7 The Mostly Concurrent Collectors

Java Hotspot VM has two mostly concurrent collectors in JDK 8:

- Concurrent Mark Sweep (CMS) Collector: This collector is for applications that prefer shorter garbage collection pauses and can afford to share processor resources with the garbage collection.
- Garbage-First Garbage Collector: This server-style collector is for multiprocessor machines with large memories. It meets garbage collection pause time goals with high probability while achieving high throughput.

Overhead of Concurrency

The mostly concurrent collector trades processor resources (which would otherwise be available to the application) for shorter major collection pause times. The most visible overhead is the use of one or more processors during the concurrent parts of the collection. On an *N* processor system, the concurrent part of the collection will use *K*/*N* of the available processors, where 1<=*K*<=ceiling{*N*/4}. (Note that the precise choice of and bounds on *K* are subject to change.) In addition to the use of processors during concurrent phases, additional overhead is incurred to enable concurrency. Thus while garbage collection pauses are typically much shorter with the concurrent collector, application throughput also tends to be slightly lower than with the other collectors.

On a machine with more than one processing core, processors are available for application threads during the concurrent part of the collection, so the concurrent garbage collector thread does not "pause" the application. This usually results in shorter pauses, but again fewer processor resources are available to the application and some slowdown should be expected, especially if the application uses all of the processing cores maximally. As *N* increases, the reduction in processor resources due to concurrent garbage collection becomes smaller, and the benefit from concurrent collection increases. The section Concurrent Mode Failure in Concurrent Mark Sweep (CMS) Collector discusses potential limits to such scaling.

Because at least one processor is used for garbage collection during the concurrent phases, the concurrent collectors do not normally provide any benefit on a uniprocessor (single-core) machine. However, there is a separate mode available for CMS (not G1) that can achieve low pauses on systems with only one or two processors; see Incremental Mode in Concurrent Mark Sweep (CMS) Collector for details. This feature is being deprecated in Java SE 8 and may be removed in a later major release.

Additional References

The Garbage-First Garbage Collector:

`http://www.oracle.com/technetwork/java/javase/tech/g1-intro-jsp-135488.html`

Garbage-First Garbage Collector Tuning:

`http://www.oracle.com/technetwork/articles/java/g1gc-1984535.html`