

Task 1: Research and present a comparison of different garbage collection algorithms (Serial, Parallel, CMS, G1, ZGC) in Java.

Ans:

Serial Garbage Collector:

- **Type:** Serial Garbage Collector is a simple, single-threaded garbage collector.
- **Usage:** It's suitable for small applications or applications with low memory requirements.
- **Algorithm:** Uses a stop-the-world mechanism, where all application threads are paused while garbage collection occurs.
- **Advantages:** Simple and efficient for small applications. Low overhead due to its single-threaded nature.
- **Disadvantages:** Not suitable for large-scale applications or applications requiring low pause times.

Parallel Garbage Collector:

- **Type:** Also known as throughput collector, Parallel GC uses multiple threads for garbage collection.
- **Usage:** Suited for mid-sized to large applications where throughput is the primary concern.
- **Algorithm:** It employs a similar stop-the-world mechanism like Serial GC but utilizes multiple threads for garbage collection, thus reducing pause times.
- **Advantages:** Higher throughput compared to Serial GC due to parallelism. Suitable for applications with medium to large heap sizes.
- **Disadvantages:** Longer pause times compared to other collectors optimized for low-latency.

Concurrent Mark-Sweep (CMS) Garbage Collector:

- **Type:** CMS GC is designed to minimize pause times by running most of the GC work concurrently with the application.
- **Usage:** Aimed at applications where low-latency is crucial.
- **Algorithm:** It works by first marking all live objects in the heap concurrently with the application threads, then sweeping unused objects during short pauses.
- **Advantages:** Reduced pause times compared to the Parallel GC, making it suitable for applications requiring low-latency.
- **Disadvantages:** Increased CPU overhead due to concurrent execution. Not suitable for applications with large heap sizes or applications with frequent allocation and deallocation of objects.

G1 (Garbage-First) Garbage Collector:

- **Type:** G1 GC is designed for large heaps with minimal pause times.
- **Usage:** Best suited for applications requiring low-latency and handling large heap sizes.
- **Algorithm:** It divides the heap into smaller regions and uses a mix of parallel, concurrent, and incremental techniques for garbage collection.

- **Advantages:** Predictable pause times, even for large heap sizes. Efficient handling of both short-lived and long-lived objects.
- **Disadvantages:** G1 GC may exhibit higher CPU overhead compared to other collectors due to its complex algorithms. Not recommended for small heap sizes.

ZGC (Z Garbage Collector):

- **Type:** ZGC is a low-latency garbage collector introduced in JDK 11.
- **Usage:** Aimed at applications requiring very low pause times, typically below 10ms.
- **Algorithm:** ZGC uses a concurrent approach similar to CMS but with additional optimizations for minimizing pause times.
- **Advantages:** Extremely low pause times even for very large heaps. It's designed to scale with the heap size and the number of available CPU cores.
- **Disadvantages:** ZGC may incur higher CPU overhead compared to other collectors due to its emphasis on low-latency. Limited platform support compared to other collectors.