

ZGC: Paving the GC on-ramp



Linux geek-dive version

Stefan Johansson

Hotspot GC engineer

Oracle

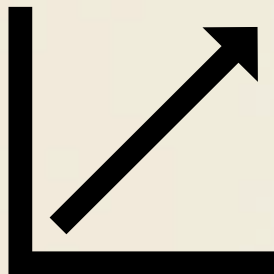


The goals of ZGC



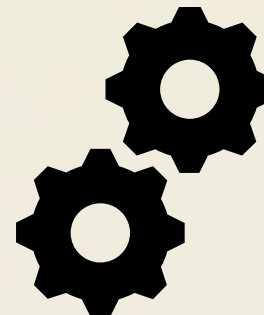
Low latency

Pause times below 1 ms



Scalability

Handle TB sized heaps



Auto-tuning

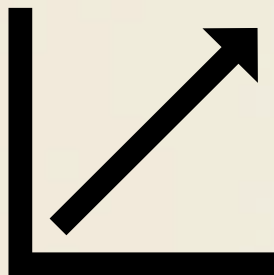
Minimal configuration required

The current status of ZGC



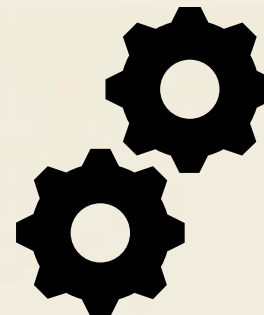
Low latency

- Pause only for synchronization
- Heavy work done concurrently



Scalability

Handle TB sized heaps



Auto-tuning

Minimal configuration required

The current status of ZGC



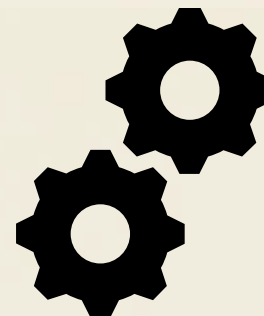
Low latency

- Pause only for synchronization
- Heavy work done concurrently



Scalability

- Support 16 TB heaps
- Pauses still short



Auto-tuning

Minimal configuration required

The current status of ZGC



Low latency

- Pause only for synchronization
- Heavy work done concurrently



Scalability

- Support 16 TB heaps
- Pauses still short



Auto-tuning

- Little configuration required
- Just set the heap size

Different design decisions



Why different

- Concurrent collector
- Different tradeoffs
- Two focus areas
 - Concurrency overhead
 - Heap memory overhead

Colored pointers

- Uses most of the 64 bits
 - 46 object address bits
 - 12 metadata bits
- Enables low GC overhead
- Enables eager memory reclamation

Discontiguous heap

- Reserve additional virtual address space
- Avoid fragmentation
- Always room to fit large allocation

Shared memory

- Needed for non-generational
 - Mapped heap at multiple addresses
 - Generational ZGC uses colorless roots
- Allow lazy unmapping
- Might move to anonymous memory

New techniques, additional configuration



Native Out of Memory

- Plenty of memory still left
- Caused by hitting a kernel limit
 - `vm.max_map_count`
- ZGC warns if the value is low
 - Suggests a value based on the heap size

Map count limit

- **Historic limit**
 - ELF format had restrictions
 - Core files could not handle more than 64K mappings
- **Default value is 64K**
- **Not only ZGC and the JVM**
- **Influence a change to the default**

Huge Pages

- **Explicit huge pages**
 - Need to decide on a heap size
 - Not good for auto configuration
- **Transparent huge pages**
 - For shared memory often requires configuration

Transparent Huge Pages configuration

- THP mode configurable

Transparent Huge Pages configuration

- THP mode configurable: always

Transparent Huge Pages configuration

- THP mode configurable: always, advise

Transparent Huge Pages configuration

- THP mode configurable: always, advise or never
 - Anonymous and shared memory configured separately
 - Different defaults: always vs never
 - Make sure to check:
`/sys/kernel/mm/transparent_hugepage/enabled`
`/sys/kernel/mm/transparent_hugepage/shmem_enabled`
- Unfair out of the box comparisons
- Explore MADV_COLLAPSE for madvise

Key take-aways



Basically no pauses

Can handle terabyte heaps

Is easy to use

ZGC
The Z Garbage Collector



Questions





Register today for JavaOne

oracle.com/javaone

March 18-20, 2025

Still taking questions...

