
Java™
ORACLE

JVM Optimizations in Java9

Vaibhav Choudhary (@vaibhav_c)
Java Platforms Team
Principal Member of Technical Staff
Bangalore JUG, Leader
https://blogs.oracle.com/vaibhav

Java Your Next (Cloud)

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, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda of the day …

1. Code Cache Segmentation - JEP 197
2. Garbage First (G1) Collector - JEP 248
3. Compact Strings - JEP 254
4. Unified JVM Logging - JEP 158
5. Compiler Control - JEP 165
6. References

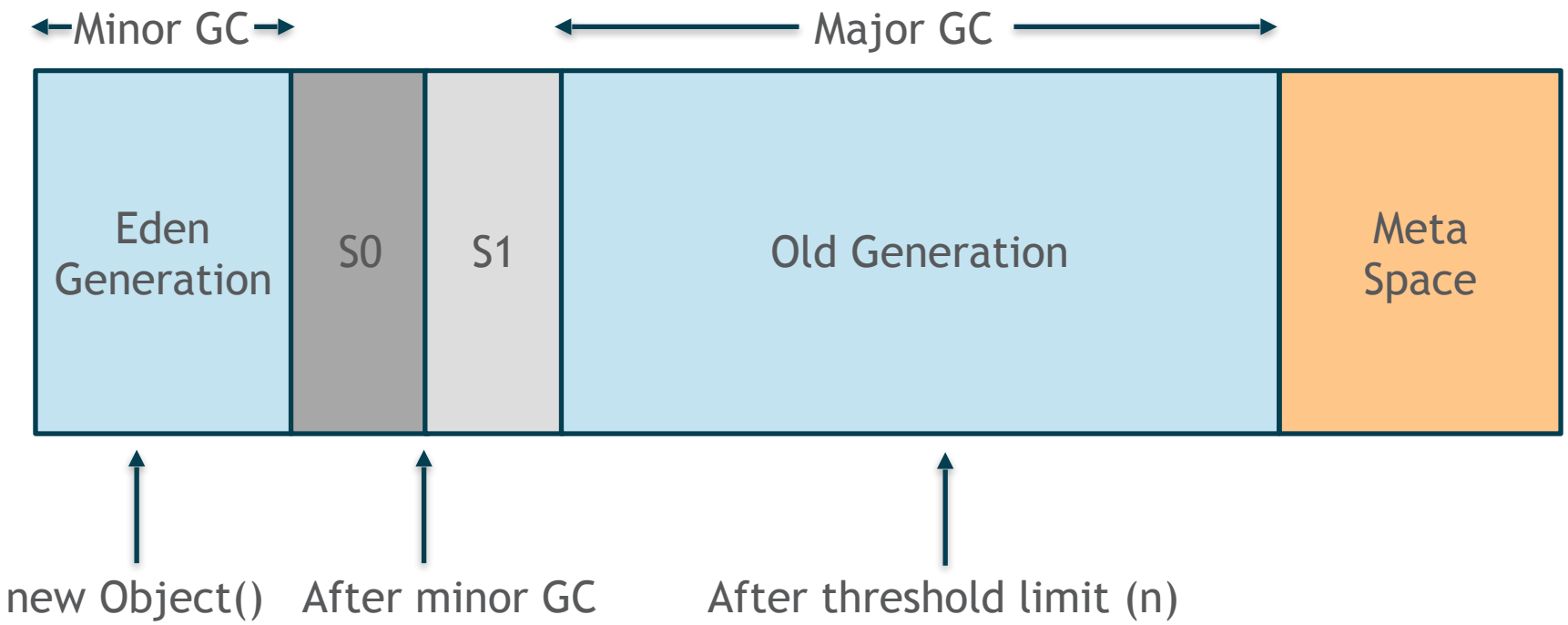
Code Cache Segmentation - JEP 197

- Lets understand a bit of Runtime Compilers
  - C1, C2
  - Tiered Compilation
- What if code cache is smaller than expected ?
- Pre-Java9 - Code cache was not segmented.
- Code cache segmentation
  - ‘non-nmethods’
  - ‘non-profiled nmethods’
  - ‘profiled nmethods’

Garbage First (G1) Collector - JEP 248

- Default collector in Java9.
- Long time replacement of CMS
- Compacting, Concurrent, Parallel, Stop the World.
- Can be used in Java7 and Java8 as well [use -XX:+UseG1GC].
- G1 Goals
  - Low latency
  - Predictable (Can’t be 100 percent)
  - Easy to use (Less parameter settings)

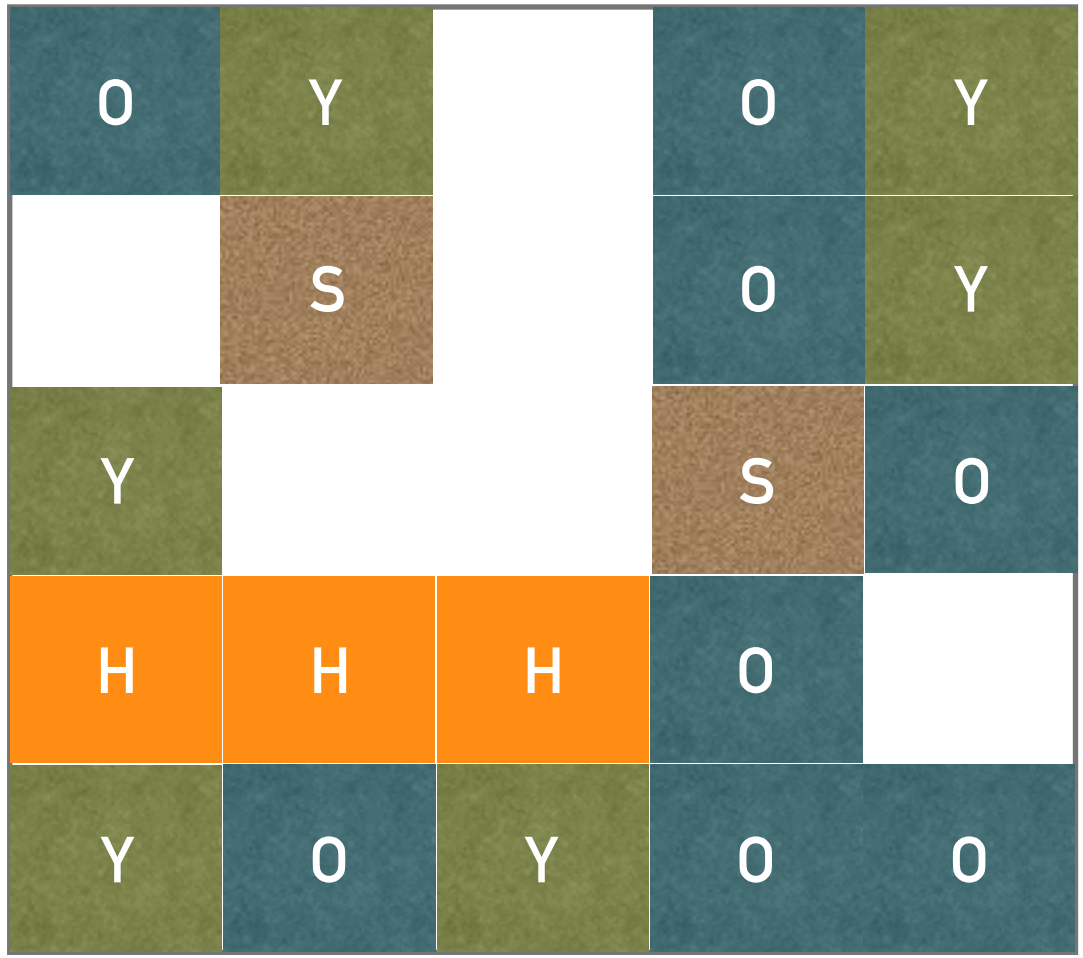
Java Heap Structure



Weak Generational Hypothesis :

- Most of the object die young.
- There are very few old to young reference.

Garbage First (G1) - Memory layout



- Memory is divided into small regions
- More than 2000 regions
- More flexible boundaries
- Use -XX:+G1HeapRegionSize
- Different regions :
  - Young
  - Survivor
  - Old
  - Humongous

Compact Strings - JEP 254

- In general, 20-25 percent of the java heap is String.
- In stead of char[], String is now byte[]
- “coder” field will decide UTF16 or Latin-1
- To disable the feature, use -XX:-CompactStrings
- Performance impact - Minimal

Unified JVM Logging - JEP 158

- Common Command Line option for all logging.
- Logging can use tags (compiler, gc, metaspace, …) and can use levels (error, warning, info, …)
- File rotations to log files.
- Print line-at-a-time
- Some examples :-
  - Xlog:gc*
  - Xlog:disable
  - Xlog:help

Compiler Control - JEP 165

- Fine grained and method context dependent control on JVM Compilers - C1 and C2.
- Ability to change the JVM compiler control at runtime.

```
[
  {
      // pattern to match against class+method+signature
      // leading and trailing wildcard (*) allowed
      match: “apa/Dingo.*",
      c2: {
          // control inlining of method
          // + force inline, - dont inline
          inline : [ "+java/util.*", "-com/sun.*"],
      }
      // applies to all compilers
      // + force inline, - dont inline
      inline : [ "+java/util.*", "-com/sun.*"],
      PrintInlining: true
  }
]
```