



Java 17 – Garbage Collector

Stephan Karrer

ZGC: A Scalable Low-Latency Garbage Collector (Java 15)

- in Java 11 als experimentell eingeführt, mit Java 15 produktionsreif
- wird aktiviert mit Kommandozeilenoption: `-XX:+UseZGC`
- Eigenschaften:
 - pausiert max. wenige (~2) Millisekunden
 - geeignet für Systeme mit minimaler Reaktionszeit, nicht für z.B. Batch Programme
 - GC Pausezeiten verlängern sich nicht mit wachsendem Heap
 - nur auf 64-bit Systemen nutzbar
 - verfügbar für Windows, MacOS, Linux/x86_64 und Linux/aarch64
- Links
 - <https://docs.oracle.com/en/java/javase/15/gctuning/z-garbage-collector.html>
 - <https://openjdk.org/jeps/377>
 - <https://www.baeldung.com/jvm-zgc-garbage-collector>
 - <https://inside.java/2023/01/25/zgc/>

Shenandoah: A Low-Pause-Time Garbage Collector (Java 15)

- Ziel: GC ohne Pausen, sondern nur im Hintergrund
- geht auf Kosten der laufenden Performance
- in Java 15 erste Version produktionsreif
- wird aktiviert mit Kommandozeilenoption: `-XX:+UseShenandoahGC`
- Eigenschaften:
 - pausiert max. 10 Millisekunden
 - geeignet für Systeme mit minimaler Reaktionszeit, nicht für z.B. Batch Programme
 - GC Pausezeiten verlängern sich nicht mit wachsendem Heap
 - unterstützt auch 32 bit Systeme
- Links
 - <https://openjdk.org/jeps/379>
 - <https://developers.redhat.com/articles/2021/11/02/how-choose-best-java-garbage-collector>

Epsilon: A No-Op Garbage Collector (Java 11)

- gibt niemals Speicher frei
- sobald der Heap voll ist, fährt die VM runter
- Sinn:
 - Performance Testing anderer GCs (Referenz)
 - künstlicher Speicherdruck für performante Programmierung
 - Mini Jobs
 - Garbage-freie Anwendungen
- Links
 - <https://openjdk.org/jeps/318>