection (System.gc()) 44902M(34%)->422M(0%)
Garbage Collection (Allocation Rate) 94510M(72%)->20456M(16%)
Garbage Collection (Allocation Rate) 59694M(46%)->25834M(20%)
Garbage Collection (Allocation Rate) 63494M(48%)->29128M(22%)
Garbage Collection (Allocation Rate) 59034M(45%)->27094M(21%)
Garbage Collection (Allocation Rate) 66110M(50%)->25278M(19%)
Garbage Collection (Allocation Rate) 73410M(56%)->27968M(21%)
Garbage Collection (Allocation Rate) 70010M(53%)->32236M(25%)
Garbage Collection (Allocation Rate) 64444M(49%)->27612M(21%)
Garbage Collection (Allocation Rate) 64484M(49%)->29910M(23%)
Garbage Collection (Allocation Rate) 64128M(49%)->33184M(25%)
Garbage Collection (Allocation Rate) 59148M(45%)->27800M(21%)
Garbage Collection (Allocation Rate) 63104M(48%)->27976M(21%)
Garbage Collection (Allocation Rate) 64418M(49%)->34390M(26%)
Garbage Collection (Allocation Rate) 52284M(40%)->30654M(23%)
Garbage Collection (Allocation Rate) 58746M(45%)->32028M(24%)
Garbage Collection (Allocation Rate) 59468M(45%)->32804M(25%)
Garbage Collection (Allocation Rate) 53342M(41%)->18436M(14%)
```

```
Garbage Collection (Proactive) 13426M(10%)->2492M(2%)
Garbage Collection (Allocation Rate) 87676M(67%)->19578M(15%)
Garbage Collection (Allocation Rate) 55302M(42%)->17646M(13%)
Garbage Collection (Allocation Rate) 61794M(47%)->26794M(20%)
Garbage Collection (Allocation Rate) 60856M(46%)->31926M(24%)
Garbage Collection (Allocation Rate) 52744M(40%)->38050M(29%)
Garbage Collection (Allocation Rate) 42542M(32%)->32204M(25%)
Garbage Collection (Allocation Rate) 49974M(38%)->8534M(7%)
Garbage Collection (System.gc()) 8534M(7%)->282M(0%)
Garbage Collection (Allocation Rate) 95454M(73%)->25660M(20%)
Garbage Collection (Allocation Rate) 42478M(32%)->23812M(18%)
Garbage Collection (Allocation Rate) 56714M(43%)->29090M(22%)
Garbage Collection (Allocation Rate) 62802M(48%)->28648M(22%)
Garbage Collection (Allocation Rate) 59748M(46%)->23770M(18%)
Garbage Collection (Allocation Rate) 74946M(57%)->23284M(18%)
Garbage Collection (System.gc()) 44902M(34%)->422M(0%)
Garbage Collection (Allocation Rate) 94510M(72%)->20456M(16%)
Garbage Collection (Allocation Rate) 59694M(46%)->25834M(20%)
Garbage Collection (Allocation Rate) 63494M(48%)->29128M(22%)
Garbage Collection (Allocation Rate) 59034M(45%)->27094M(21%)
Garbage Collection
Garbage Collection
Garbage Collection
Garbage Collection
Garbage Collection (Allocation Rate) 64484M(49%)->29910M(23%)
Garbage Collection (Allocation Rate) 64128M(49%)->33184M(25%)
Garbage Collection (Allocation Rate) 59148M(45%)->27800M(21%)
Garbage Collection (Allocation Rate) 63104M(48%)->27976M(21%)
Garbage Collection (Allocation Rate) 64418M(49%)->34390M(26%)
Garbage Collection (Allocation Rate) 52284M(40%)->30654M(23%)
Garbage Collection (Allocation Rate) 58746M(45%)->32028M(24%)
Garbage Collection (Allocation Rate) 59468M(45%)->32804M(25%)
Garbage Collection (Allocation Rate) 53342M(41%)->18436M(14%)
```

```
Garbage Collection (Allocation Rate) 95454M(73%)->25660M(20%)
```

```
Garbage Collection (Allocation Rate)
Pause Mark Start 0.483ms
Concurrent Mark 2885.207ms
Pause Mark End 0.439ms
Concurrent Process Non-Strong References 23.932ms
Concurrent Reset Relocation Set 5.682ms
Concurrent Select Relocation Set 215.098ms
Pause Relocate Start 0.531ms
Concurrent Relocate 1071.711ms
Load: 57.66/38.08/17.83
MMU: 2ms/41.5%, 5ms/76.6%, 10ms/87.3%, 20ms/92.1%, 50ms/96.1%, 100ms/98.0%
Mark: 4 stripe(s), 3 proactive flush(es), 1 terminate flush(es), 1 completion(s), 0 continuation(s)
NMethods: 5686 registered, 533 unregistered
Metaspace: 19M used, 20M capacity, 20M committed, 1042M reserved
Soft: 449128 encountered, 0 discovered, 0 enqueued
Weak: 4299 encountered, 2155 discovered, 119 enqueued
Final: 56 encountered, 6 discovered, 0 enqueued
Phantom: 89 encountered, 60 discovered, 0 enqueued
Small Pages: 26830 / 53660M(93%), Empty: 6768M(12%), Compacting: 46374M(80%)->700M(1%)
Medium Pages: 118 / 3776M(7%), Empty: 3520M(6%), Compacting: 256M(0%)->64M(0%)
Large Pages: 6 / 280M(0%), Empty: 0M(0%), Compacting: 0M(0%)->0M(0%)
Relocation: Successful
Min Capacity: 131072M(100%)
Max Capacity: 131072M(100%)
Soft Max Capacity: 131072M(100%)
                Mark Start          Mark End        Relocate Start      Relocate End            High               Low
 Capacity:   131072M (100%)     131072M (100%)     131072M (100%)      131072M (100%)     131072M (100%)     131072M (100%)
  Reserve:       72M (0%)           72M (0%)           72M (0%)            72M (0%)           72M (0%)           72M (0%)
     Free:    73284M (56%)       53408M (41%)       61818M (47%)       100740M (77%)      101796M (78%)       53120M (41%)
     Used:    57716M (44%)       77592M (59%)       69182M (53%)        30260M (23%)       77880M (59%)       29204M (22%)
     Live:         -              1405M (1%)         1405M (1%)          1405M (1%)             -                  -
Allocated:         -             19876M (15%)       21754M (17%)        29462M (22%)             -                  -
  Garbage:         -             56310M (43%)       46022M (35%)         4990M (4%)              -                  -
Reclaimed:         -                  -             10288M (8%)         51320M (39%)             -                  -
Garbage Collection (Allocation Rate) 57716M(44%)->30260M(23%)
```

```
Garbage Collection (Allocation Rate)
Pause Mark Start 0.483ms
Concurrent Mark 2885.207ms
Pause Mark End 0.439ms
Concurrent Process Non-Strong References 23.932ms
Concurrent Reset Relocation Set 5.682ms
Concurrent Select Relocation Set 215.098ms
Pause Relocate Start 0.531ms
Concurrent Relocate 1071.711ms
Load: 57.66/38.08/17.83
MMU: 2ms/41.5%, 5ms/76.6%, 10ms/87.3%, 20ms/
Mark: 4 stripe(s), 3 proactive flush(es), 1
NMethods: 5686 registered, 533 unregistered
Metaspace: 19M used, 20M capacity, 20M commi
Soft: 449128 encountered, 0 discovered, 0 en
Weak: 4299 encountered, 2155 discovered, 119
Final: 56 encountered, 6 discovered, 0 enque
Phantom: 89 encountered, 60 discovered, 0 en
Small Pages: 26830 / 53660M(93%), Empty: 676
Medium Pages: 118 / 3776M(7%), Empty: 3520M(
Large Pages: 6 / 280M(0%), Empty: 0M(0%), Co
Relocation: Successful
Min Capacity: 131072M(100%)
Max Capacity: 131072M(100%)
Soft Max Capacity: 131072M(100%)
                 Mark Start          Mark End        Relocate Start      Relocate End           High               Low
 Capacity:   131072M (100%)     131072M (100%)     131072M (100%)     131072M (100%)     131072M (100%)     131072M (100%)
  Reserve:       72M (0%)           72M (0%)           72M (0%)           72M (0%)           72M (0%)           72M (0%)
     Free:    73284M (56%)       53408M (41%)       61818M (47%)      100740M (77%)      101796M (78%)       53120M (41%)
     Used:    57716M (44%)       77592M (59%)       69182M (53%)       30260M (23%)       77880M (59%)       29204M (22%)
     Live:         -              1405M (1%)         1405M (1%)         1405M (1%)            -                  -
Allocated:         -             19876M (15%)       21754M (17%)       29462M (22%)            -                  -
  Garbage:         -             56310M (43%)       46022M (35%)        4990M (4%)             -                  -
Reclaimed:         -                  -             10288M (8%)        51320M (39%)            -                  -
Garbage Collection (Allocation Rate) 57716M(44%)->30260M(23%)
```


```
Pause Mark Start 0.483ms
Concurrent Mark 2885.207ms
Pause Mark End 0.439ms
Concurrent Process Non-Strong References 23.932ms
Concurrent Reset Relocation Set 5.682ms
Concurrent Select Relocation Set 215.098ms
Pause Relocate Start 0.531ms
Concurrent Relocate 1071.711ms
```

```
Garbage Collection (Allocation Rate)
Pause Mark Start 0.483ms
Concurrent Mark 2885.207ms
Pause Mark End 0.439ms
Concurrent Process Non-Strong References 23.932ms
Concurrent Reset Relocation Set 5.682ms
Concurrent Selec
Pause Relocate St
Concurrent Reloca
Load: 57.66/38.08
MMU: 2ms/41.5%, 5
Mark: 4 stripe(s)
NMethods: 5686 re
Metaspace: 19M us
Soft: 449128 enco
Weak: 4299 encour
Final: 56 encount
Phantom: 89 encou
Small Pages: 268
Medium Pages: 118
Large Pages: 6 / 280M(0%), Empty: 0M(0%), Compacting: 0M(0%)->0M(0%)
Relocation: Successful
Min Capacity: 131072M(100%)
Max Capacity: 131072M(100%)
Soft Max Capacity: 131072M(100%)
```

| | Mark Start | Mark End | Relocate Start | Relocate End | High | Low |
|---|---|---|---|---|---|---|
| Capacity: | 131072M (100%) | 131072M (100%) | 131072M (100%) | 131072M (100%) | 131072M (100%) | 131072M (100%) |
| Reserve: | 72M (0%) | 72M (0%) | 72M (0%) | 72M (0%) | 72M (0%) | 72M (0%) |
| Free: | 73284M (56%) | 53408M (41%) | 61818M (47%) | 100740M (77%) | 101796M (78%) | 53120M (41%) |
| Used: | 57716M (44%) | 77592M (59%) | 69182M (53%) | 30260M (23%) | 77880M (59%) | 29204M (22%) |
| Live: | - | 1405M (1%) | 1405M (1%) | 1405M (1%) | - | - |
| Allocated: | - | 19876M (15%) | 21754M (17%) | 29462M (22%) | - | - |
| Garbage: | - | 56310M (43%) | 46022M (35%) | 4990M (4%) | - | - |
| Reclaimed: | - | - | 10288M (8%) | 51320M (39%) | - | - |

```
Garbage Collection (Allocation Rate) 57716M(44%)->30260M(23%)
```

```
Garbage Collection (Allocation Rate)
Pause Mark Start 0.483ms
Concurrent Mark 2885.207ms
Pause Mark End 0.439ms
Concurrent Process Non-Strong References 23.932ms
Concurrent Reset Relocation Set 5.682ms
Concurrent Select Relocation Set 215.098ms
Pause Relocate Start 0.531ms
Concurrent Relocate 1071.711ms
Load: 57.66/38.08/17.83
MMU: 2ms/41.5%, 5ms/76.6%, 10ms/87.3%, 20ms/92.1%, 50ms/96.1%, 100ms/98.0%
Mark: 4 stripe(s), 3 proactive flush(es), 1 terminate flush(es), 1 completion(s), 0 continuation(s)
NMethods: 5686 registered, 533 unregistered
Metaspace: 19M used, 20M capacity, 20M committed, 1042M reserved
Soft: 449128 encountered, 0 discovered, 0 enqueued
Weak: 4299 encountered, 2155 discovered, 119 enqueued
Final: 56 encountered, 6 discovered, 0 enqueued
Phantom: 89 encountered, 60 discovered, 0 enqueued
Small Pages: 26830 / 53660M(93%), Empty: 6768M(12%), Compacting: 46374M(80%)->700M(1%)
Medium Pages: 118 / 3776M(7%), Empty: 3520M(6%), Compacting: 256M(0%)->64M(0%)
Large Pages: 6 / 280M(0%), Empty: 0M(0%), Compacting: 0M(0%)->0M(0%)
Relocation: Successful
Min Capacity: 131072M(100%)
Max Capacity: 131072M(100%)
Soft Max Capacity: 131072M(100%)
                Mark Start          Mark End        Relocate Start      Relocate End            High               Low
 Capacity:   131072M (100%)     131072M (100%)     131072M (100%)      131072M (100%)     131072M (100%)     131072M (100%)
  Reserve:       72M (0%)           72M (0%)           72M (0%)            72M (0%)           72M (0%)           72M (0%)
     Free:    73284M (56%)       53408M (41%)       61818M (47%)       100740M (77%)      101796M (78%)       53120M (41%)
     Used:    57716M (44%)       77592M (59%)       69182M (53%)        30260M (23%)       77880M (59%)       29204M (22%)
     Live:         -              1405M (1%)         1405M (1%)          1405M (1%)             -                  -
Allocated:         -             19876M (15%)       21754M (17%)        29462M (22%)             -                  -
  Garbage:         -             56310M (43%)       46022M (35%)         4990M (4%)              -                  -
Reclaimed:         -                  -             10288M (8%)         51320M (39%)             -                  -
Garbage Collection (Allocation Rate) 57716M(44%)->30260M(23%)
```

```
=== Garbage Collection Statistics ==========================================================================================================
                                                      Last 10s              Last 10m              Last 10h                Total
                                                      Avg / Max             Avg / Max             Avg / Max             Avg / Max
  Collector: Garbage Collection Cycle               393.373 / 393.373    2457.026 / 4420.944   2457.026 / 4420.944   2457.026 / 4420.944    ms
 Contention: Mark Segment Reset Contention               4 / 47                1 / 165               1 / 165               1 / 165         ops/s
 Contention: Mark SeqNum Reset Contention                0 / 0                 0 / 1                 0 / 1                 0 / 1           ops/s
 Contention: Relocation Contention                       0 / 6                13 / 1609             13 / 1609             13 / 1609        ops/s
   Critical: Allocation Stall                      0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000       ms
   Critical: Allocation Stall                          0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
   Critical: GC Locker Stall                       0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000       ms
   Critical: GC Locker Stall                           0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Allocation Rate                        862 / 2892           2229 / 13172          2229 / 13172          2229 / 13172        MB/s
     Memory: Heap Used After Mark                  6038 / 6038          55657 / 107472        55657 / 107472        55657 / 107472       MB
     Memory: Heap Used After Relocation            1496 / 1496          14059 / 31084         14059 / 31084         14059 / 31084        MB
     Memory: Heap Used Before Mark                 6038 / 6038          47444 / 94452         47444 / 94452         47444 / 94452        MB
     Memory: Heap Used Before Relocation           3452 / 3452          47152 / 94430         47152 / 94430         47152 / 94430        MB
     Memory: Out Of Memory                             0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Page Cache Flush                          0 / 0                 0 / 0                 0 / 0                 0 / 0           MB/s
     Memory: Page Cache Hit L1                       350 / 1183            765 / 3549            765 / 3549            765 / 3549        ops/s
     Memory: Page Cache Hit L2                         0 / 0                 4 / 234               4 / 234               4 / 234         ops/s
     Memory: Page Cache Hit L3                         2 / 7               127 / 2928            127 / 2928            127 / 2928        ops/s
     Memory: Page Cache Miss                           0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Uncommit                                  0 / 0                 0 / 0                 0 / 0                 0 / 0           MB/s
     Memory: Undo Object Allocation Failed             0 / 0                 0 / 53                0 / 53                0 / 53          ops/s
     Memory: Undo Object Allocation Succeeded          0 / 6                12 / 1609             12 / 1609             12 / 1609        ops/s
     Memory: Undo Page Allocation                      0 / 0                 6 / 463               6 / 463               6 / 463         ops/s
      Phase: Concurrent Mark                     317.841 / 317.841    1564.321 / 3056.192   1564.321 / 3056.192   1564.321 / 3056.192    ms
      Phase: Concurrent Mark Continue              0.000 / 0.000         8.173 / 8.173         8.173 / 8.173         8.173 / 8.173       ms
      Phase: Concurrent Process Non-Strong References  1.533 / 1.533    16.783 / 57.370       16.783 / 57.370       16.783 / 57.370      ms
      Phase: Concurrent Relocate                  49.909 / 49.909      748.800 / 1299.535    748.800 / 1299.535    748.800 / 1299.535    ms
      Phase: Concurrent Reset Relocation Set       5.138 / 5.138         4.566 / 12.218        4.566 / 12.218        4.566 / 12.218      ms
      Phase: Concurrent Select Relocation Set     16.672 / 16.672      111.356 / 259.445     111.356 / 259.445     111.356 / 259.445     ms
      Phase: Pause Mark End                        0.063 / 0.063         0.411 / 1.142         0.411 / 1.142         0.411 / 1.142       ms
      Phase: Pause Mark Start                      0.250 / 0.250         0.407 / 0.706         0.407 / 0.706         0.407 / 0.706       ms
      Phase: Pause Relocate Start                  0.513 / 0.513         0.598 / 1.170         0.598 / 1.170         0.598 / 1.170       ms
   Subphase: Concurrent Classes Purge              0.112 / 0.112         0.293 / 0.709         0.293 / 0.709         0.293 / 0.709       ms
   Subphase: Concurrent Classes Unlink             0.897 / 0.897         5.852 / 18.071        5.852 / 18.071        5.852 / 18.071      ms
   Subphase: Concurrent Mark                     315.060 / 317.380     814.912 / 3055.673    814.912 / 3055.673    814.912 / 3055.673    ms
   Subphase: Concurrent Mark Idle                  1.075 / 1.161         1.958 / 21.000        1.958 / 21.000        1.958 / 21.000      ms
   Subphase: Concurrent Mark Try Flush             0.137 / 0.449         6.628 / 51.534        6.628 / 51.534        6.628 / 51.534      ms
   ...
```

```
=== Garbage Collection Statistics ==========================================================================================
                                                   Last 10s              Last 10m              Last 10h                Total
                                                   Avg / Max             Avg / Max             Avg / Max             Avg / Max
  Collector: Garbage Collection Cycle            393.373 / 393.373    2457.026 / 4420.944   2457.026 / 4420.944   2457.026 / 4420.944    ms
 Contention: Mark Segment Reset Contention             4 / 47                1 / 165               1 / 165               1 / 165         ops/s
 Contention: Mark SeqNum Reset Contention              0 / 0                 0 / 1                 0 / 1                 0 / 1           ops/s
 Contention: Relocation Contention                     0 / 6                13 / 1609             13 / 1609             13 / 1609        ops/s
   Critical: Allocation Stall                    0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000       ms
   Critical: Allocation Stall                        0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
   Critical: GC Locker Stall                     0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000       ms
   Critical: GC Locker Stall                         0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Allocation Rate                      862 / 2892           2229 / 13172          2229 / 13172          2229 / 13172        MB/s
     Memory: Heap Used After Mark                6038 / 6038          55657 / 107472        55657 / 107472        55657 / 107472       MB
     Memory: Heap Used After Relocation          1496 / 1496          14059 / 31084         14059 / 31084         14059 / 31084        MB
     Memory: Heap Used Before Mark               6038 / 6038          47444 / 94452         47444 / 94452         47444 / 94452        MB
     Memory: Heap Used Before Relocation         3452 / 3452          47152 / 94430         47152 / 94430         47152 / 94430        MB
     Memory: Out Of Memory                          0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Page Cache Flush                       0 / 0                 0 / 0                 0 / 0                 0 / 0           MB/s
     Memory: Page Cache Hit L1                    350 / 1183            765 / 3549            765 / 3549            765 / 3549        ops/s
```

| | Last 10s Avg / Max | Last 10m Avg / Max | Last 10h Avg / Max | Total Avg / Max | |
|---|---|---|---|---|---|
| Allocation Rate | 862 / 2892 | 2229 / 13172 | 2229 / 13172 | 2229 / 13172 | MB/s |

```
     Memory: Undo Page Allocation                   0 / 0                 6 / 463               6 / 463               6 / 463         ops/s
      Phase: Concurrent Mark                  317.841 / 317.841    1564.321 / 3056.192   1564.321 / 3056.192   1564.321 / 3056.192    ms
      Phase: Concurrent Mark Continue           0.000 / 0.000        8.173 / 8.173         8.173 / 8.173         8.173 / 8.173       ms
      Phase: Concurrent Process Non-Strong References  1.533 / 1.533   16.783 / 57.370       16.783 / 57.370       1783 / 57.370       ms
      Phase: Concurrent Relocate               49.909 / 49.909      748.800 / 1299.535    748.800 / 1299.535    748.800 / 1299.535    ms
      Phase: Concurrent Reset Relocation Set    5.138 / 5.138        4.566 / 12.218        4.566 / 12.218        4.566 / 12.218      ms
      Phase: Concurrent Select Relocation Set  16.672 / 16.672      111.356 / 259.445     111.356 / 259.445     111.356 / 259.445    ms
      Phase: Pause Mark End                     0.063 / 0.063        0.411 / 1.142         0.411 / 1.142         0.411 / 1.142       ms
      Phase: Pause Mark Start                   0.250 / 0.250        0.407 / 0.706         0.407 / 0.706         0.407 / 0.706       ms
      Phase: Pause Relocate Start               0.513 / 0.513        0.598 / 1.170         0.598 / 1.170         0.598 / 1.170       ms
   Subphase: Concurrent Classes Purge           0.112 / 0.112        0.293 / 0.709         0.293 / 0.709         0.293 / 0.709       ms
   Subphase: Concurrent Classes Unlink          0.897 / 0.897        5.852 / 18.071        5.852 / 18.071        5.852 / 18.071      ms
   Subphase: Concurrent Mark                  315.060 / 317.380    814.912 / 3055.673    814.912 / 3055.673    814.912 / 3055.673    ms
   Subphase: Concurrent Mark Idle               1.075 / 1.161        1.958 / 21.000        1.958 / 21.000        1.958 / 21.000      ms
   Subphase: Concurrent Mark Try Flush          0.137 / 0.449        6.628 / 51.534        6.628 / 51.534        6.628 / 51.534      ms
   ...
```

```
=== Garbage Collection Statistics ==========================================================================================
                                                    Last 10s              Last 10m              Last 10h                Total
                                                    Avg / Max             Avg / Max             Avg / Max             Avg / Max
  Collector: Garbage Collection Cycle             393.373 / 393.373    2457.026 / 4420.944   2457.026 / 4420.944   2457.026 / 4420.944    ms
 Contention: Mark Segment Reset Contention              4 / 47                1 / 165               1 / 165               1 / 165        ops/s
 Contention: Mark SeqNum Reset Contention               0 / 0                 0 / 1                 0 / 1                 0 / 1          ops/s
 Contention: Relocation Contention                      0 / 6                13 / 1609             13 / 1609             13 / 1609       ops/s
   Critical: Allocation Stall                     0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000      ms
   Critical: Allocation Stall                         0 / 0                 0 / 0                 0 / 0                 0 / 0          ops/s
   Critical: GC Locker Stall                      0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000      ms
   Critical: GC Locker Stall                          0 / 0                 0 / 0                 0 / 0                 0 / 0          ops/s
```

| | Last 10s Avg / Max | Last 10m Avg / Max | Last 10h Avg / Max | Total Avg / Max | |
|---|---|---|---|---|---|
| Pause Mark End | 0.063 / 0.063 | 0.411 / 1.142 | 0.411 / 1.142 | 0.411 / 1.142 | ms |
| Pause Mark Start | 0.250 / 0.250 | 0.407 / 0.706 | 0.407 / 0.706 | 0.407 / 0.706 | ms |
| Pause Relocate Start | 0.513 / 0.513 | 0.598 / 1.170 | 0.598 / 1.170 | 0.598 / 1.170 | ms |

```
     Memory: Page Cache Hit L3                          2 / 7                127 / 2928            127 / 2928            127 / 2928       ops/s
     Memory: Page Cache Miss                            0 / 0                 0 / 0                 0 / 0                 0 / 0          ops/s
     Memory: Uncommit                                   0 / 0                 0 / 0                 0 / 0                 0 / 0          MB/s
     Memory: Undo Object Allocation Failed              0 / 0                 0 / 53                0 / 53                0 / 53         ops/s
     Memory: Undo Object Allocation Succeeded           0 / 6                12 / 1609             12 / 1609             12 / 1609       ops/s
     Memory: Undo Page Allocation                       0 / 0                 6 / 463               6 / 463               6 / 463        ops/s
      Phase: Concurrent Mark                      317.841 / 317.841    1564.321 / 3056.192   1564.321 / 3056.192   1564.321 / 3056.192    ms
      Phase: Concurrent Mark Continue               0.000 / 0.000         8.173 / 8.173         8.173 / 8.173         8.173 / 8.173      ms
      Phase: Concurrent Process Non-Strong References 1.533 / 1.533      16.783 / 57.370       16.783 / 57.370        1783 / 57.370     ms
      Phase: Concurrent Relocate                   49.909 / 49.909     748.800 / 1299.535    748.800 / 1299.535    748.800 / 1299.535    ms
      Phase: Concurrent Reset Relocation Set        5.138 / 5.138        4.566 / 12.218        4.566 / 12.218        4.566 / 12.218     ms
      Phase: Concurrent Select Relocation Set      16.672 / 16.672      111.356 / 259.445     111.356 / 259.445     111.356 / 259.445    ms
      Phase: Pause Mark End                         0.063 / 0.063        0.411 / 1.142         0.411 / 1.142         0.411 / 1.142      ms
      Phase: Pause Mark Start                       0.250 / 0.250        0.407 / 0.706         0.407 / 0.706         0.407 / 0.706      ms
      Phase: Pause Relocate Start                   0.513 / 0.513        0.598 / 1.170         0.598 / 1.170         0.598 / 1.170      ms
   Subphase: Concurrent Classes Purge               0.112 / 0.112        0.293 / 0.709         0.293 / 0.709         0.293 / 0.709      ms
   Subphase: Concurrent Classes Unlink              0.897 / 0.897        5.852 / 18.071        5.852 / 18.071        5.852 / 18.071     ms
   Subphase: Concurrent Mark                      315.060 / 317.380    814.912 / 3055.673    814.912 / 3055.673    814.912 / 3055.673    ms
   Subphase: Concurrent Mark Idle                   1.075 / 1.161        1.958 / 21.000        1.958 / 21.000        1.958 / 21.000     ms
   Subphase: Concurrent Mark Try Flush              0.137 / 0.449        6.628 / 51.534        6.628 / 51.534        6.628 / 51.534     ms
   ...
```

```
=== Garbage Collection Statistics ==========================================================================================================
                                                      Last 10s              Last 10m              Last 10h                Total
                                                      Avg / Max             Avg / Max             Avg / Max             Avg / Max
  Collector: Garbage Collection Cycle               393.373 / 393.373    2457.026 / 4420.944   2457.026 / 4420.944   2457.026 / 4420.944    ms
 Contention: Mark Segment Reset Contention               4 / 47                1 / 165               1 / 165               1 / 165         ops/s
 Contention: Mark SeqNum Reset Contention                0 / 0                 0 / 1                 0 / 1                 0 / 1           ops/s
 Contention: Relocation Contention                       0 / 6                13 / 1609             13 / 1609             13 / 1609        ops/s
   Critical: Allocation Stall                      0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000       ms
   Critical: Allocation Stall                          0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
   Critical: GC Locker Stall                       0.000 / 0.000         0.000 / 0.000         0.000 / 0.000         0.000 / 0.000       ms
   Critical: GC Locker Stall                           0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Allocation Rate                        862 / 2892           2229 / 13172          2229 / 13172          2229 / 13172        MB/s
     Memory: Heap Used After Mark                  6038 / 6038          55657 / 107472        55657 / 107472        55657 / 107472       MB
     Memory: Heap Used After Relocation            1496 / 1496          14059 / 31084         14059 / 31084         14059 / 31084        MB
     Memory: Heap Used Before Mark                 6038 / 6038          47444 / 94452         47444 / 94452         47444 / 94452        MB
     Memory: Heap Used Before Relocation           3452 / 3452          47152 / 94430         47152 / 94430         47152 / 94430        MB
     Memory: Out Of Memory                             0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Page Cache Flush                          0 / 0                 0 / 0                 0 / 0                 0 / 0           MB/s
     Memory: Page Cache Hit L1                       350 / 1183            765 / 3549            765 / 3549            765 / 3549        ops/s
     Memory: Page Cache Hit L2                         0 / 0                 4 / 234               4 / 234               4 / 234         ops/s
     Memory: Page Cache Hit L3                         2 / 7               127 / 2928            127 / 2928            127 / 2928        ops/s
     Memory: Page Cache Miss                           0 / 0                 0 / 0                 0 / 0                 0 / 0           ops/s
     Memory: Uncommit                                  0 / 0                 0 / 0                 0 / 0                 0 / 0           MB/s
     Memory: Undo Object Allocation Failed             0 / 0                 0 / 53                0 / 53                0 / 53          ops/s
     Memory: Undo Object Allocation Succeeded          0 / 6                12 / 1609             12 / 1609             12 / 1609        ops/s
     Memory: Undo Page Allocation                      0 / 0                 6 / 463               6 / 463               6 / 463         ops/s
      Phase: Concurrent Mark                     317.841 / 317.841    1564.321 / 3056.192   1564.321 / 3056.192   1564.321 / 3056.192    ms
      Phase: Concurrent Mark Continue              0.000 / 0.000         8.173 / 8.173         8.173 / 8.173         8.173 / 8.173       ms
      Phase: Concurrent Process Non-Strong References  1.533 / 1.533    16.783 / 57.370       16.783 / 57.370       16.783 / 57.370      ms
      Phase: Concurrent Relocate                  49.909 / 49.909      748.800 / 1299.535    748.800 / 1299.535    748.800 / 1299.535    ms
      Phase: Concurrent Reset Relocation Set       5.138 / 5.138         4.566 / 12.218        4.566 / 12.218        4.566 / 12.218      ms
      Phase: Concurrent Select Relocation Set     16.672 / 16.672      111.356 / 259.445     111.356 / 259.445     111.356 / 259.445     ms
      Phase: Pause Mark End                        0.063 / 0.063         0.411 / 1.142         0.411 / 1.142         0.411 / 1.142       ms
      Phase: Pause Mark Start                      0.250 / 0.250         0.407 / 0.706         0.407 / 0.706         0.407 / 0.706       ms
      Phase: Pause Relocate Start                  0.513 / 0.513         0.598 / 1.170         0.598 / 1.170         0.598 / 1.170       ms
   Subphase: Concurrent Classes Purge              0.112 / 0.112         0.293 / 0.709         0.293 / 0.709         0.293 / 0.709       ms
   Subphase: Concurrent Classes Unlink             0.897 / 0.897         5.852 / 18.071        5.852 / 18.071        5.852 / 18.071      ms
   Subphase: Concurrent Mark                     315.060 / 317.380     814.912 / 3055.673    814.912 / 3055.673    814.912 / 3055.673    ms
   Subphase: Concurrent Mark Idle                  1.075 / 1.161         1.958 / 21.000        1.958 / 21.000        1.958 / 21.000      ms
   Subphase: Concurrent Mark Try Flush             0.137 / 0.449         6.628 / 51.534        6.628 / 51.534        6.628 / 51.534      ms
   ...
```

# Roadmap

Sub-millisecond max pause times

Generational ZGC

# Sub-millisecond max pause times

# Initial Goals

Max GC pause time

**10ms**

Multi-terabyte heaps

**TB**

Max throughput reduction

**15%**

# Current Goals

Max GC pause time

~~10 ms~~

1 ms

Multi-terabyte heaps

TB

Max throughput reduction

15%

# What's Concurrent?

*) Old Gen Only
**) Not yet

| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | - | - | ✓ * | ✓ * | ✓ |
| Compaction | - | - | - | - | ✓ |
| Reference Processing | - | - | - | - | ✓ |
| Relocation Set Selection | - | - | - | - | ✓ |
| StringTable Cleaning | - | - | - | - | ✓ |
| JNI WeakRef Cleaning | - | - | - | - | ✓ |
| JNI GlobalRefs Scanning | - | - | - | - | ✓ |
| Class Unloading | - | - | - | - | ✓ |
| Thread Stack Scanning | - | - | - | - | - ** |

*) Old Gen Only

# What's Concurrent?

| | Serial | Parallel | G1 | CMS | ZGC |
|---|---|---|---|---|---|
| Marking | - | - | ✓ * | ✓ * | ✓ |
| Compaction | - | - | - | - | ✓ |
| Reference Processing | - | - | - | - | ✓ |
| Relocation Set Selection | - | - | - | - | ✓ |
| StringTable Cleaning | - | - | - | - | ✓ |
| JNI WeakRef Cleaning | - | - | - | - | ✓ |
| JNI GlobalRefs Scanning | - | - | - | - | ✓ |
| Class Unloading | - | - | - | - | ✓ |
| Thread Stack Scanning | - | - | - | - | ✓ |

# ZGC pause times **do** increase with the root-set size

(Number of Java Threads)

do not

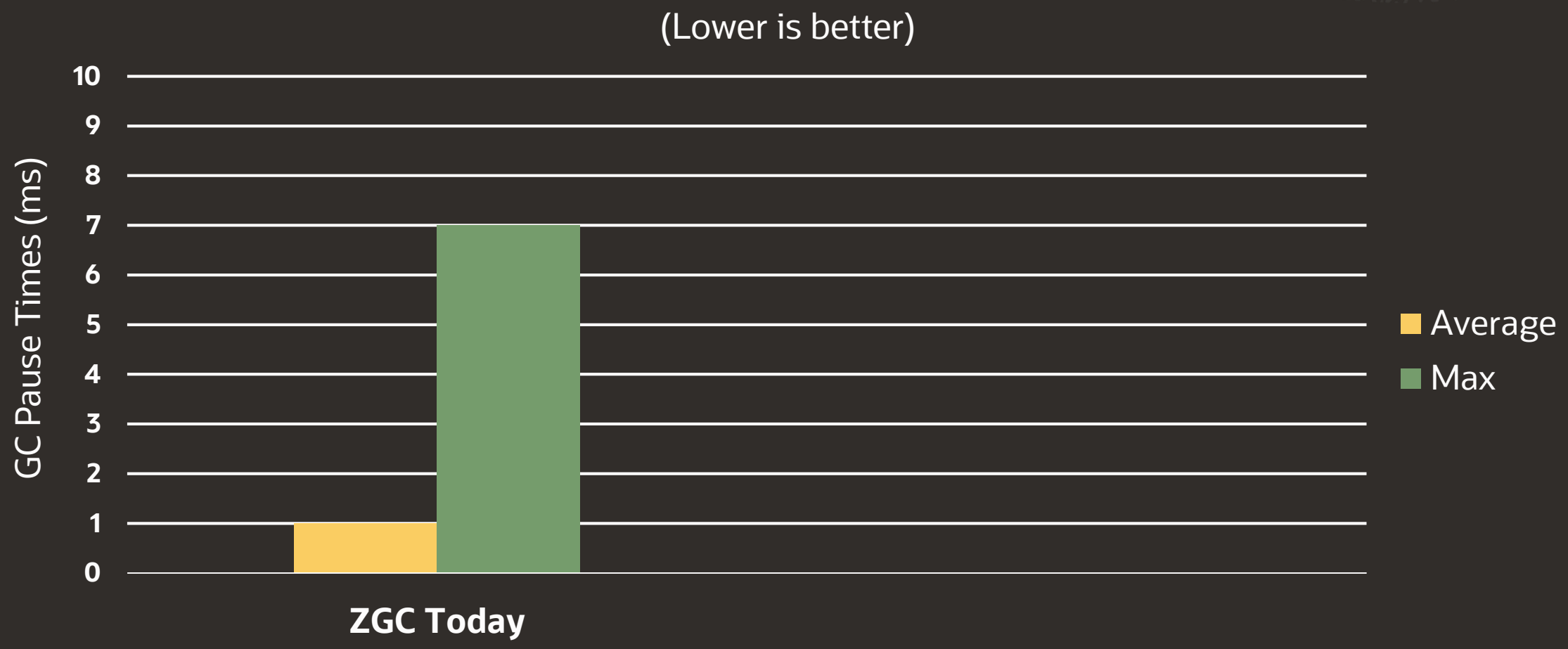
ZGC pause times ~~do~~ increase with the root-set size

(Number of Java Threads)

# JEP 376: ZGC: Concurrent Thread-Stack Processing

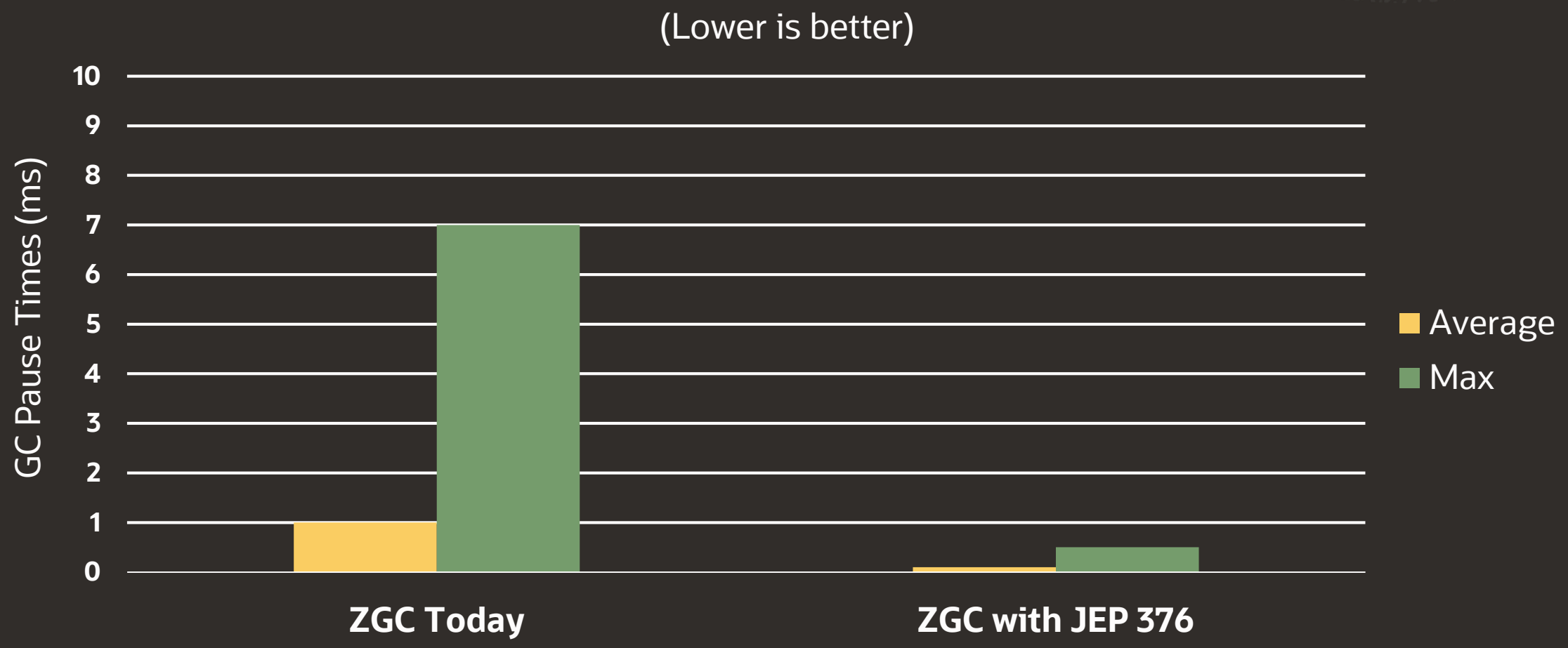
# SPECjbb2015 – Preliminary Results

**3T** Heap
**224** Hyper-threads (Intel)
**~2100** Java threads



# SPECjbb2015 – Preliminary Results

**3T** Heap
**224** Hyper-threads (Intel)
**~2100** Java threads



# Generational ZGC

# Generational ZGC

Young/Old Generation

- Exploit the fact that most objects are short-lived

Benefits

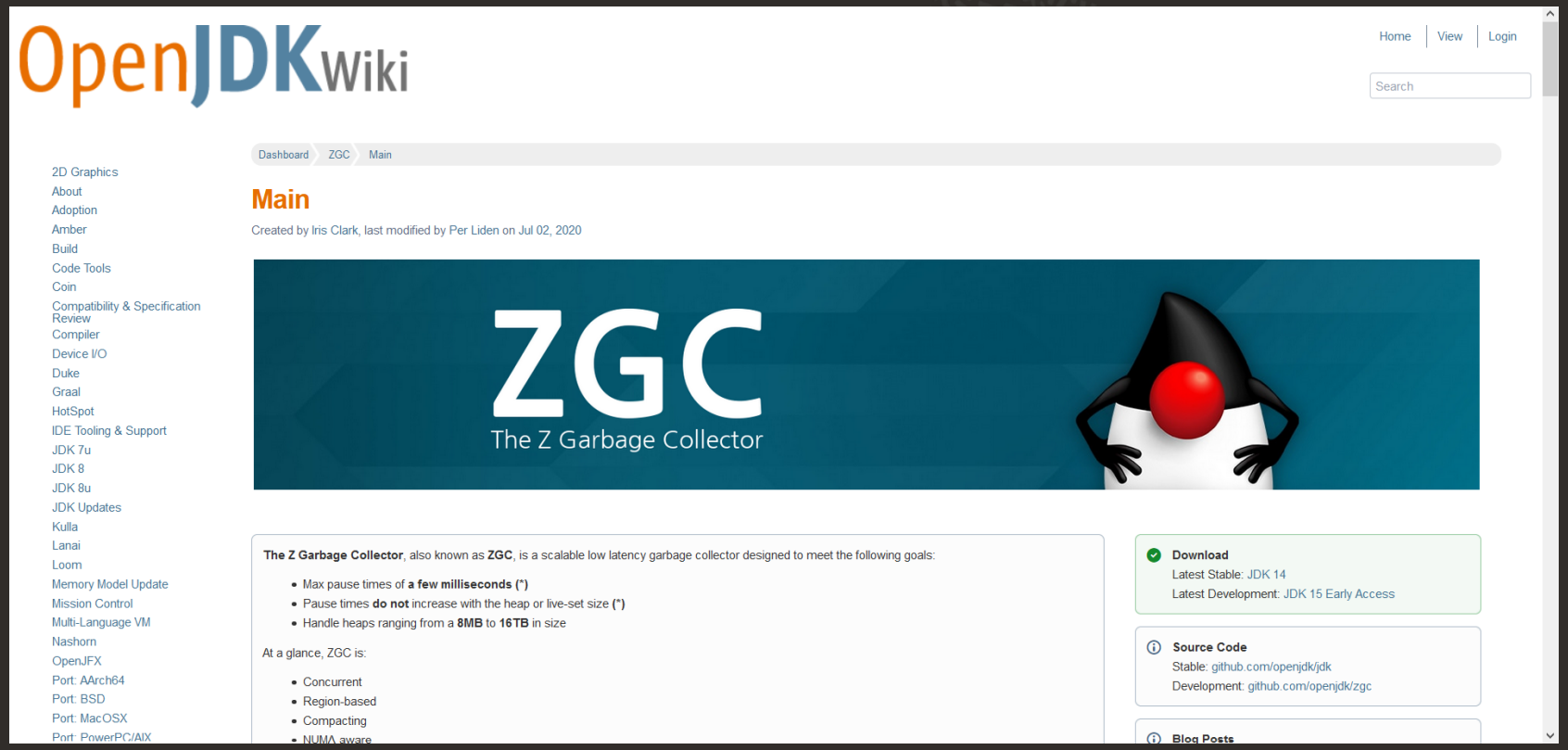
- Withstand higher allocation rates
- Lower heap headroom
- Lower CPU usage

Work in Progress

# More Information

# More Information

## ZGC Wiki




https://wiki.openjdk.java.net/display/zgc

# More Information
## GC Tuning Guide

Help Center

Search Account

Java / Java SE / 14

# HotSpot Virtual Machine Garbage Collection Tuning Guide

Table of Contents



## 9 The Z Garbage Collector

The Z Garbage Collector (ZGC) is a scalable low latency garbage collector. ZGC performs all expensive work concurrently, without stopping the execution of application threads for more than 10ms, which makes is suitable for applications which require low latency and/or use a very large heap (multi-terabytes).

The Z Garbage Collector is available as an experimental feature, and is enabled with the command-line options `-XX:+UnlockExperimentalVMOptions -XX:+UseZGC`.

### Setting the Heap Size

The most important tuning option for ZGC is setting the max heap size (`-Xmx`). Since ZGC is a concurrent collector a max heap size must be selected such that, 1) the heap can accommodate the live-set of your application, and 2) there is enough headroom in the heap to allow allocations to be serviced while the GC is running. How much headroom is needed very much depends on the allocation rate and the live-set size of the application. In general, the more memory you give to ZGC the better. But at the same time, wasting memory is undesirable, so it's all about finding a balance between memory usage and how often the GC needs to run.

### Setting Number of Concurrent GC Threads

The second tuning option one might want to look at is setting the number of concurrent GC threads (`-XX:ConcGCThreads`). ZGC has heuristics to automatically select this number. This heuristic usually works well but depending on the characteristics of the application this might need to be adjusted. This option essentially dictates how much CPU-time the GC should be given. Give it too much and the GC will steal too much CPU-time from the application. Give it too little, and the application might allocate garbage faster than the GC can collect it.

https://docs.oracle.com/en/java/javase/15/gctuning/

# More Information
My Blog

perliden

About Twitter GitHub

## ZGC | What's new in JDK 14

*23 Mar 2020*

Like clockwork, **JDK 14** was released on March 17, six months after **JDK 13**. From a **ZGC** point of view, this was a big release since we managed to complete several important milestones and improvements towards making it a fully production ready garbage collector. In total, **more than 80** enhancements and bug fixes were committed to ZGC itself. More if you also count "ZGC related" commits done in other parts of HotSpot.

In this post I'll highlight some of the more important and interesting ZGC enhancements. If you're using ZGC today, I highly recommend you to upgrade to JDK 14. In short, you'll get better performance, lower latency, new features, and improved stability. If you're on JDK 11 or later, upgrading to 14 should be straight forward.

For feedback and questions about ZGC, feel free to post to the **mailing list**.

### Windows and macOS support

**JEP 365** and **JEP 364** brought Windows and macOS support to ZGC. Support for these platforms has perhaps been the most common feature request we received. All commonly used platforms are now supported, and the complete list looks like this.

- **Linux/x86_64** (since JDK 11)
- **Linux/aarch64** (since JDK 13)

https://malloc.se

# Download JDK 15!

https://jdk.java.net/15

# Q&A

ORACLE