Запуск проекта проводился на следующей сборке:

*ОС: Windows 10 Домашняя. Вер. 10.0.19045*

*Процессор: Intel Core i5-10400F CPU 2.90GHz, 2901 МГц, 6 ядер*

*Оперативная память: 16 ГБ*

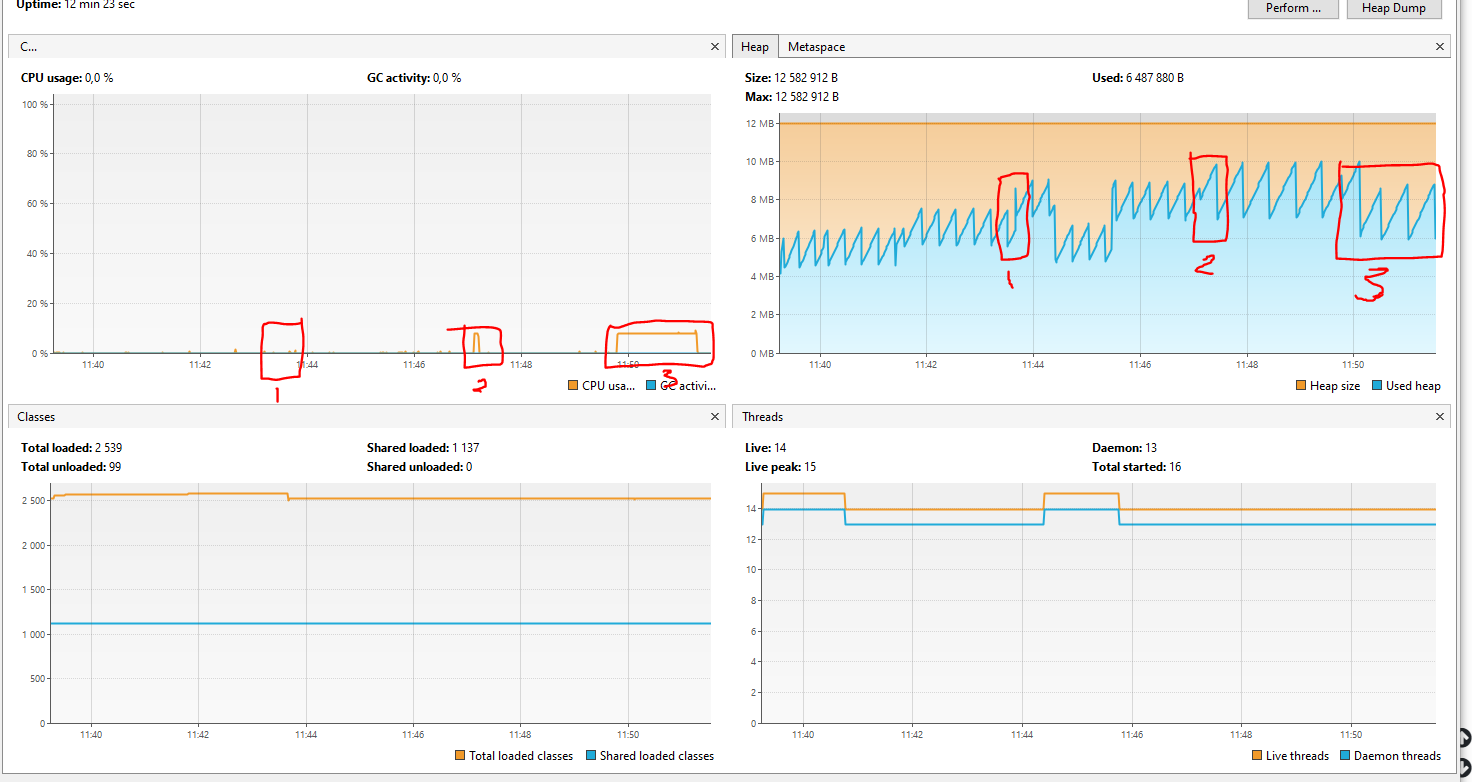
*Мониторинг: VisualVM*

*Максимальный размер heap: 12MB*

**Тестирование ParallelGC**

В состоянии покоя чистка heap происходит часто и скачкообразно.

1. MergeSort: 9мс. Процессор при данной сортировки был нагружен минимально (до 0.1%), однако резко возросло использование памяти до 8,5MB.
2. InsertSort: 5сек. В данной сортировке потребление CPU увеличилось до 8% и длилось на протяжении всего времени работы метода. Скачок использования памяти наблюдается, но уже не в такой значительной степени как при MergeSort.
3. BubbleSort: 1мин. 30сек. CPU был нагружен до 8% все время работы метода, однако на память существенного влияния не оказал



**Тестирование G1GC**

В состоянии покоя чистка heap происходит более плавно, чем при ParallelGC.

1. MergeSort: 8мс. Нагрузка процессора как и при ParallelGC. Наблюдается такой же резкий скачек использования памяти с 6MB до 8MB.
2. InsertSort: 5сек. Нагрузка процессора как и при ParallelGC. В это время наблюдается скачок, однако он такой же как и при состоянии покоя и на графике неотличим.
3. BubbleSort: 1мин. 36сек. Нагрузка процессора как и при ParallelGC. При работе данного метода можно увидеть увеличение средней планки потребления памяти с 5MB до 7MB и резкий скачок до 9MB и спад к 7MB. После этого, до завершения работы метода, heap не очищался.



**Тестирование G1GC**

*Максимальный размер heap увеличен до 24MB.*

В состоянии покоя чистка heap происходит еще более плавно и планомерно, чем при G1GC.

1. MergeSort: 0.8мс. Нагрузка процессора как и при ParallelGC. Наблюдается скачек использования памяти в середине времени работы метода.
2. InsertSort: 4,5сек. Нагрузка процессора как и при ParallelGC. Такой же скачек в середине работы метода, как и в MergeSort.
3. BubbleSort: 1мин. 31сек. Нагрузка процессора как и при ParallelGC, но перед завершением есть скачок до 9% CPU. Метод оказал влияние на heap только в середине работы. После этого, до завершения работы метода, heap не очищался.

