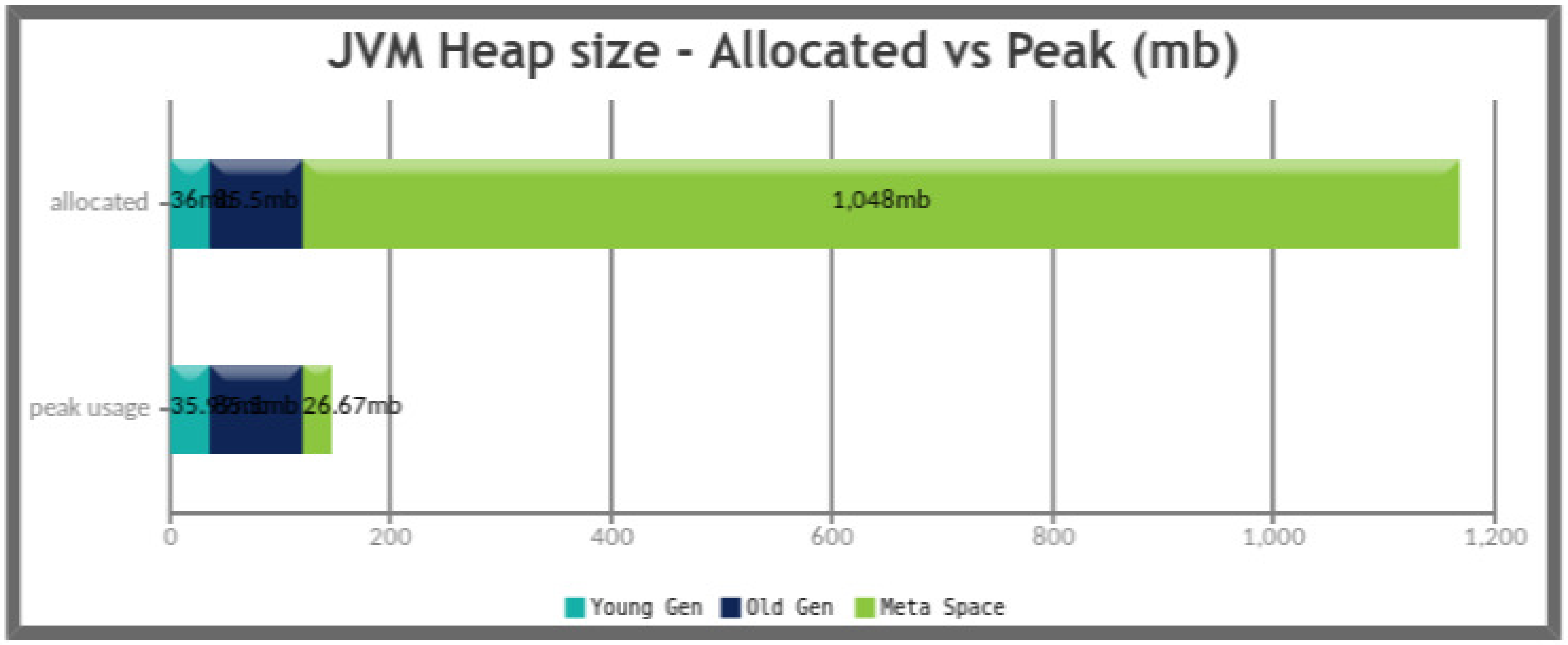
Analysis Report

JVM Heap Size

Generation	Allocated	Peak
Young Generation	36 mb	35.99 mb
Old Generation	85.5 mb	85.5 mb
Meta Space	1.02 gb	26.67 mb
Young + Old + Meta space	1.15 gb	138.67 mb



JVM Heap size - Allocated vs Peak (mb)

Key Performance Indicators

(Important section of the report. To learn more about KPIs, click here)

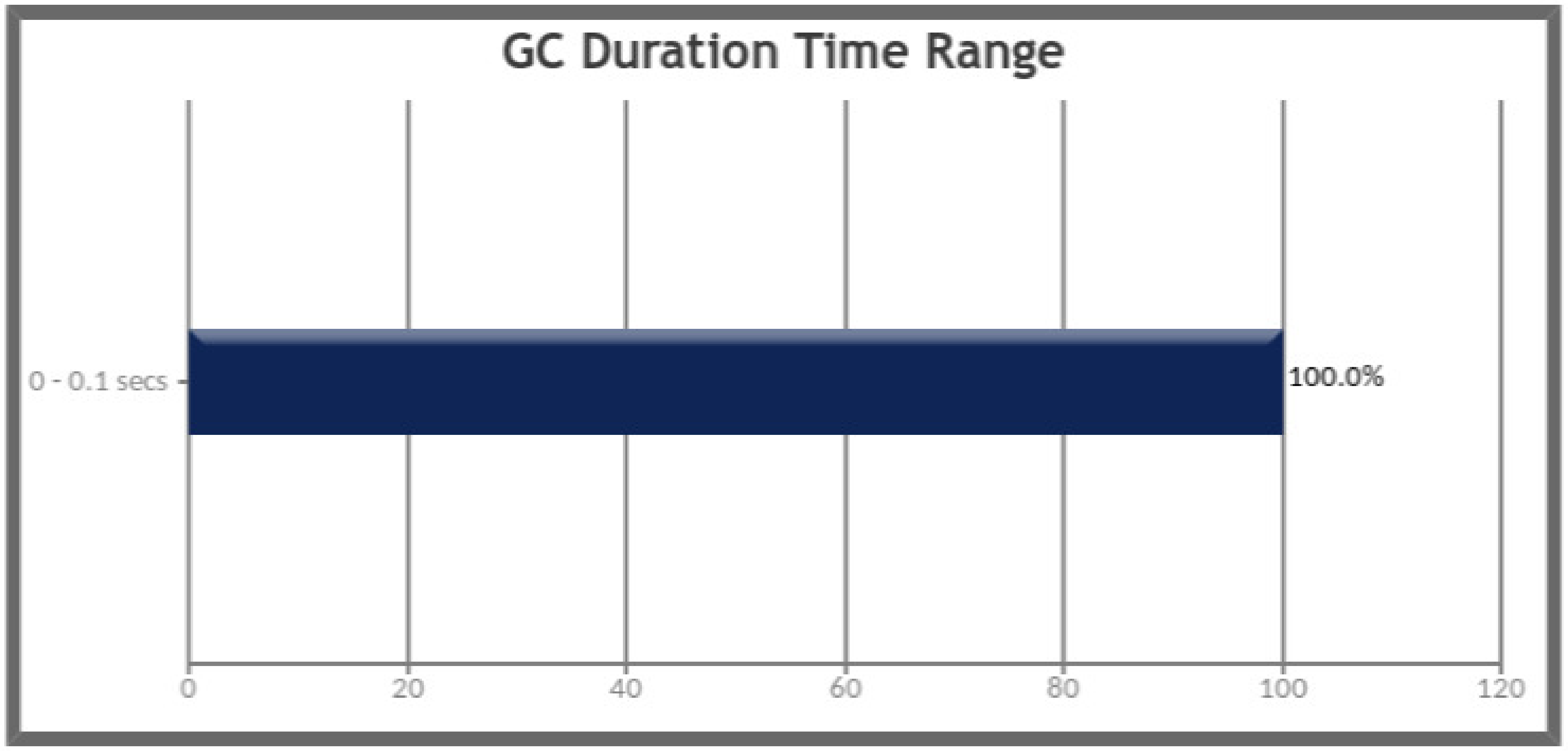
❶ **Throughput : 58.746%**

❷ **Latency:**

Avg Pause GC Time	**22 ms**
Max Pause GC Time	**80 ms**

GC **Pause** Duration Time Range:

Duration (secs)	No. of GCs	Percentage
0 - 0.1	210	100.0%

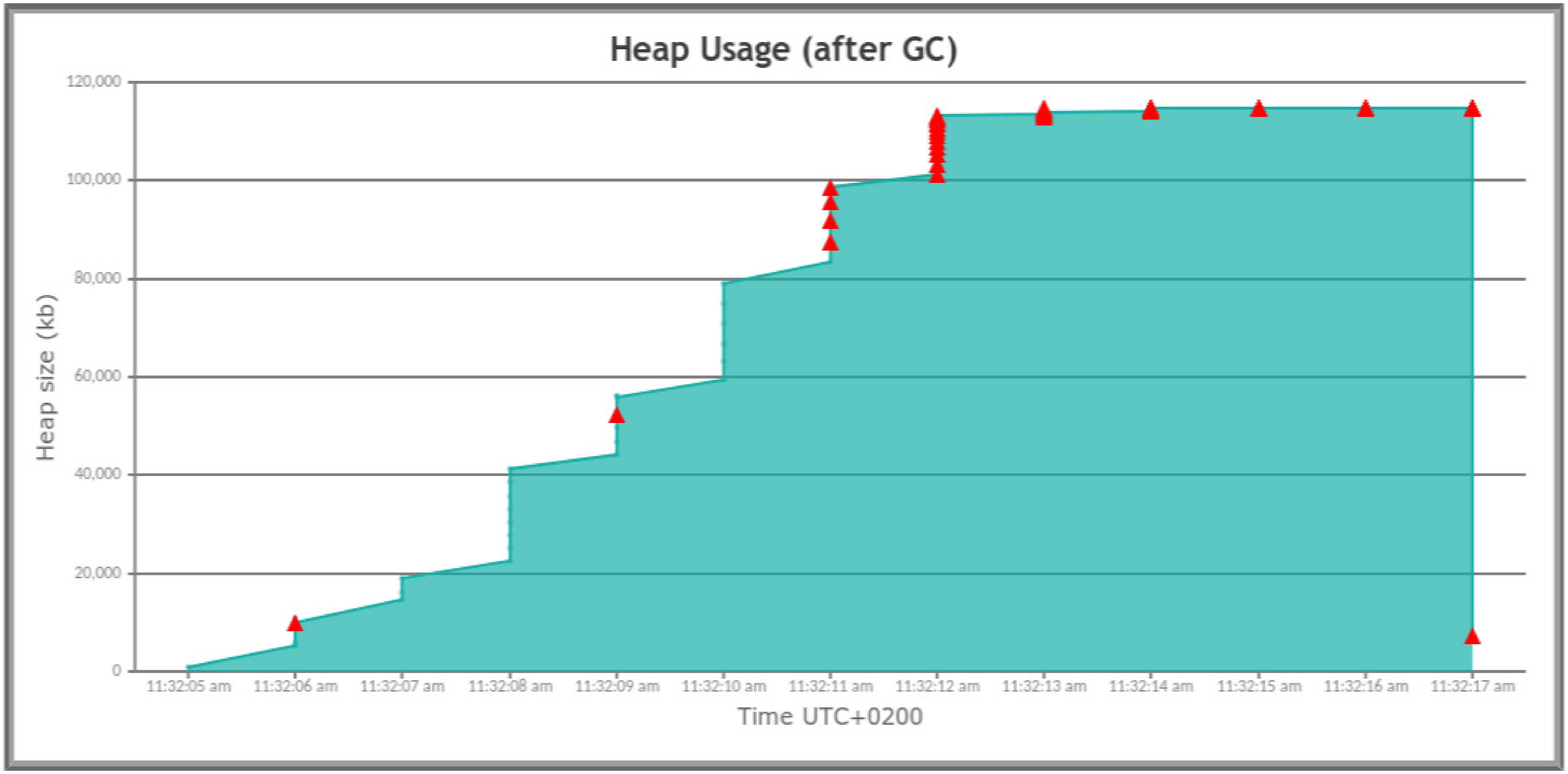


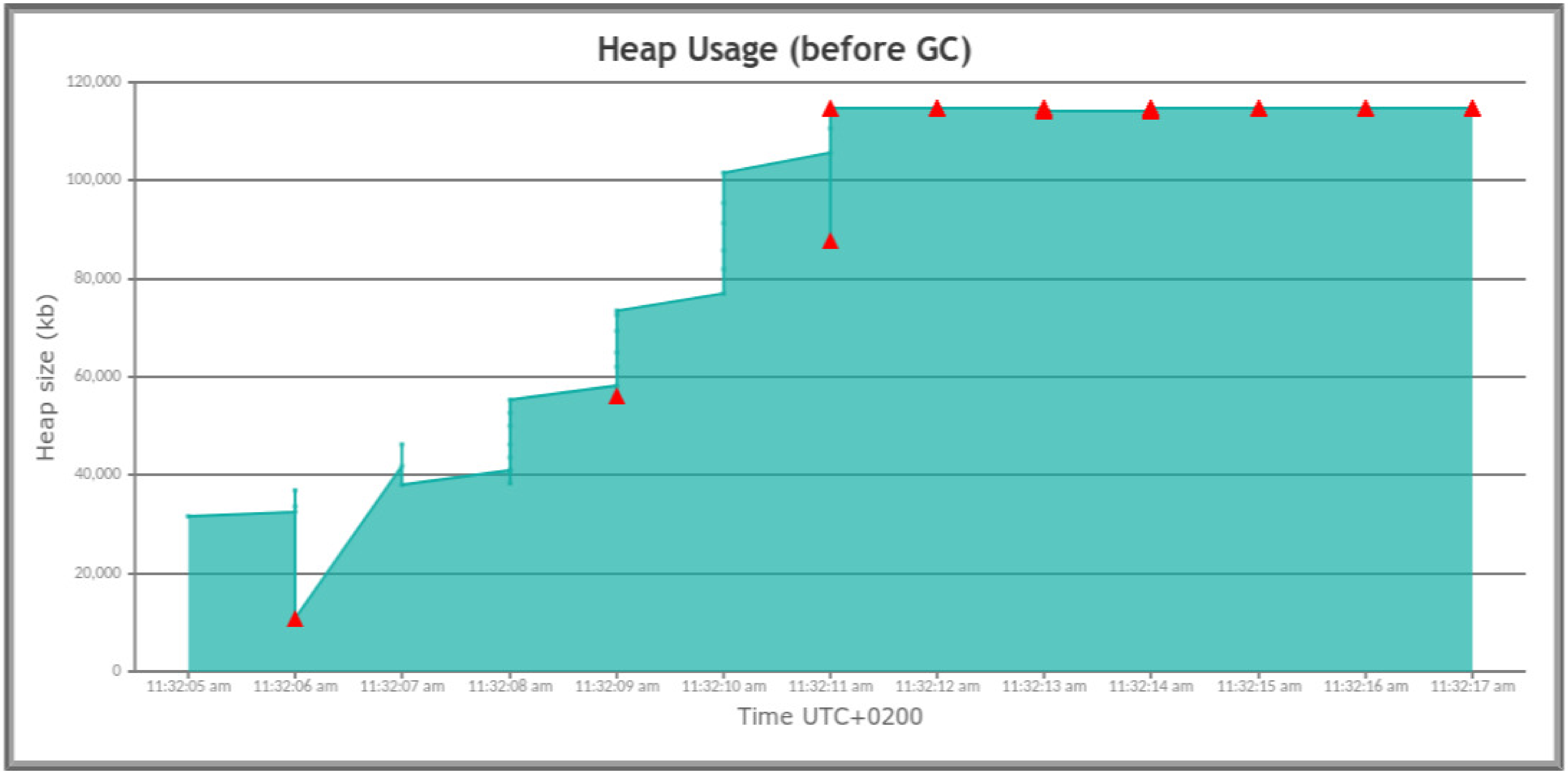
GC Duration Time Range

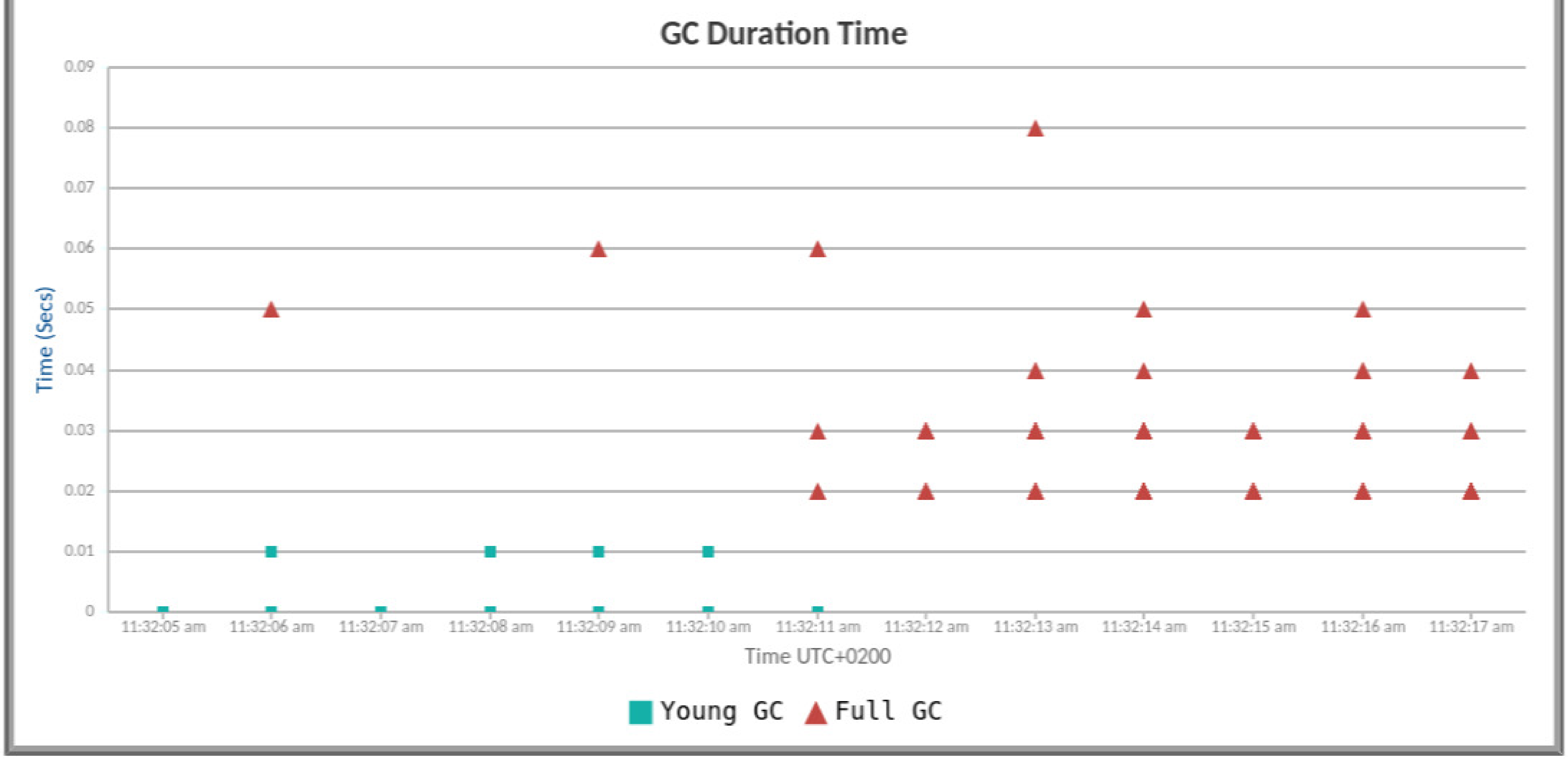
Interactive Graphs

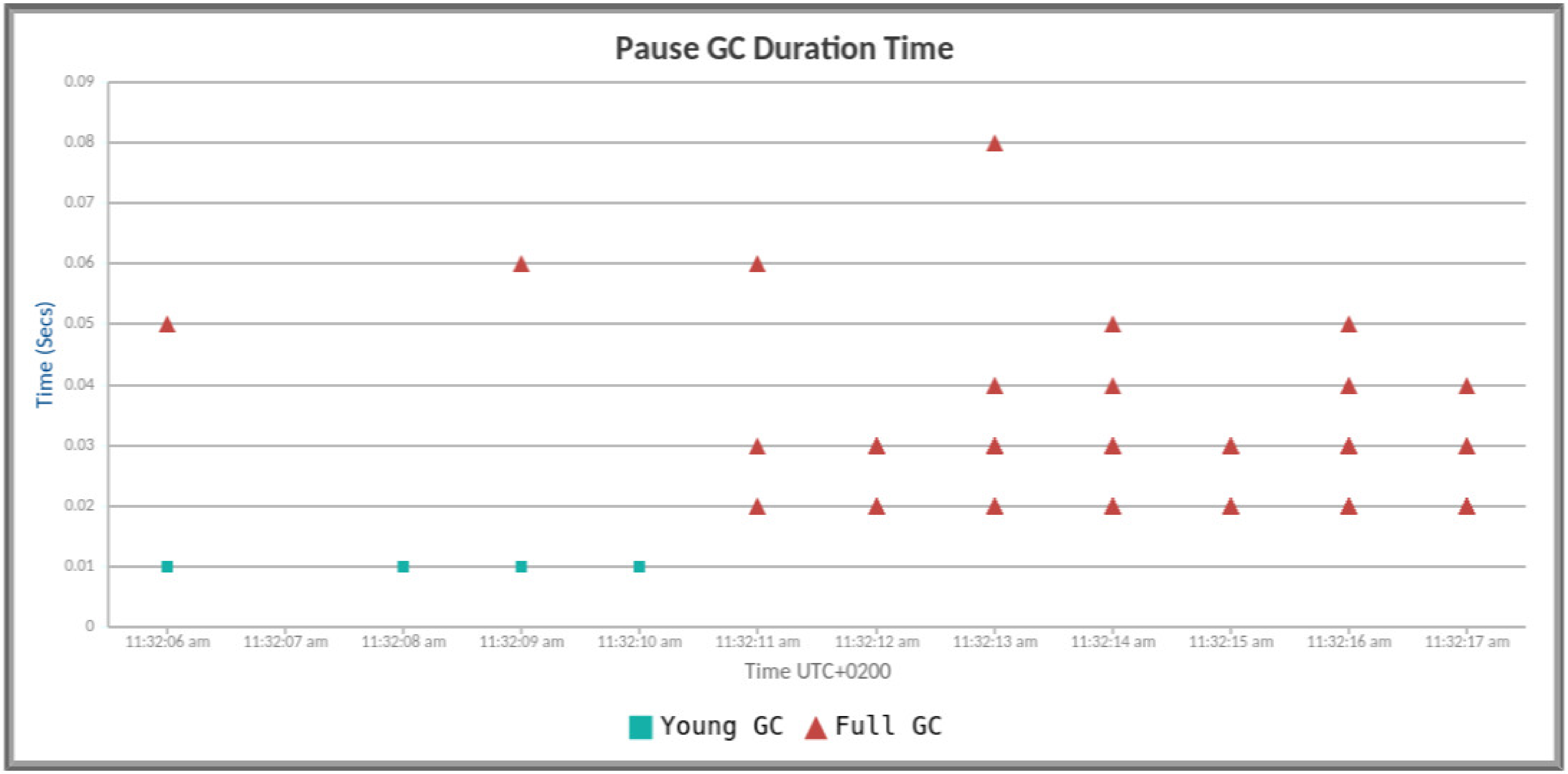
(All graphs are zoomable)

(All graphs are zoomable)

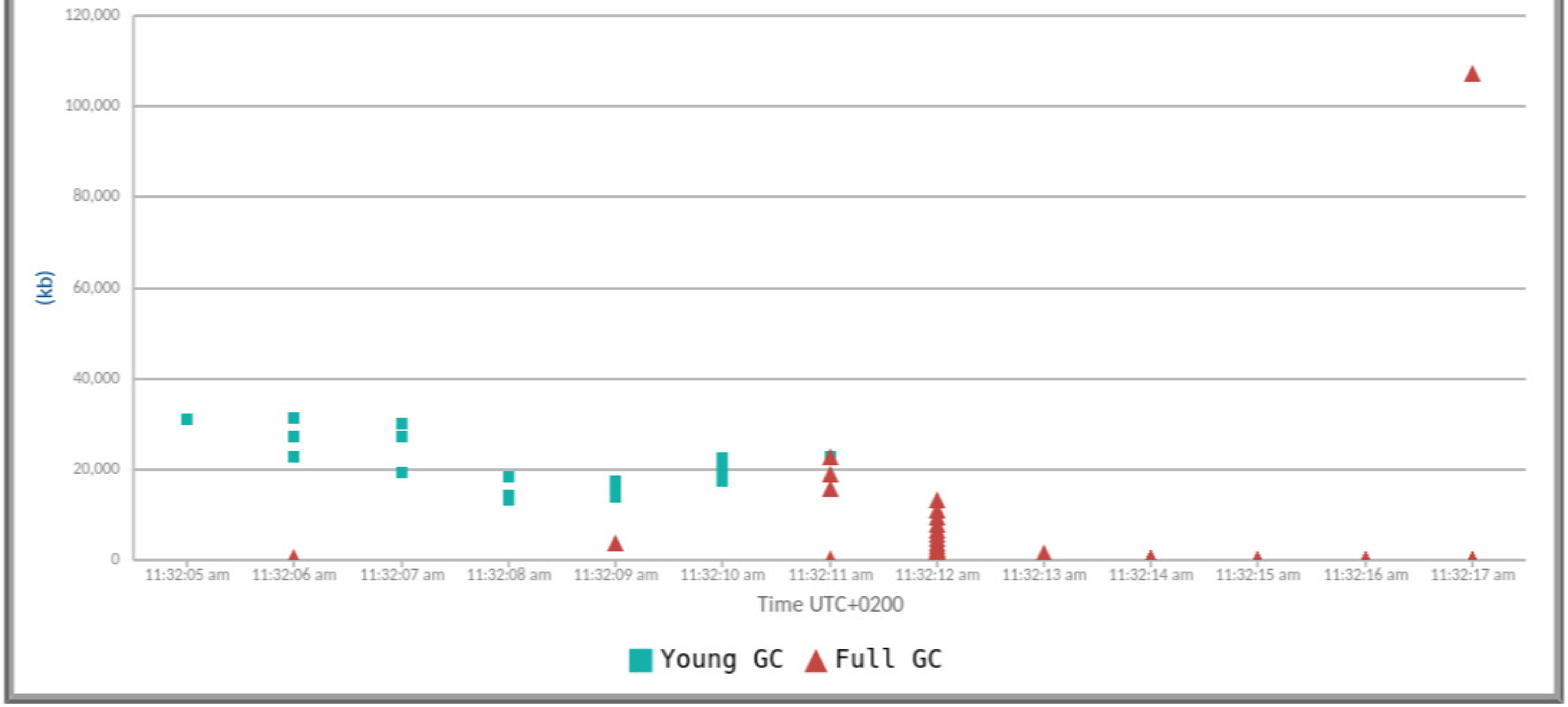




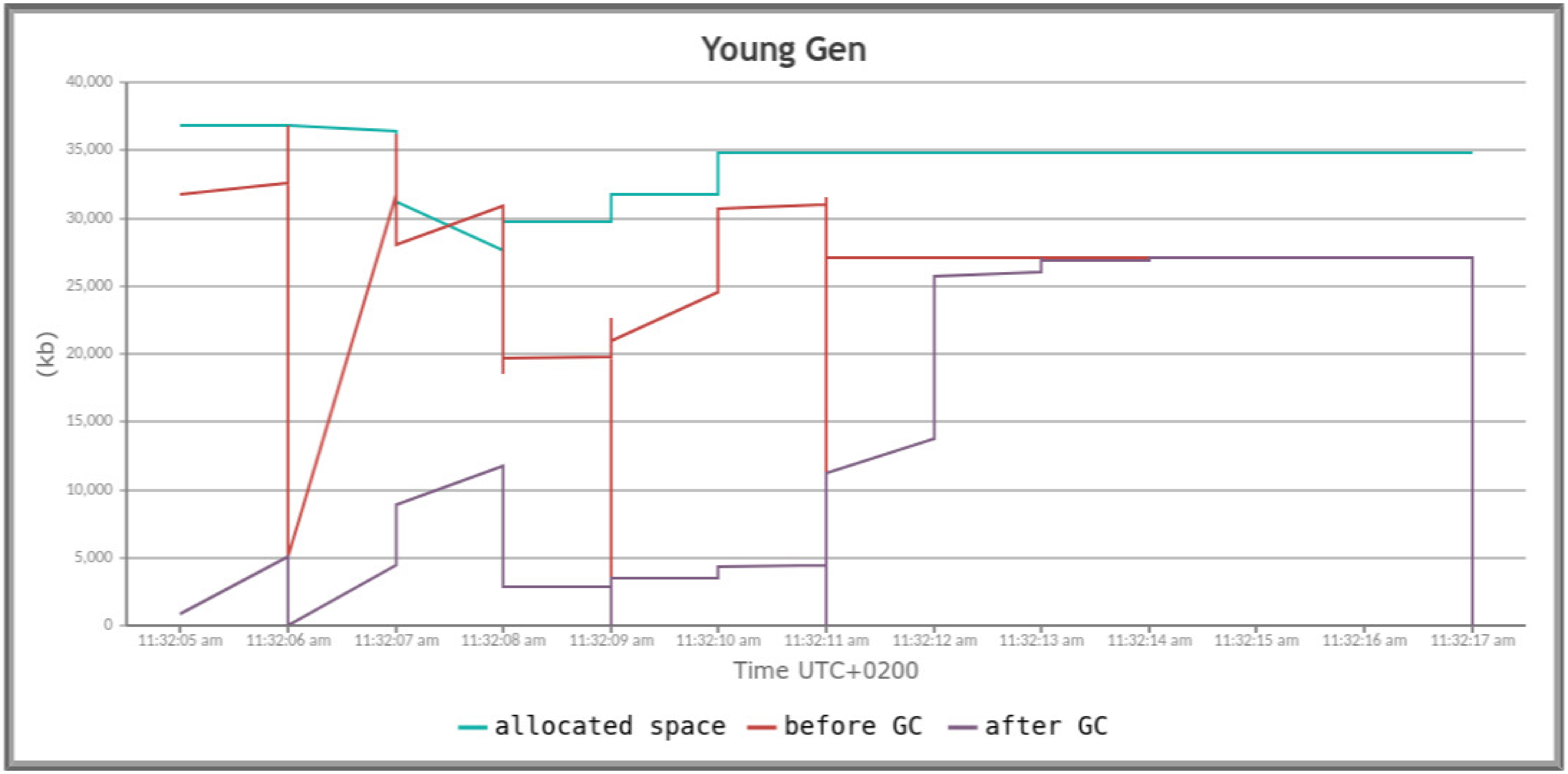




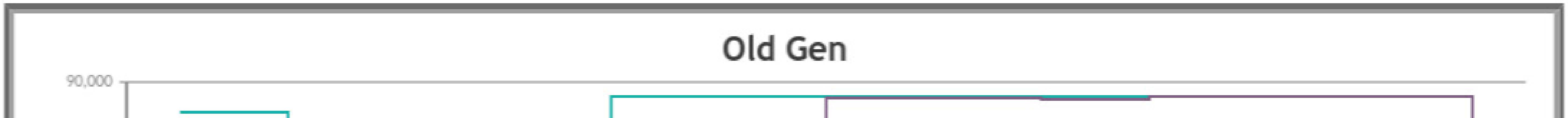
Reclaimed Bytes



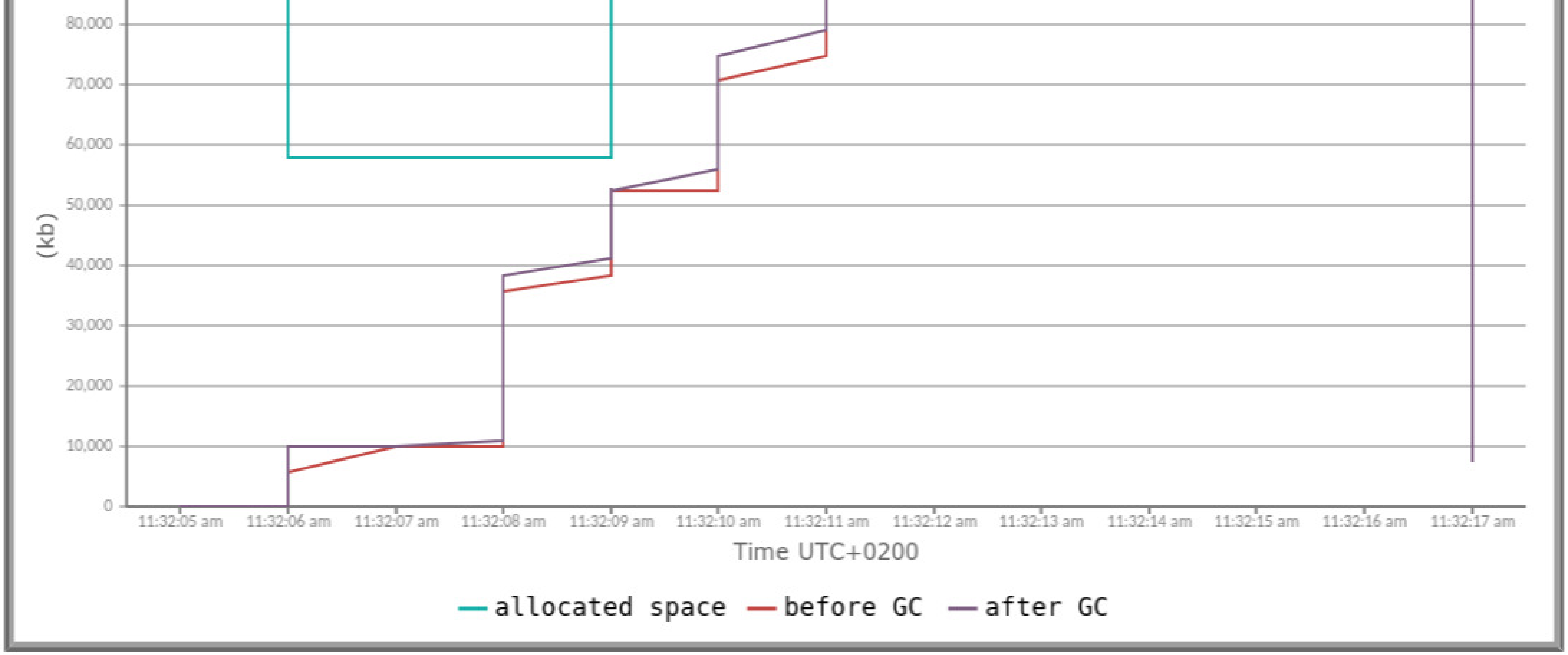
120,000
100,000
80,000
60,000
40,000
20,000
0
(kb)
11:32:05 am
11:32:06 am
11:32:07 am
11:32:08 am
11:32:09 am
11:32:10 am
11:32:11 am
11:32:12 am
11:32:13 am
11:32:14 am
11:32:15 am
11:32:16 am
11:32:17 am
Time UTC+0200
Young GC
Full GC

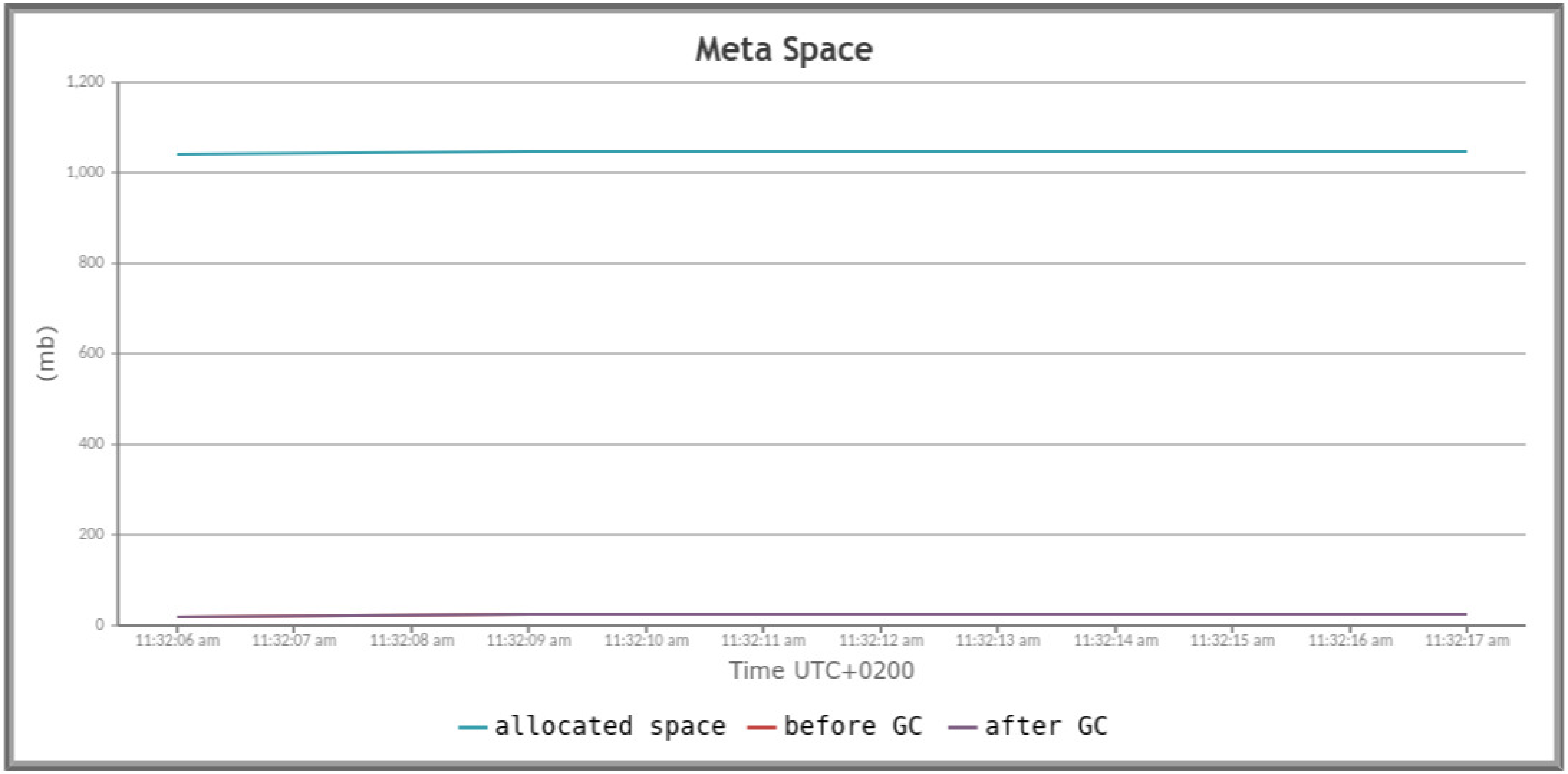


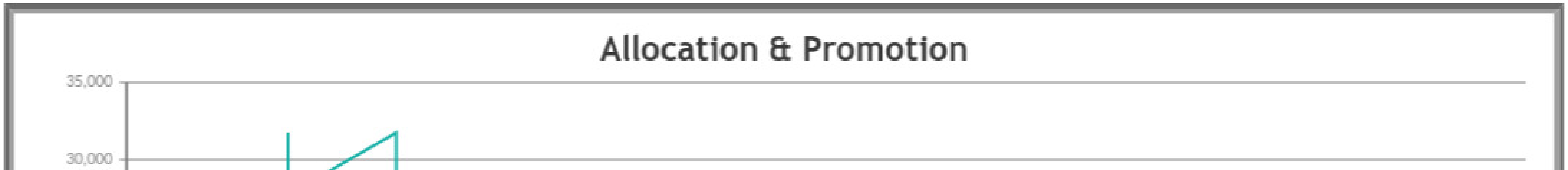
Young Gen
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
(kb)
11:32:05 am
11:32:06 am
11:32:07 am
11:32:08 am
11:32:09 am
11:32:10 am
11:32:11 am
11:32:12 am
11:32:13 am
11:32:14 am
11:32:15 am
11:32:16 am
11:32:17 am
Time UTC+0200
allocated space
before GC
after GC

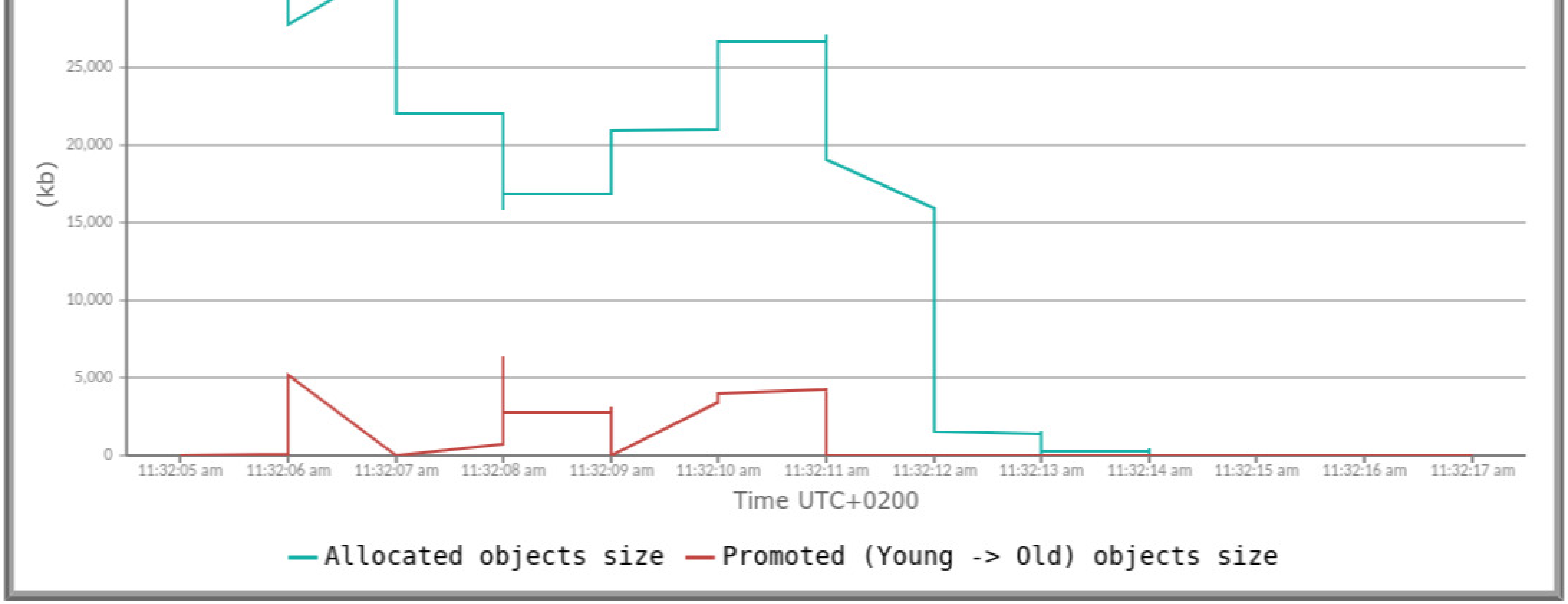


Old Gen
90,000

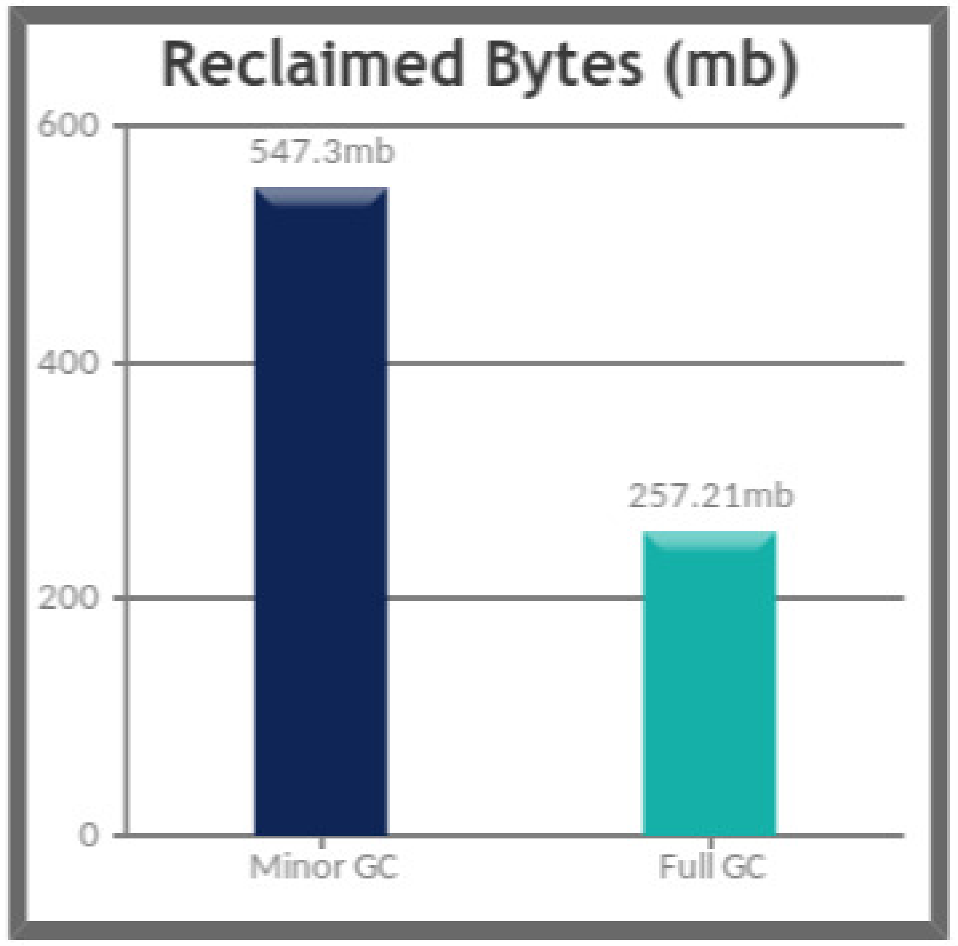

80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
(kb)
11:32:05 am
11:32:06 am
11:32:07 am
11:32:08 am
11:32:09 am
11:32:10 am
11:32:11 am
11:32:12 am
11:32:13 am
11:32:14 am
11:32:15 am
11:32:16 am
11:32:17 am
Time UTC+0200
allocated space
before GC
after GC

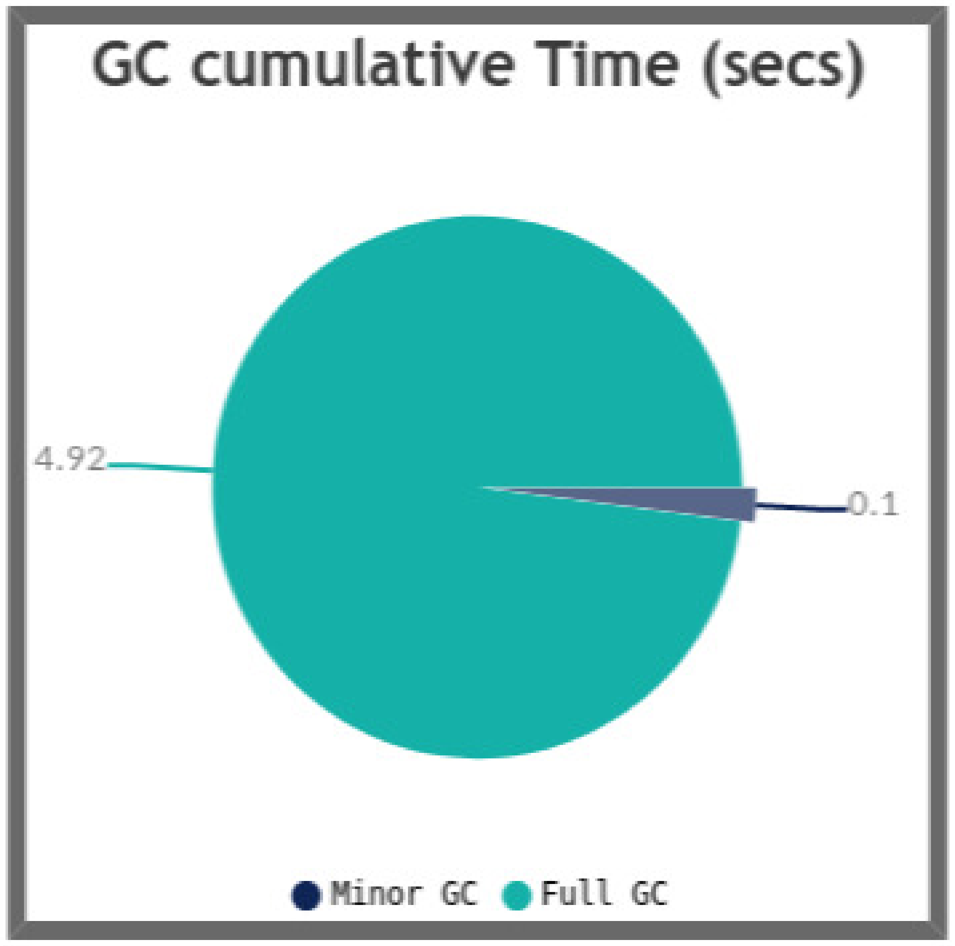

Meta Space
1,200
1,000
800
600
400
200
0
(mb)
11:32:06 am
11:32:07 am
11:32:08 am
11:32:09 am
11:32:10 am
11:32:11 am
11:32:12 am
11:32:13 am
11:32:14 am
11:32:15 am
11:32:16 am
11:32:17 am
Time UTC+0200
allocated space
before GC
after GC

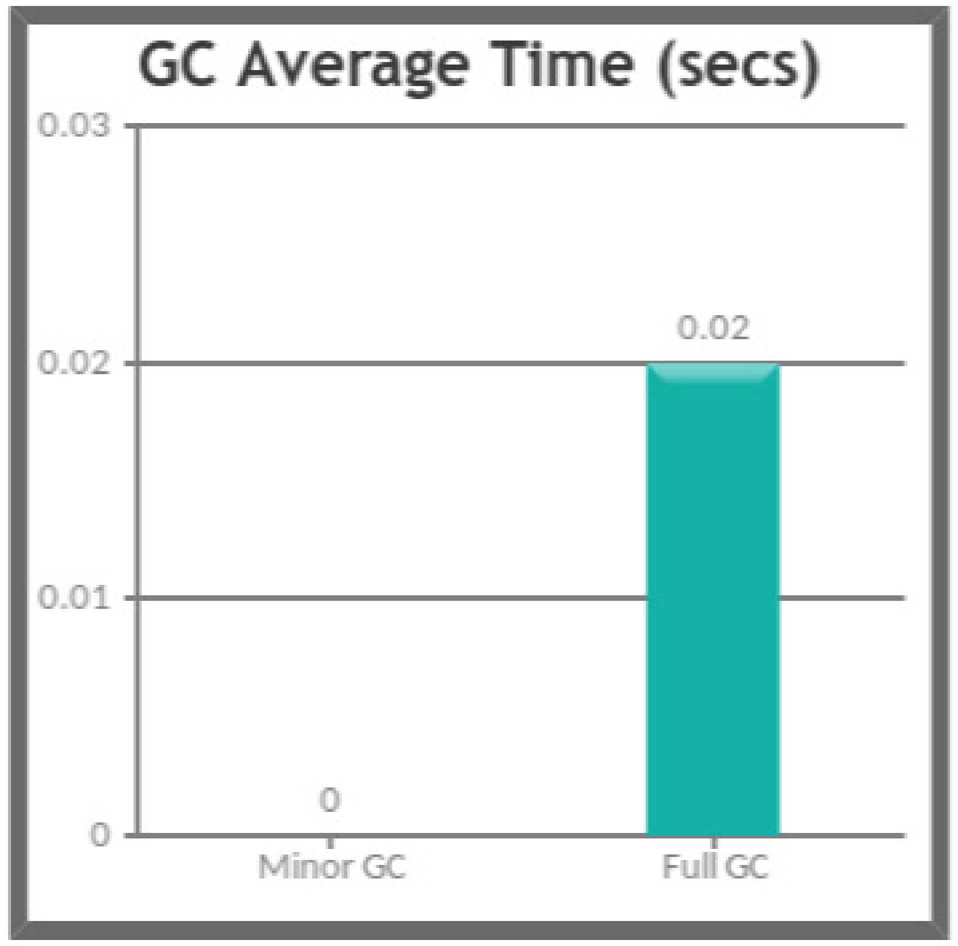

Allocation & Promotion
35,000
30,000



GC Statistics


Reclaimed Bytes (mb)


GC cumulative Time (secs)


GC Average Time (secs)

Total GC stats

Total GC count	229
Total reclaimed bytes	804.52 mb
Total GC time	5 sec 20 ms
Avg GC time	22 ms

Minor GC stats

Minor GC count	29
Minor GC reclaimed	547.3 mb
Minor GC total time	100 ms
Minor GC avg time	3 ms

Full GC stats

Full GC Count	200
Full GC reclaimed	257.21 mb
Full GC total time	4 sec 920 ms
Full GC avg time	25 ms

GC avg time std dev	11 ms
GC min/max time	0 / 80 ms
GC Interval avg time	53 ms

Minor GC avg time std dev	5 ms
Minor GC min/max time	0 / 10 ms
Minor GC Interval avg	217 ms

Full GC avg time std dev	8 ms
Full GC min/max time	20 ms / 80 ms
Full GC Interval avg	53 ms

GC Pause Statistics

Pause Count	229
Pause total time	5 sec 20 ms
Pause avg time	22 ms
Pause avg time std dev	0.0
Pause min/max time	0 / 80 ms

Object Stats

(These are perfect micro-metrics to include in your performance reports)

Total created bytes	811.72 mb
Total promoted bytes	77.65 mb
Avg creation rate	66.72 mb/sec
Avg promotion rate	6.38 mb/sec

Memory Leak

No major memory leaks.

(**Note:** there are 8 flavours of OutOfMemoryErrors. With GC Logs you can diagnose only 5 flavours of them(Java heap space, GC overhead limit exceeded, Requested array size exceeds VM limit, Permgen space, Metaspace). So in other words, your application could be still suffering from memory leaks, but need other tools to diagnose them, not just GC Logs.)

Consecutive Full GC

None.

Long Pause

None.

Safe Point Duration

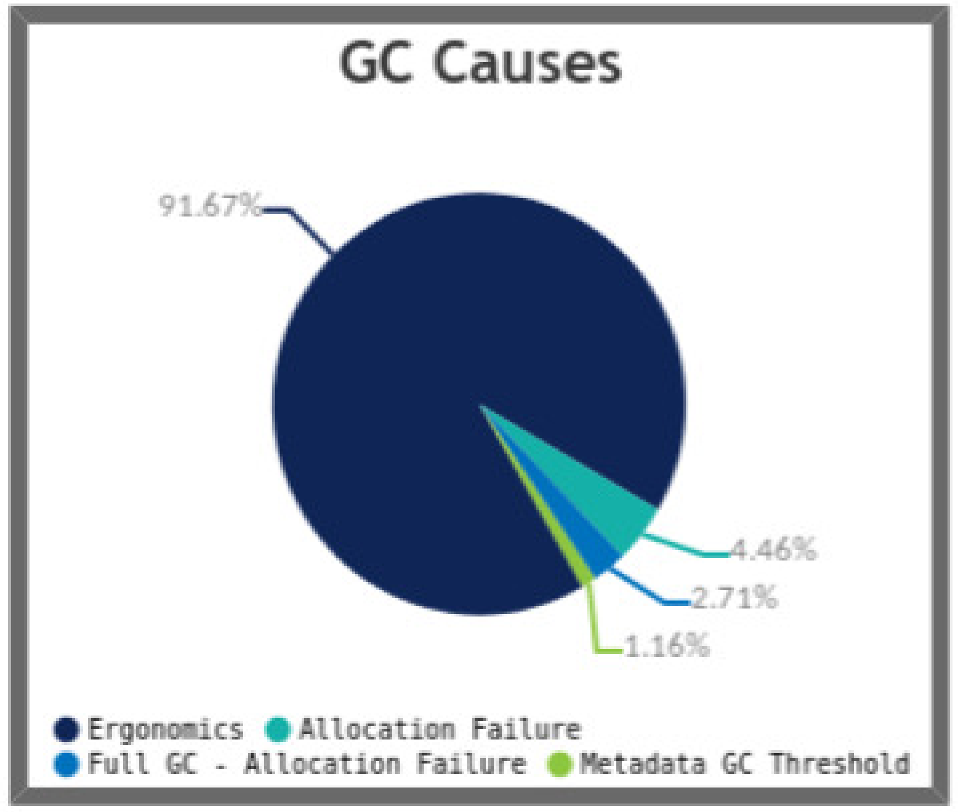
(To learn more about SafePoint duration, click here)

	Total Time	Avg Time	% of total duration
Total time for which app threads were stopped	5.07 secs	0.019 secs	42.25 %
Time taken to stop app threads	0.005 secs	0.0 secs	0.044 %

GC Causes

(What events caused the GCs, how much time it consumed?)

Cause	Count	Avg Time	Max Time	Total Time	Time %
Ergonomics	196	24 ms	60 ms	4 sec 730 ms	91.67%
Allocation Failure	28	8 ms	80 ms	230 ms	4.46%
Full GC - Allocation Failure	3	47 ms	80 ms	140 ms	2.71%
Metadata GC Threshold	2	30 ms	50 ms	60 ms	1.16%
Total	229	n/a	n/a	5 sec 160 ms	100.0%



Tenuring Summary

Not reported in the log.

Command Line Flags

```
-XX:GCLogFileSize=10485760 -XX:InitialHeapSize=129405184 -XX:MaxHeapSize=134217728 -XX:+PrintGC -XX:+PrintGCApplicationStoppedTime -XX:+PrintGCDateStamps -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintReferenceGC -XX:-PrintTenuringDistribution -XX:+UseCompressedClassPointers -XX:+UseCompressedOops -XX:-UseGCLogFileRotation -XX:+UseParallelGC
```