GC Algorithms: Questions

Here is a list of questions you could consider to influence performance, latency, and memory utilization.

Application Characteristics:

- What is the nature of your application (web, desktop, mobile, etc.)?
- Does your application have specific latency or response time requirements?
- Is your application memory-intensive or CPU-intensive?
- Example:
  - Web applications: Parallel or G1 GC
  - Desktop applications: Serial or Parallel GC
  - Mobile applications: G1 GC

Heap Size and Memory Constraints:

- How much memory does your application require?
- Do you have memory constraints or limitations?
- Does your application have a small or large heap size?
- Example:
  - Small heap size: Serial GC
  - Large heap size: G1 GC

Hardware and Environment:

- How many processors or cores are available in your environment?
- Are you running your application on a single machine or a distributed system?
- What is the available memory capacity of your hardware?
- Example:
  - Single machine with limited cores: Serial GC
  - Multiprocessor or multicore hardware: Parallel or G1 GC

GC Pause Time Tolerance:

- What are the acceptable pause times for your application?
- Does your application require low pause times or can it tolerate longer pauses?
- Example:
  - Strict low pause time requirements: G1 GC
  - Tolerant to longer pauses: Parallel or CMS GC

Throughput and Performance Requirements:

- Does your application require high throughput (maximum work accomplished in a given time)?
- Are there specific performance goals or targets for your application?
- Example:
  - High throughput and maximum performance: Parallel GC
  - Balanced performance with low pause times: G1 GC

Testing and Profiling:

- Have you conducted performance testing and profiling of your application with different GC algorithms?
- Have you analyzed GC logs, memory usage patterns, and pause times?
- Example:
  - Performance testing and profiling are crucial to determine the best GC algorithm for your specific application. It is recommended to analyze GC logs, memory usage patterns, and pause times to make informed decisions.

Development and Deployment Environment:

- Which JVM implementation are you using (HotSpot, OpenJ9, GraalVM)?
- Does your JVM implementation offer additional GC algorithms or customizations?
- Example:
  - The choice of GC algorithm may depend on the JVM implementation. Each JVM (such as HotSpot, OpenJ9, GraalVM) may have different default GC algorithms and additional options to customize GC behavior.

Development and Maintenance Effort:

- Are you comfortable with configuring and tuning JVM parameters for different GC algorithms?
- Do you have the resources and expertise to monitor and maintain the chosen GC algorithm?
- Example:
  - Serial GC requires minimal configuration and tuning effort.
  - Parallel and G1 GC may require more expertise and effort for optimization and fine-tuning.

Considering these questions and factors will help you evaluate the trade-offs and choose the appropriate GC algorithm for your Java application. It's important to experiment, measure performance, and analyze the behavior of your application with different GC algorithms to find the best fit.

Semirah Dolan's LinkedIn page: www.linkedin.com/in/semirahdolan