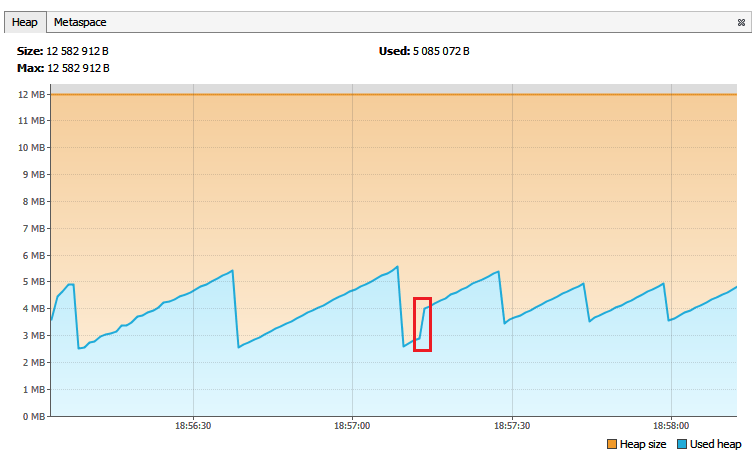
**1. Профилирование сборщика мусора Parallel**

Установил опции для класса Menu и запустил из main. Далее запускал по пунктам из консоли

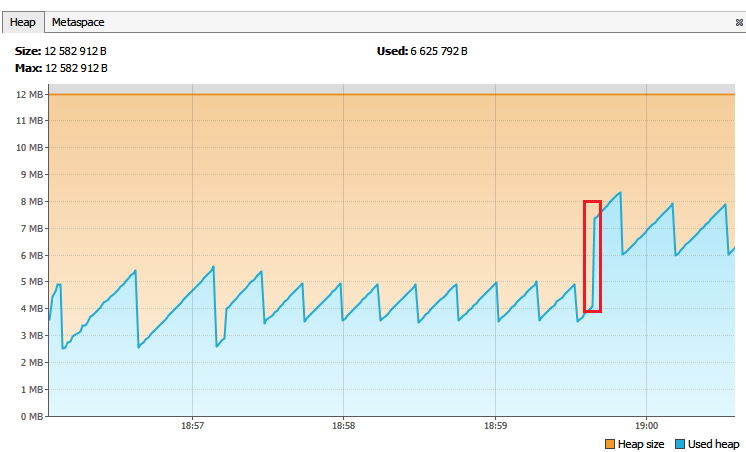
-XX:+UseParallelGC -Xmx12m -Xms12m -Xlog:gc:log.txt:time,level,tags

Использовал для визуализации профилировщик Visual VM для pid класса ru.job4j.gc.prof.Menu

**1) Создание массива из 250000 элементов**



На графике видно, что размер массиваувеличился на ~ 1 Мб

**2) Сортировка слиянием**

На графике видно, что размер массива скачком увеличился на ~ 3,4 Мб

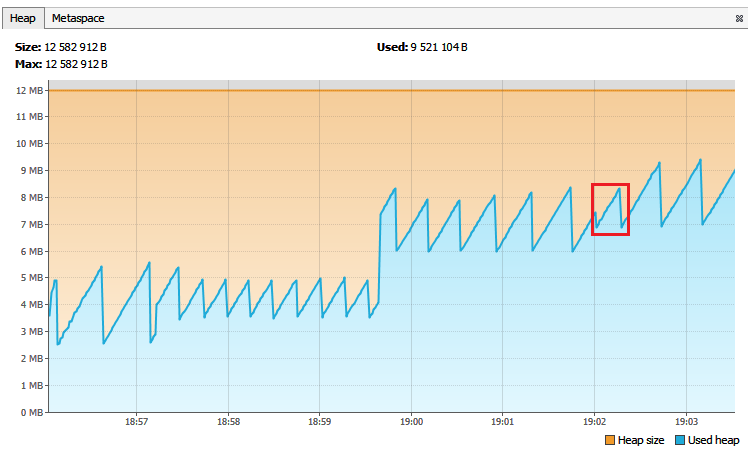
MergeSort Начало сортировки - 18:59:38.852423600

MergeSort Конец сортировки - 18:59:38.931428100

Вся сортировка заняла 79 миллисекунду

Дальше было несколько малых сборок

**3) Сортировка методом вставки**

****

Хип вырос на размер массива ~ 1,3 Мб

InsertSort Начало сортировки - 19:02:01.011554700

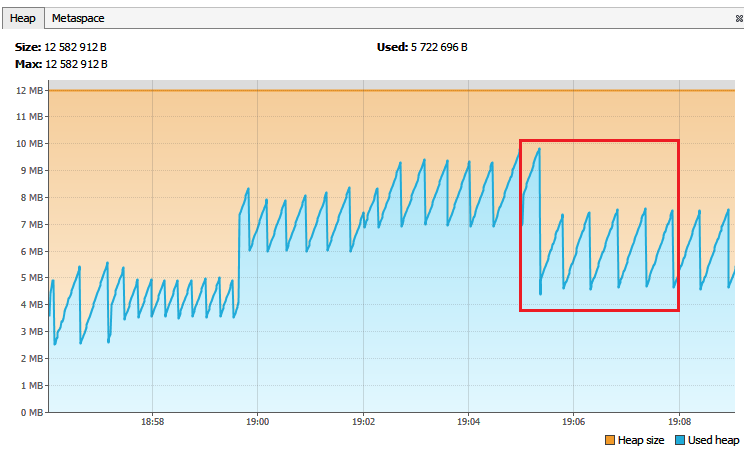
InsertSort Конец сортировки - 19:02:35.394521200

Вся сортировка заняла 34 секунды

Дальше было несколько малых сборок

**4) Сортировка пузырьком**

Начинаем сортировку пузырьком. В хип добавился клон массива и затем происходит полная сборка



После добавления клона массива хип скачком уменьшился на более чем 5 Мб

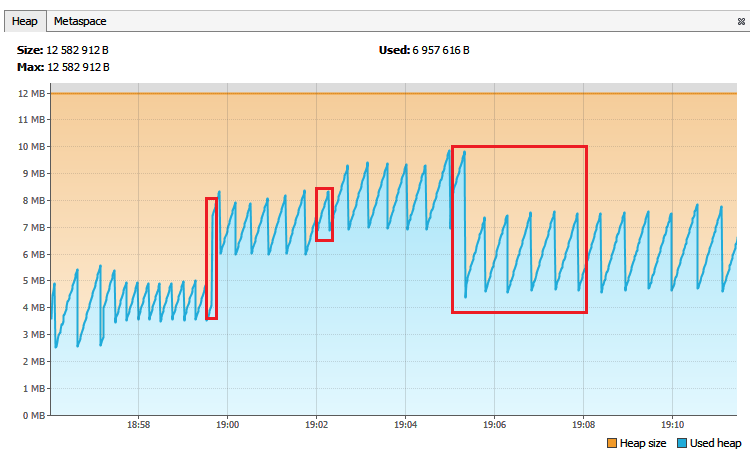
И далее в процессе сортировки было еще несколько малых сборок

BubbleSort Начало сортировки - 19:05:01.878899700

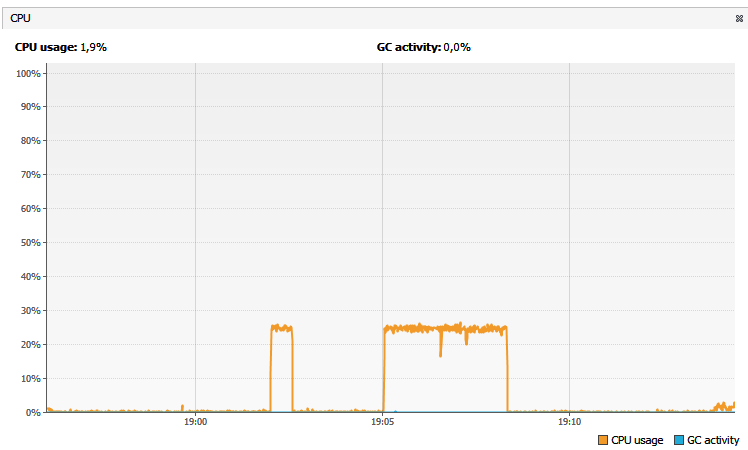
BubbleSort Конец сортировки - 19:08:19.097179900

Вся сортировка заняла 3 мин 18 сек

**5) Общий график, на котором выделены сортировки**

****

**6) График загрузки процессора на каждой сборке**



Сортировка пузырьком была самая длительная и потребовала больше всего ресурсов процессора.

Сортировка слиянием заняла меньше всего времени и ресурсов процессора.

**7) Лог сборщика мусора**

[2024-10-28T18:59:38.862+0300][info][gc] GC(14) Pause Young (Allocation Failure) 5M->4M(10M) 0.647ms  
[2024-10-28T18:59:38.867+0300][info][gc] GC(15) Pause Young (Allocation Failure) 6M->4M(10M) 0.421ms  
[2024-10-28T18:59:38.871+0300][info][gc] GC(16) Pause Young (Allocation Failure) 6M->4M(10M) 0.590ms  
[2024-10-28T18:59:38.875+0300][info][gc] GC(17) Pause Young (Allocation Failure) 6M->4M(10M) 0.408ms  
[2024-10-28T18:59:38.879+0300][info][gc] GC(18) Pause Young (Allocation Failure) 6M->4M(10M) 0.348ms  
[2024-10-28T18:59:38.883+0300][info][gc] GC(19) Pause Young (Allocation Failure) 6M->4M(10M) 0.453ms  
[2024-10-28T18:59:38.888+0300][info][gc] GC(20) Pause Young (Allocation Failure) 6M->5M(10M) 0.840ms  
[2024-10-28T18:59:38.894+0300][info][gc] GC(21) Pause Young (Allocation Failure) 6M->5M(11M) 1.426ms  
[2024-10-28T18:59:38.899+0300][info][gc] GC(22) Pause Young (Allocation Failure) 6M->5M(10M) 0.475ms  
[2024-10-28T18:59:38.904+0300][info][gc] GC(23) Pause Young (Allocation Failure) 7M->5M(11M) 0.600ms  
[2024-10-28T18:59:38.909+0300][info][gc] GC(24) Pause Young (Allocation Failure) 7M->5M(11M) 0.377ms  
[2024-10-28T18:59:38.914+0300][info][gc] GC(25) Pause Young (Allocation Failure) 7M->5M(11M) 0.336ms  
[2024-10-28T18:59:38.920+0300][info][gc] GC(26) Pause Young (Allocation Failure) 7M->5M(11M) 0.418ms  
[2024-10-28T18:59:38.925+0300][info][gc] GC(27) Pause Young (Allocation Failure) 7M->6M(11M) 0.361ms  
[2024-10-28T18:59:38.928+0300][info][gc] GC(28) Pause Young (Allocation Failure) 7M->6M(11M) 0.380ms  
[2024-10-28T18:59:49.663+0300][info][gc] GC(29) Pause Young (Allocation Failure) 8M->5M(11M) 0.428ms  
[2024-10-28T19:00:10.685+0300][info][gc] GC(30) Pause Young (Allocation Failure) 7M->5M(11M) 0.766ms

. . .

[2024-10-28T19:02:01.013+0300][info][gc] GC(35) Pause Young (Allocation Failure) 7M->5M(11M) 0.942ms  
[2024-10-28T19:02:17.515+0300][info][gc] GC(36) Pause Young (Allocation Failure) 8M->6M(11M) 0.719ms  
[2024-10-28T19:02:43.535+0300][info][gc] GC(37) Pause Young (Allocation Failure) 9M->6M(11M) 0.971ms

. . .

[2024-10-28T19:05:21.531+0300][info][gc] GC(43) Pause Young (Allocation Failure) 9M->7M(11M) 0.779ms  
[2024-10-28T19:05:21.543+0300][info][gc] GC(44) Pause Full (Ergonomics) 7M->4M(11M) 11.591ms  
[2024-10-28T19:05:47.996+0300][info][gc] GC(45) Pause Young (Allocation Failure) 7M->4M(11M) 0.628ms  
[2024-10-28T19:06:18.534+0300][info][gc] GC(46) Pause Young (Allocation Failure) 7M->4M(11M) 0.544ms  
[2024-10-28T19:06:50.090+0300][info][gc] GC(47) Pause Young (Allocation Failure) 7M->4M(11M) 0.369ms  
[2024-10-28T19:07:22.118+0300][info][gc] GC(48) Pause Young (Allocation Failure) 7M->4M(11M) 0.289ms  
[2024-10-28T19:07:53.145+0300][info][gc] GC(49) Pause Young (Allocation Failure) 7M->4M(11M) 0.419ms  
[2024-10-28T19:08:24.603+0300][info][gc] GC(50) Pause Young (Allocation Failure) 7M->4M(11M) 0.627ms

Один раз произошла полная сборка мусора: Pause Full - 7M->4M(11M) 11.591ms

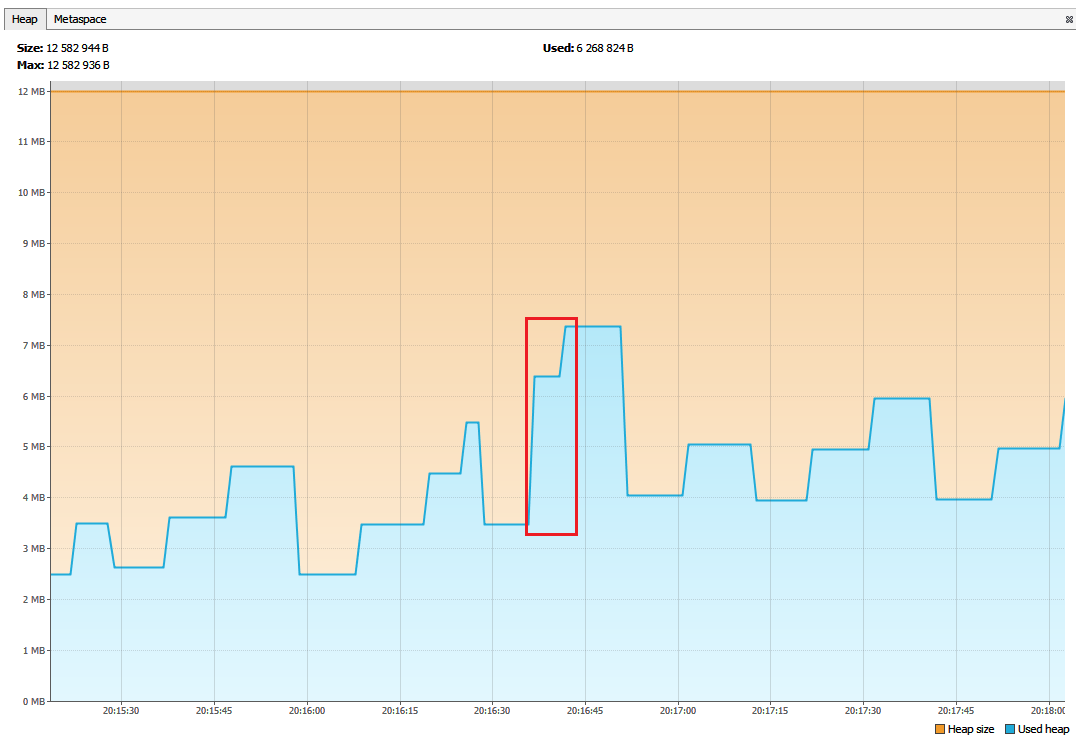
**2. Профилирование сборщика мусора G1**

Установил опции для класса Menu и запустил из main. Далее запускал по пунктам из консоли

-XX:+UseG1GC -Xmx12m -Xms12m -Xlog:gc:log.txt:time,level,tags

Использовал для визуализации профилировщик Visual VM для pid класса ru.job4j.gc.prof.Menu

**1) Сортировка слиянием**

****

На графике видно, что размер массива на период сортировки увеличился на ~ 3 Мб и дальше еще на ~ 1 Мб

MergeSort Начало сортировки - 20:16:36.558484300

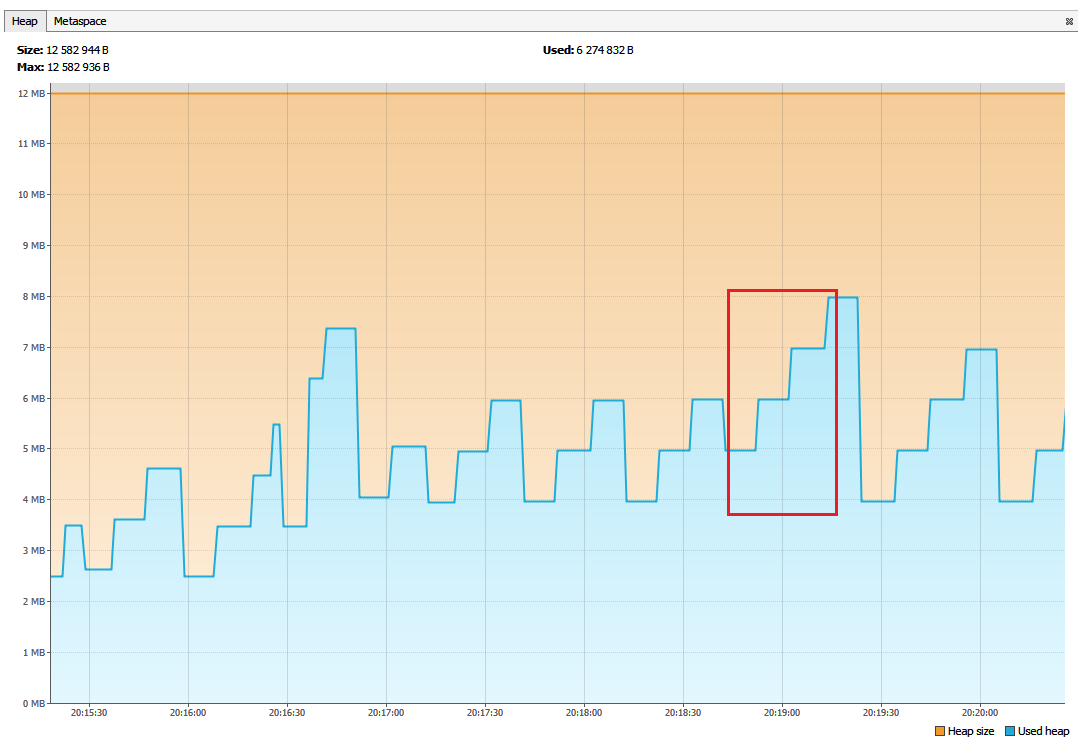
MergeSort Конец сортировки - 20:16:36.661490200

Вся сортировка заняла 103 миллисекунду

Дальше был еще скачек хипа на ~ 1 Мб, но не пилообразно, как для *Parallel GC,* а ступенчато с горизонтальным плато*,* видимо за счет того, что часть heap чиститься наряду с работой приложения.

Дальше было несколько малых сборок, но их количество меньше, чем для *Parallel GC*

**3) Сортировка методом вставки**



Хип вырос на размер массива ~ 1 Мб и далее еще на ~ 2 Мб

InsertSort Начало сортировки - 20:18:42.037661300

InsertSort Конец сортировки - 20:19:17.689700400

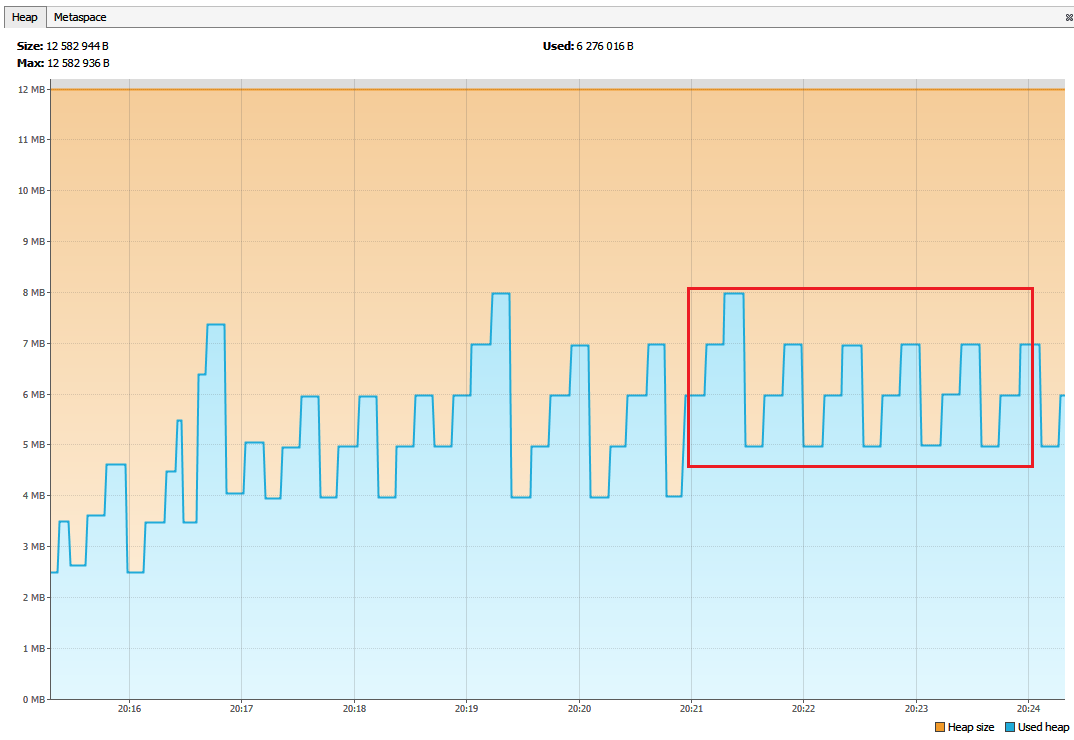
Вся сортировка заняла 35 секунд

За время сортировки было 3 скачка хипа по ~ 1 Мб каждый, т.е. увеличение хипа было также ступенчато, а не пилообразно, как для *Parallel GC.*

Дальше было несколько малых сборок и их количество также меньше, чем для *Parallel GC*

**4) Сортировка пузырьком**

Начинаем сортировку пузырьком. В хип добавился клон массива, но полной сборки, как для *Parallel GC* не произошло. На протяжении сортировки было только несколько малых сборок



После добавления клона массива хип сначала возрос на ~ 1 Мб и далее скачком уменьшился на ~ 3 Мб

И далее в процессе сортировки было еще несколько малых сборок

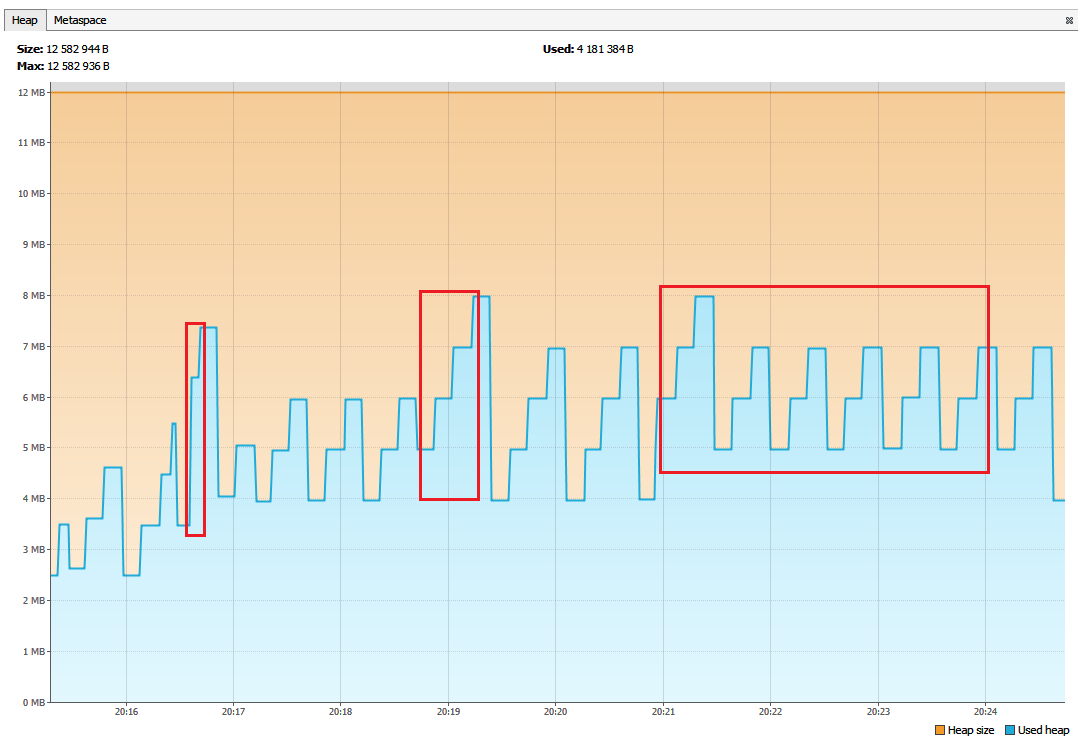
BubbleSort Начало сортировки - 20:20:55.728307900

BubbleSort Конец сортировки - 20:24:09.199373800

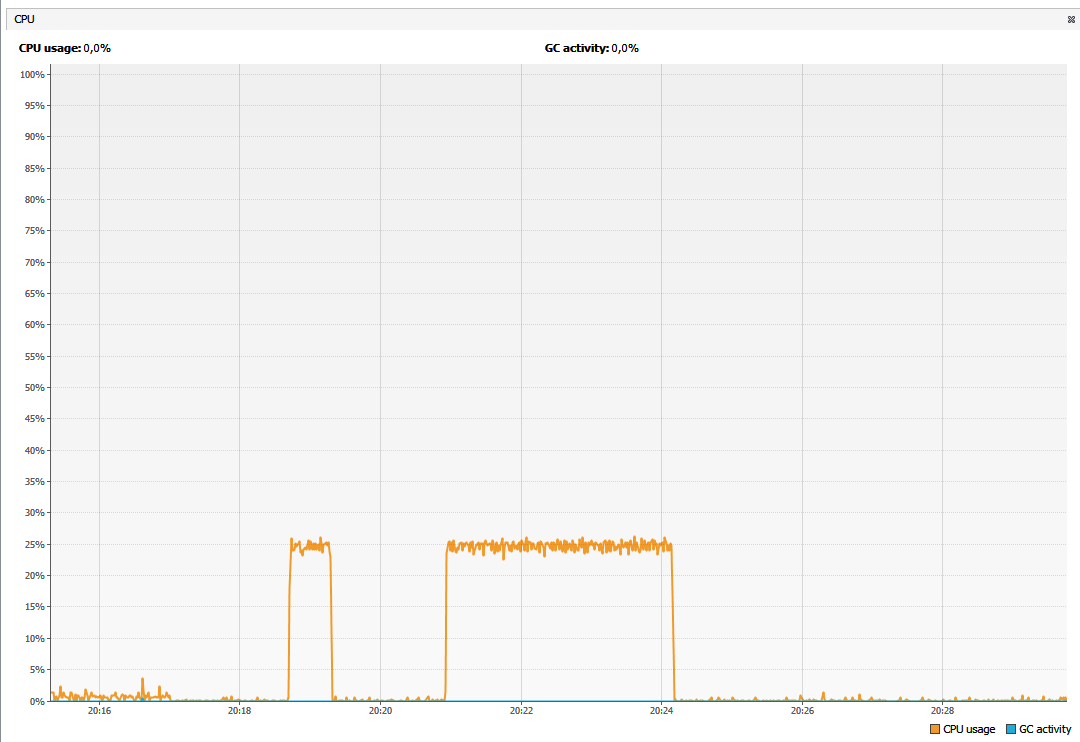
Вся сортировка заняла 3 мин 14 сек

Увеличение хипа было также ступенчато, а не пилообразно, как для *Parallel GC.*

**5) Общий график, на котором выделены сортировки**



**6) График загрузки процессора на каждой сборке**

****

Сортировка пузырьком была самая длительная и потребовала больше всего ресурсов процессора.

Сортировка слиянием заняла меньше всего времени и ресурсов процессора.

**7) Лог сборщика мусора**

[2024-11-10T20:16:36.581+0300][info][gc] GC(6) Pause Young (Normal) (G1 Evacuation Pause) 7M->4M(12M) 2.851ms  
[2024-11-10T20:16:36.589+0300][info][gc] GC(7) Pause Young (Normal) (G1 Evacuation Pause) 6M->4M(12M) 2.489ms  
[2024-11-10T20:16:36.597+0300][info][gc] GC(8) Pause Young (Normal) (G1 Evacuation Pause) 6M->4M(12M) 1.966ms  
[2024-11-10T20:16:36.604+0300][info][gc] GC(9) Pause Young (Normal) (G1 Evacuation Pause) 6M->4M(12M) 1.248ms  
[2024-11-10T20:16:36.611+0300][info][gc] GC(10) Pause Young (Normal) (G1 Evacuation Pause) 6M->4M(12M) 1.087ms  
[2024-11-10T20:16:36.617+0300][info][gc] GC(11) Pause Young (Normal) (G1 Evacuation Pause) 6M->4M(12M) 1.022ms  
[2024-11-10T20:16:36.623+0300][info][gc] GC(12) Pause Young (Normal) (G1 Evacuation Pause) 6M->5M(12M) 0.770ms  
[2024-11-10T20:16:36.628+0300][info][gc] GC(13) Pause Young (Normal) (G1 Evacuation Pause) 7M->5M(12M) 0.703ms  
[2024-11-10T20:16:36.633+0300][info][gc] GC(14) Pause Young (Normal) (G1 Evacuation Pause) 7M->5M(12M) 0.461ms  
[2024-11-10T20:16:36.638+0300][info][gc] GC(15) Pause Young (Normal) (G1 Evacuation Pause) 7M->5M(12M) 0.586ms  
[2024-11-10T20:16:36.653+0300][info][gc] GC(16) Pause Young (Normal) (G1 Evacuation Pause) 8M->5M(12M) 7.181ms  
[2024-11-10T20:16:36.656+0300][info][gc] GC(17) Pause Young (Concurrent Start) (G1 Humongous Allocation) 6M->5M(12M) 0.803ms  
[2024-11-10T20:16:36.656+0300][info][gc] GC(18) Concurrent Cycle  
[2024-11-10T20:16:36.667+0300][info][gc] GC(18) Pause Remark 6M->6M(12M) 2.435ms  
[2024-11-10T20:16:36.671+0300][info][gc] GC(18) Pause Cleanup 6M->6M(12M) 0.023ms  
[2024-11-10T20:16:36.671+0300][info][gc] GC(18) Concurrent Cycle 14.731ms  
[2024-11-10T20:16:51.735+0300][info][gc] GC(19) Pause Young (Normal) (G1 Evacuation Pause) 8M->4M(12M) 2.098ms

. . .

[2024-11-10T20:18:41.899+0300][info][gc] GC(23) Pause Young (Normal) (G1 Evacuation Pause) 6M->3M(12M) 2.136ms  
[2024-11-10T20:19:23.801+0300][info][gc] GC(24) Pause Young (Normal) (G1 Evacuation Pause) 8M->3M(12M) 1.445ms

. . .  
[2024-11-10T20:21:28.046+0300][info][gc] GC(27) Pause Young (Normal) (G1 Evacuation Pause) 8M->4M(12M) 0.724ms  
[2024-11-10T20:21:59.803+0300][info][gc] GC(28) Pause Young (Normal) (G1 Evacuation Pause) 7M->4M(12M) 0.708ms  
[2024-11-10T20:22:31.106+0300][info][gc] GC(29) Pause Young (Normal) (G1 Evacuation Pause) 7M->4M(12M) 1.088ms  
[2024-11-10T20:23:02.136+0300][info][gc] GC(30) Pause Young (Normal) (G1 Evacuation Pause) 7M->5M(12M) 0.838ms  
[2024-11-10T20:23:34.164+0300][info][gc] GC(31) Pause Young (Normal) (G1 Evacuation Pause) 8M->4M(12M) 0.889ms  
[2024-11-10T20:24:06.193+0300][info][gc] GC(32) Pause Young (Normal) (G1 Evacuation Pause) 7M->4M(12M) 0.649ms

Полная сборка мусора за время всех сортировок так и не произошла

**3. Профилирование сборщика мусора ZGC  - не запускается, выдает ошибку**

Установил опции для класса Menu и запустил из main

-XX:+UnlockExperimentalVMOptions -XX:+UseZGC –Xmx512m –Xms512m -Xlog:gc:log.txt:time,level,tags

**ZGC не запускается, выдает ошибку:**

Error occurred during initialization of VM

Option -XX:+UseZGC not supported