250000 элементов

**Parallel**

MergeSort Начало сортировки - 22:23:46.353803900

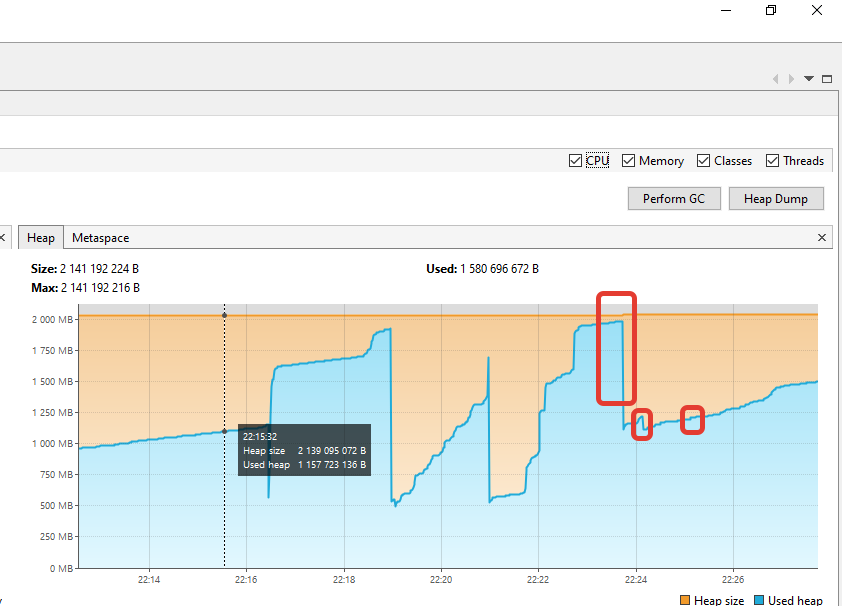
MergeSort Конец сортировки - 22:23:46.401681300

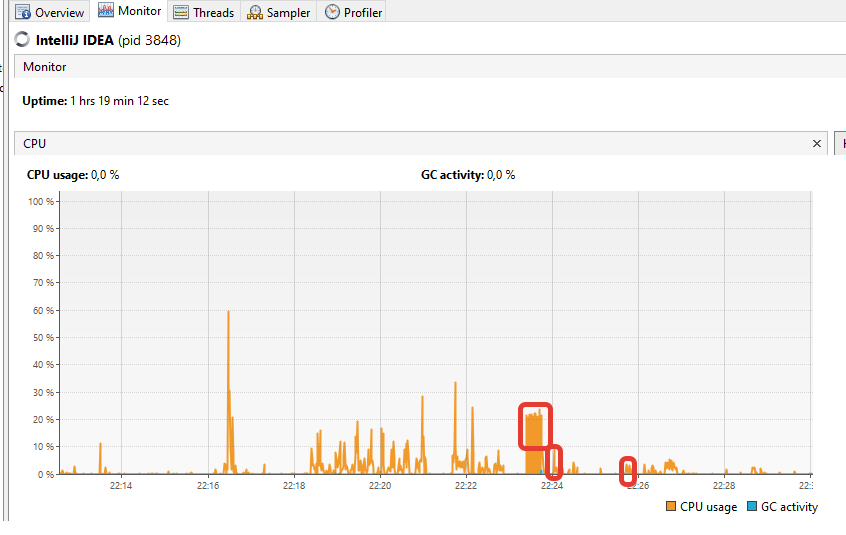
BubbleSort Начало сортировки - 22:24:04.701621200

BubbleSort Конец сортировки - 22:25:39.575289200

InsertSort Начало сортировки - 22:25:46.701223500

InsertSort Конец сортировки - 22:25:51.759719600





14 сборок мусора, самая первая 3 мс.

Последняя 0.3 мс.

**G1**

250000

MergeSort Начало сортировки - 22:33:32.042826800

MergeSort Конец сортировки - 22:33:32.105753900

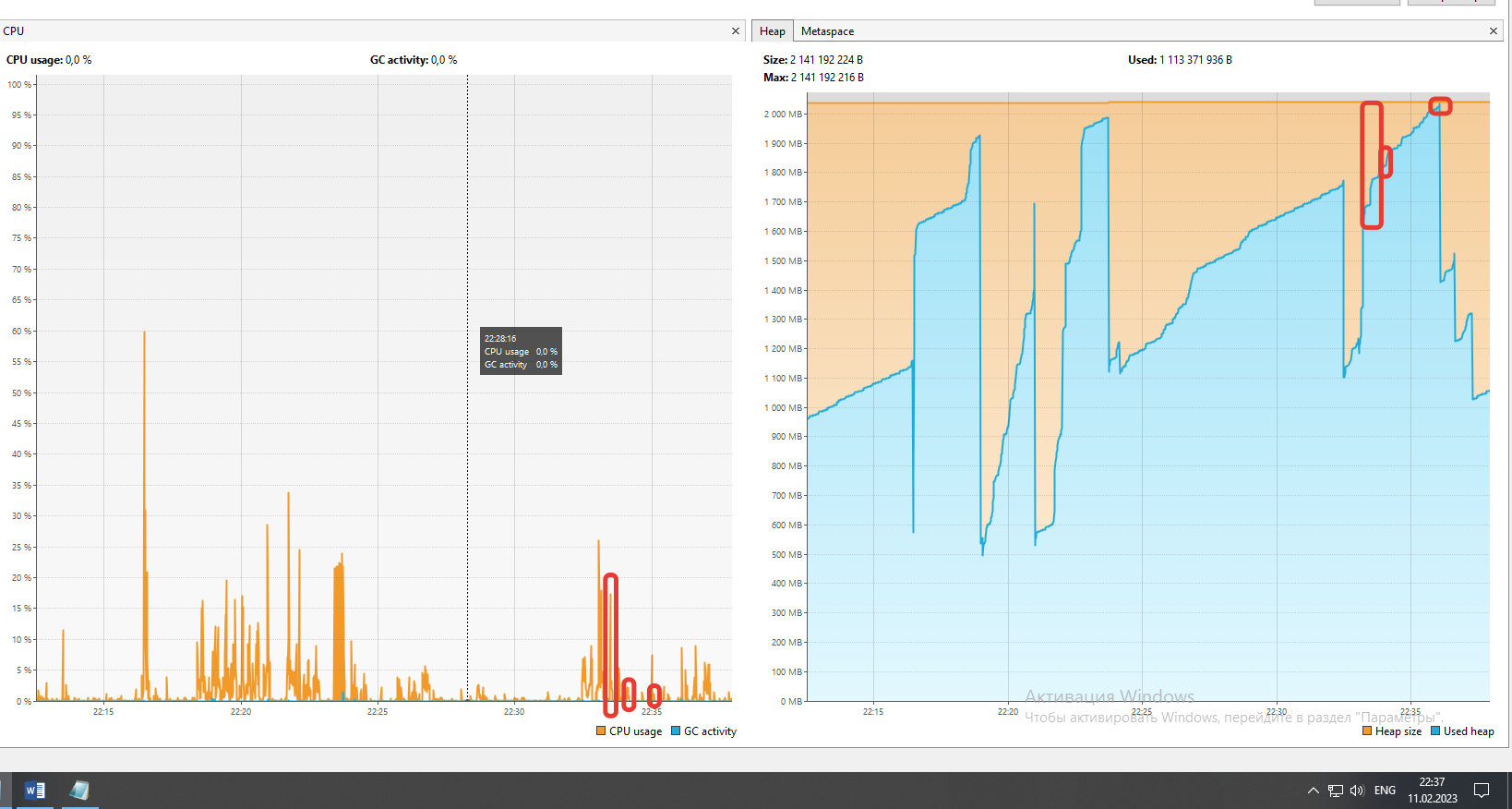
BubbleSort Начало сортировки - 22:34:08.561258200

BubbleSort Конец сортировки - 22:35:42.403362

InsertSort Начало сортировки - 22:36:16.252765400

InsertSort Конец сортировки - 22:36:21.301291200

22 сборки мусора. Первая 3 мс. Последняя 0, 1 мс.



**Z**

MergeSort Начало сортировки - 22:45:27.218963100

MergeSort Конец сортировки - 22:45:27.275187900

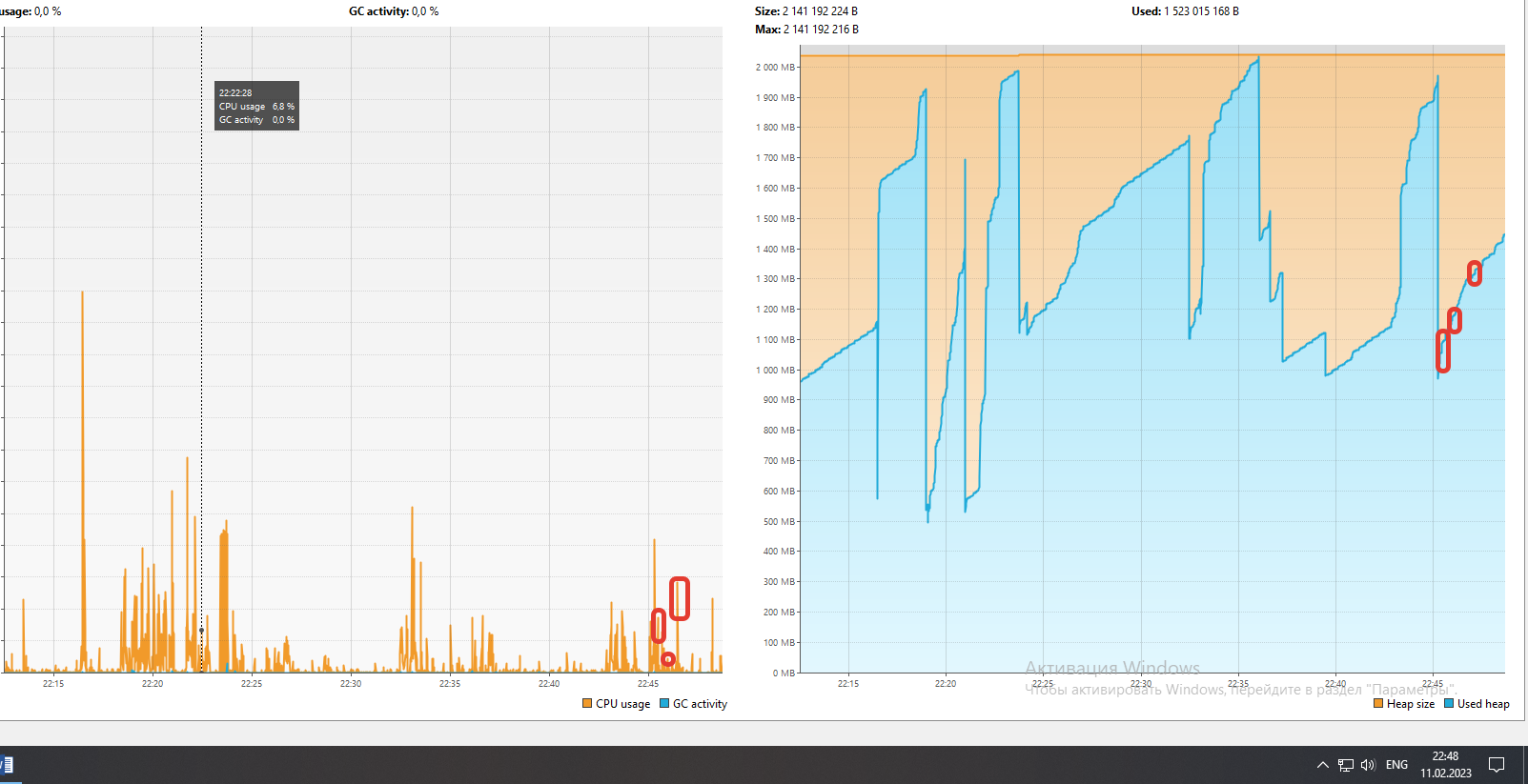
BubbleSort Начало сортировки - 22:45:58.572042500

BubbleSort Конец сортировки - 22:47:35.795005

InsertSort Начало сортировки - 22:48:10.671038700

InsertSort Конец сортировки - 22:48:15.715571600

3 сборки мусора. 3 – 6 – 6 сек.



**Выводы.**

Судя по общему графику, на моем пк (I5-8400, 16 GB RAM, HDD 1TB) технологии сборки мусора влияли именно на процессор, т.е. задействовали многоточность (6 физических ядер и 6 потоков).

Сложнее всего приходилось процессору при создании массива. Это видно по графикам, особенно последнему.

Parallel провел самую крупную сборку мусора после сортировки слиянием. Остальные, судя пол графику, довольствовались минорными сборками.

Тем не менее, очевидна зависимость от ОЗУ и ядер процессора. На другой конфигурации были бы другие показатели.

Если сравнивать быстродействие сортировок, MergeSort на уровне погрешности. Bubble Sort - Z на 3 секунды больше других, InsertSort – на уровне погрешности.

На моем ПК больше всего HEAP был нагружен при работе G1 Bubble Sort. Процессор – Parallel.