Analysis Report

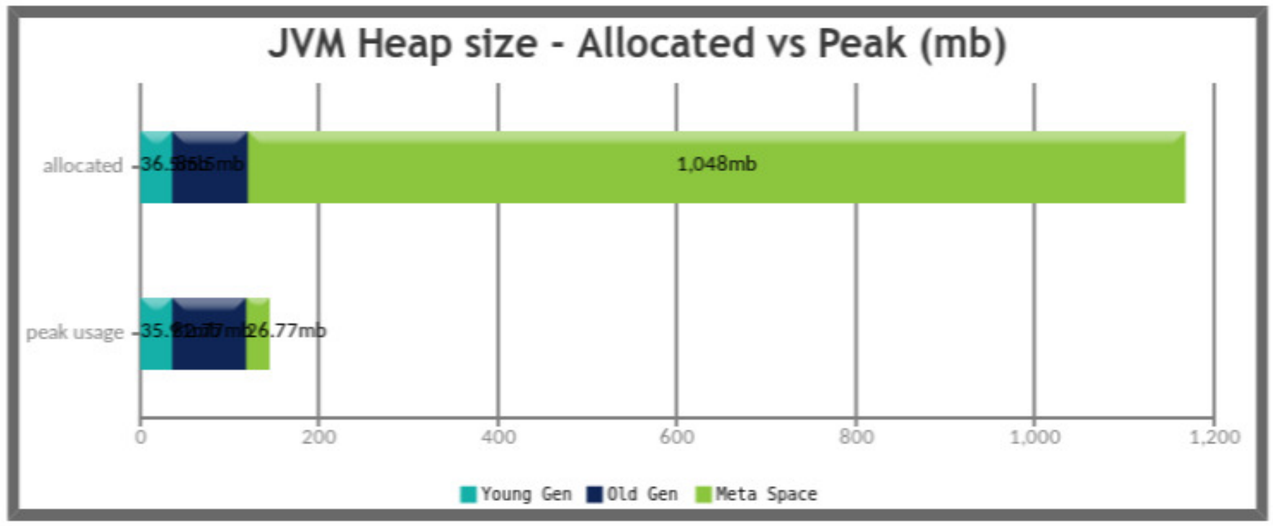
Tips to reduce GC Time

(**CAUTION:** Please do thorough testing before implementing out the recommendations. These are generic recommendations & may not be applicable for your application.)

- ✔ **6.25%** of GC time (i.e 70 ms) is caused by **'Metadata GC Threshold'**. This GC is triggered when metaspace got filled up and JVM wants to create new objects in this space..
  Solution:
  If this GC repeatedly happens, increase the metaspace size in your application with the command line option '-XX:MetaspaceSize'.

JVM Heap Size

Generation	Allocated	Peak
Young Generation	36.5 mb	35.91 mb
Old Generation	85.5 mb	82.77 mb
Meta Space	1.02 gb	26.77 mb
Young + Old + Meta space	1.15 gb	140.12 mb



Key Performance Indicators

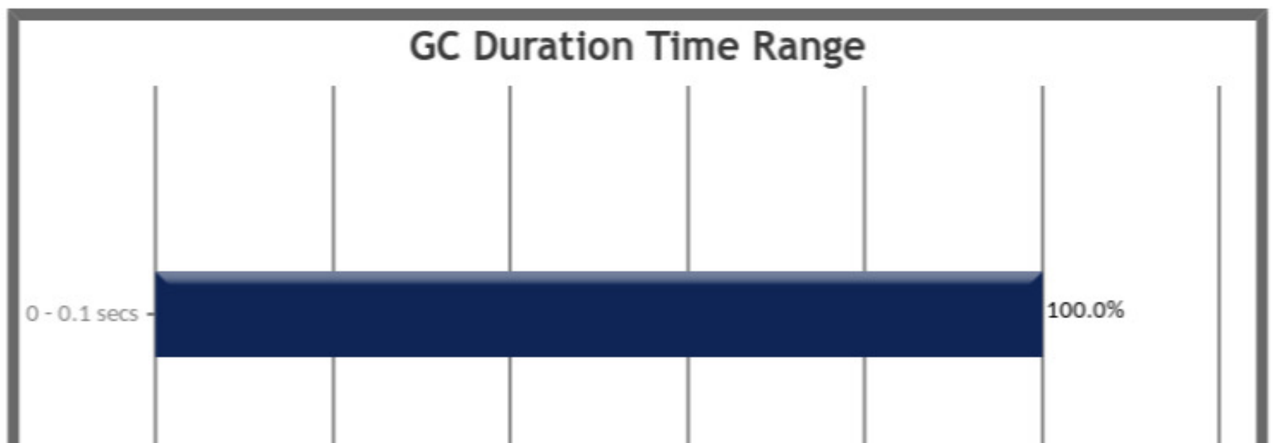
(Important section of the report. To learn more about KPIs, click here)

❶ Throughput : 96.929%

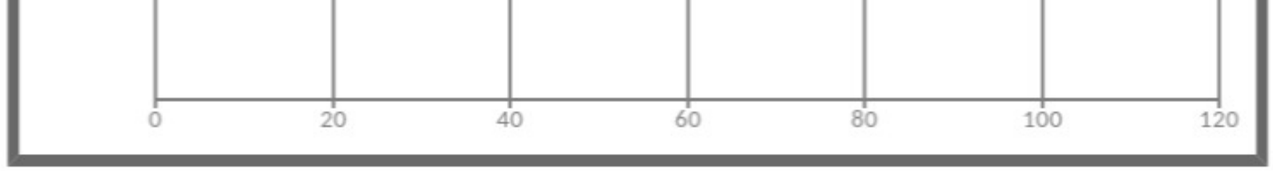
❷ Latency:

Avg Pause GC Time	6 ms
Max Pause GC Time	60 ms

GC **Pause** Duration Time Range :

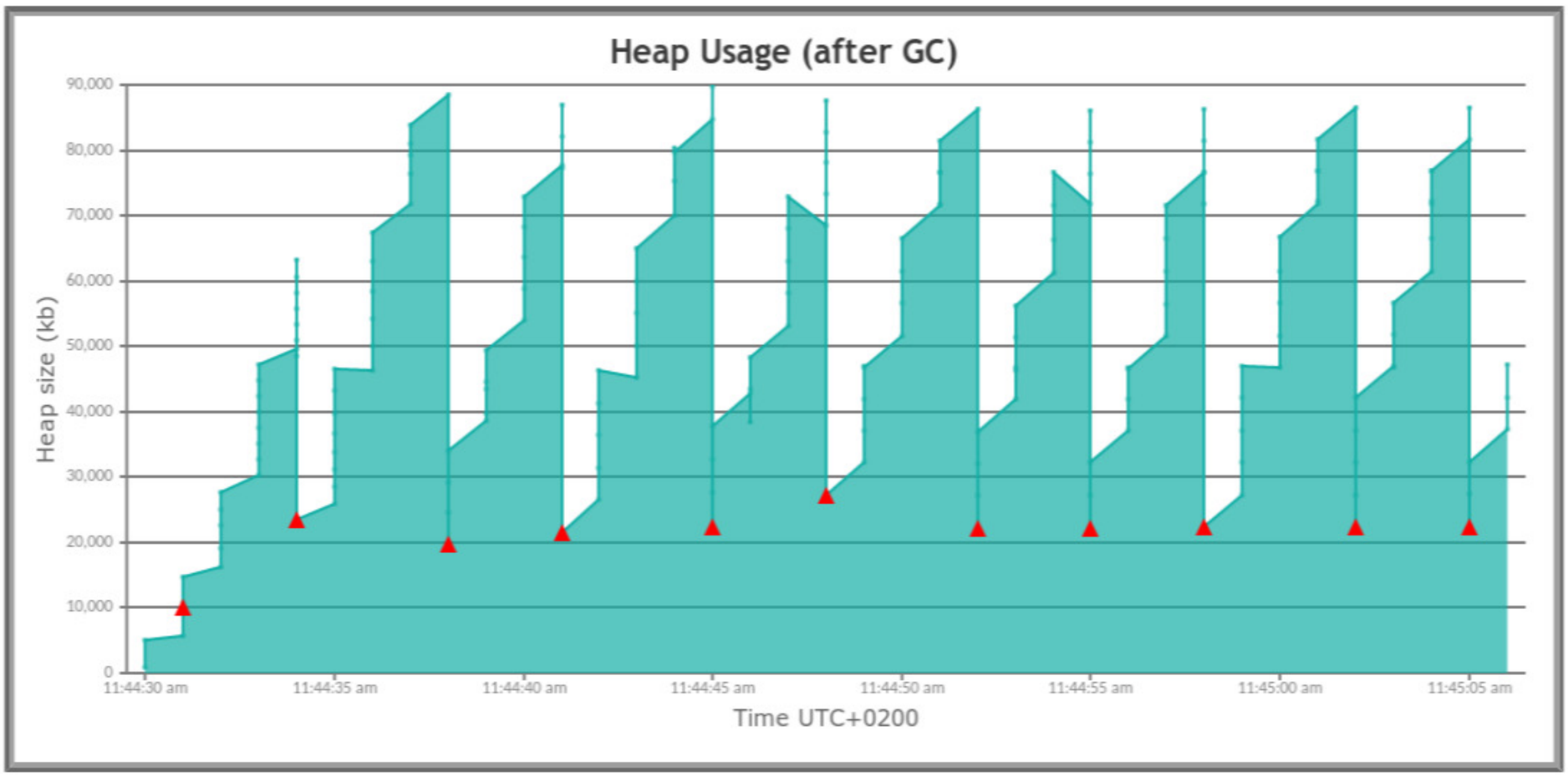


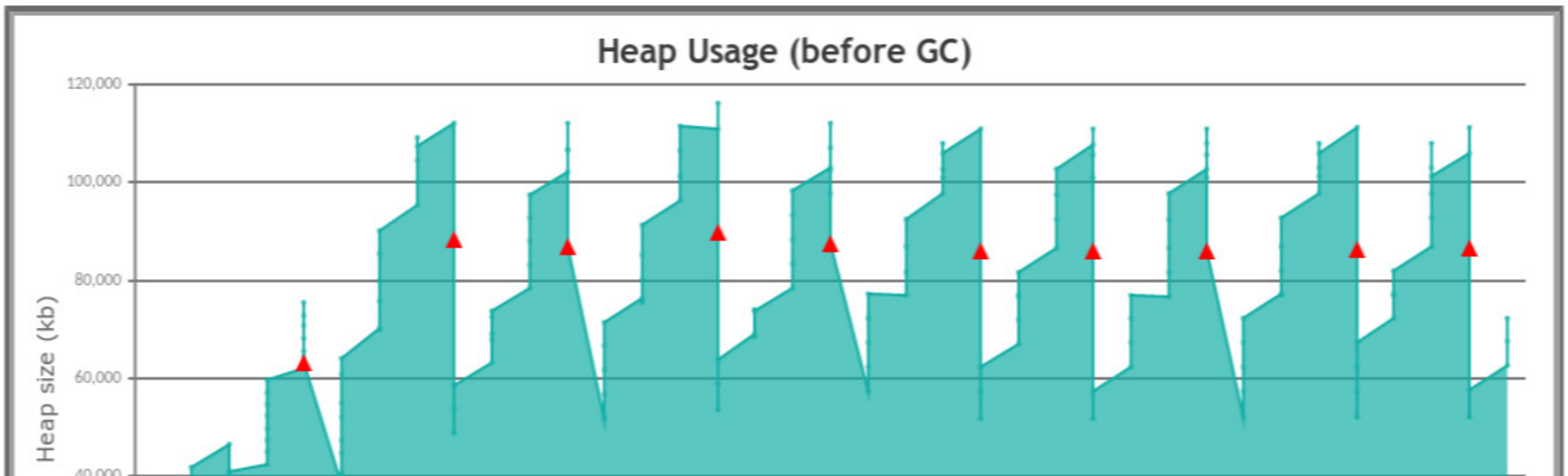
Duration (secs)	No. of GCs	Percentage
0 - 0.1	84	100.0%

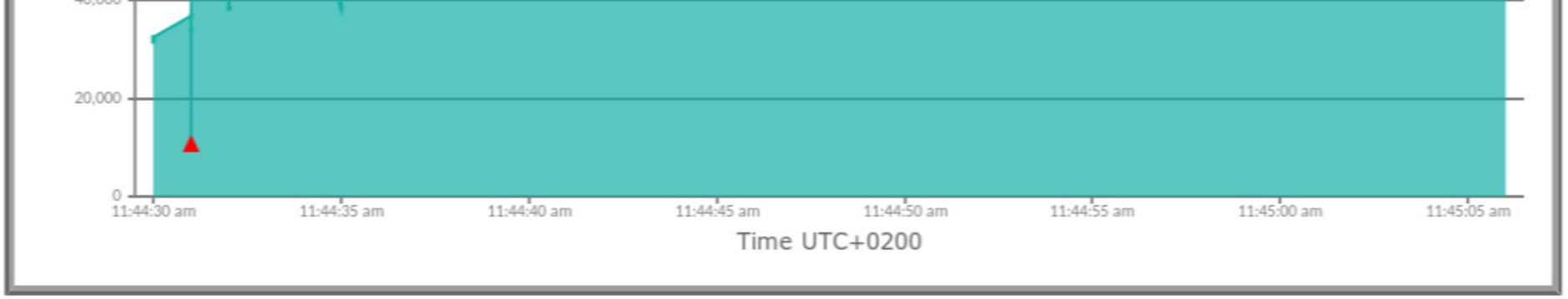


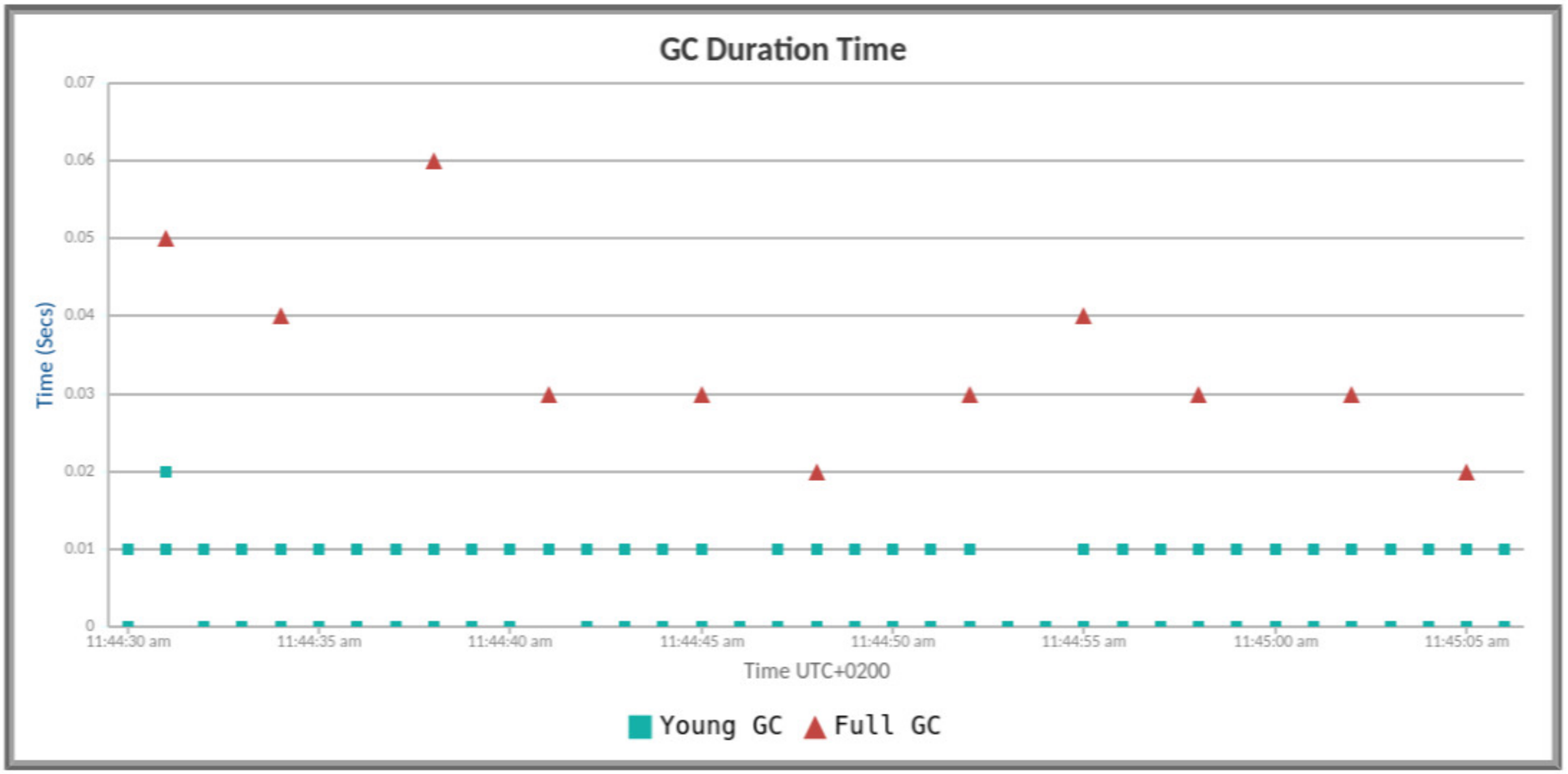
Interactive Graphs

(All graphs are zoomable)

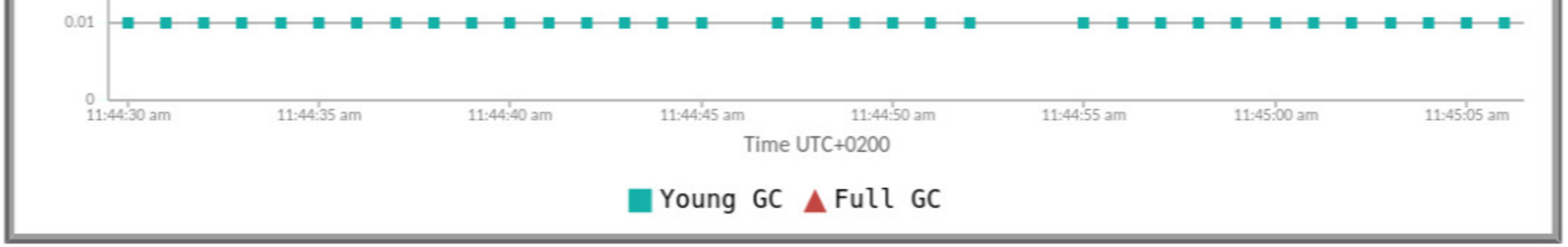


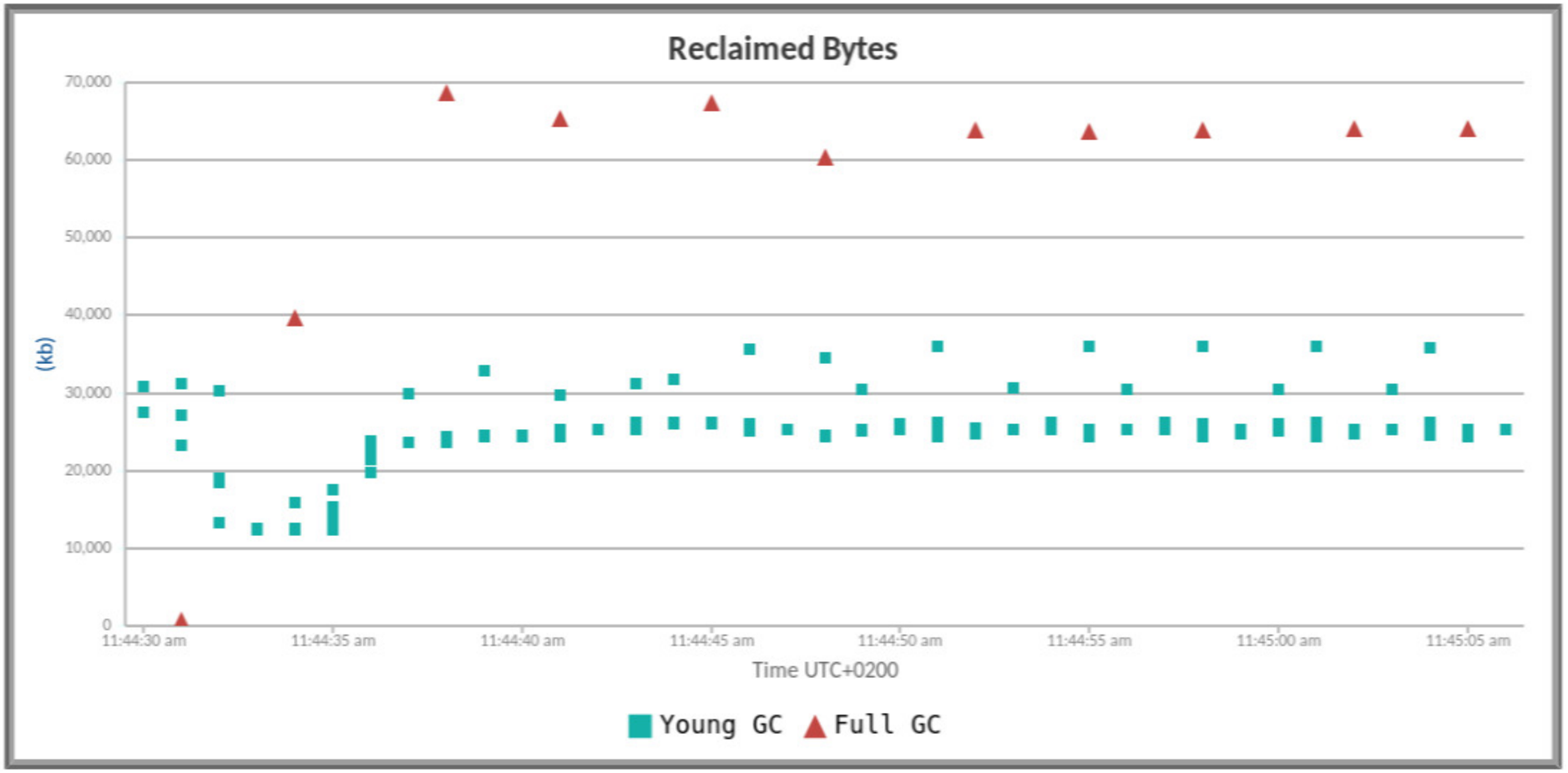



20,000
0
11:44:30 am
11:44:35 am
11:44:40 am
11:44:45 am
11:44:50 am
11:44:55 am
11:45:00 am
11:45:05 am
Time UTC+0200

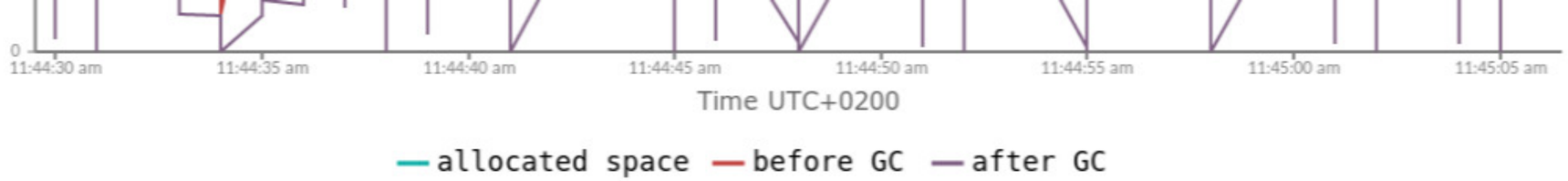

GC Duration Time
Time (Secs)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
11:44:30 am
11:44:35 am
11:44:40 am
11:44:45 am
11:44:50 am
11:44:55 am
11:45:00 am
11:45:05 am
Time UTC+0200
Young GC
Full GC

Pause GC Duration Time
Time (Secs)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
11:44:30 am
11:44:35 am
11:44:40 am
11:44:45 am
11:44:50 am
11:44:55 am
11:45:00 am
11:45:05 am
Time UTC+0200
Young GC
Full GC

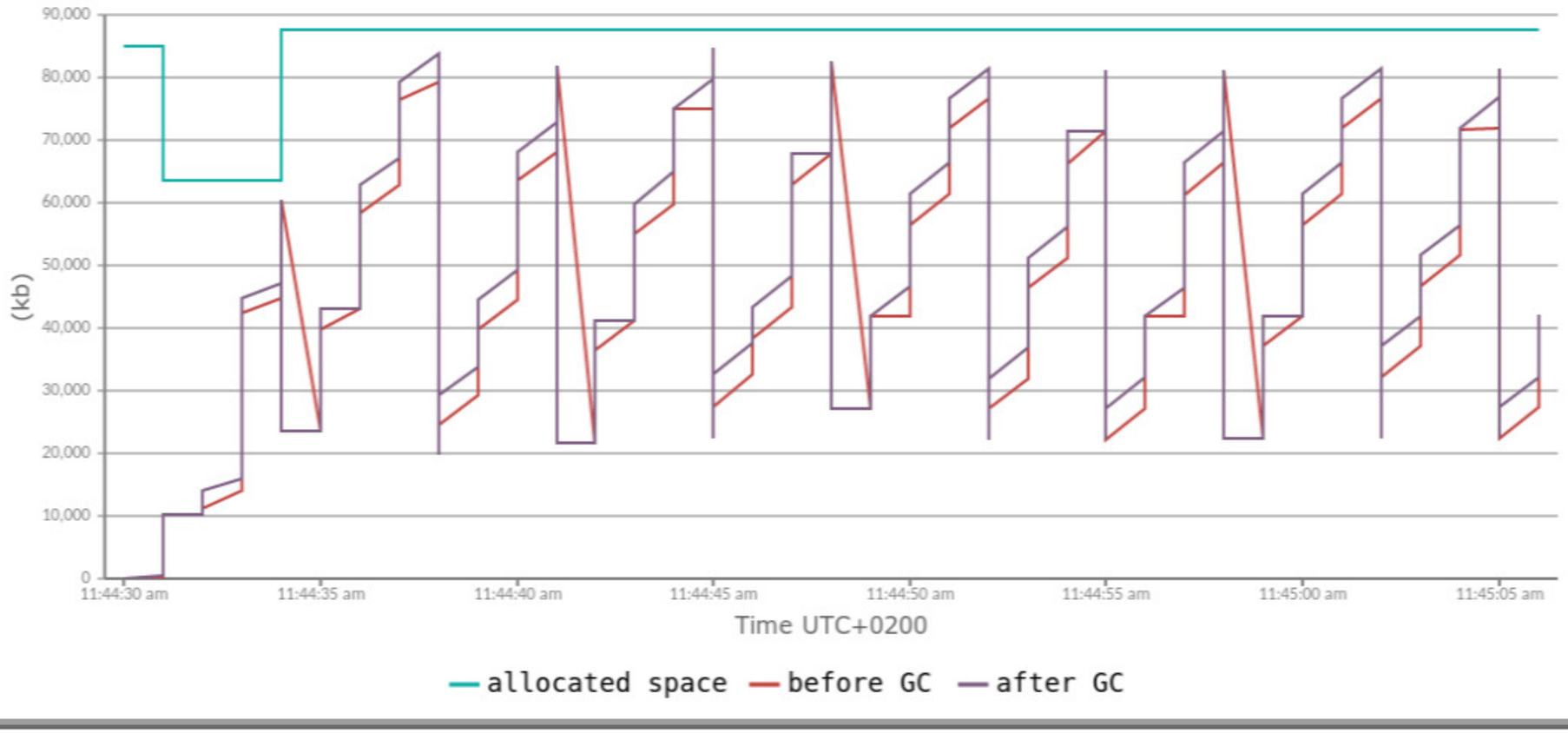

0.01
0
11:44:30 am
11:44:35 am
11:44:40 am
11:44:45 am
11:44:50 am
11:44:55 am
11:45:00 am
11:45:05 am
Time UTC+0200
Young GC
Full GC


Reclaimed Bytes
70,000
60,000
50,000
40,000
30,000
20,000
10,000
0
(kb)
11:44:30 am
11:44:35 am
11:44:40 am
11:44:45 am
11:44:50 am
11:44:55 am
11:45:00 am
11:45:05 am
Time UTC+0200
Young GC
Full GC

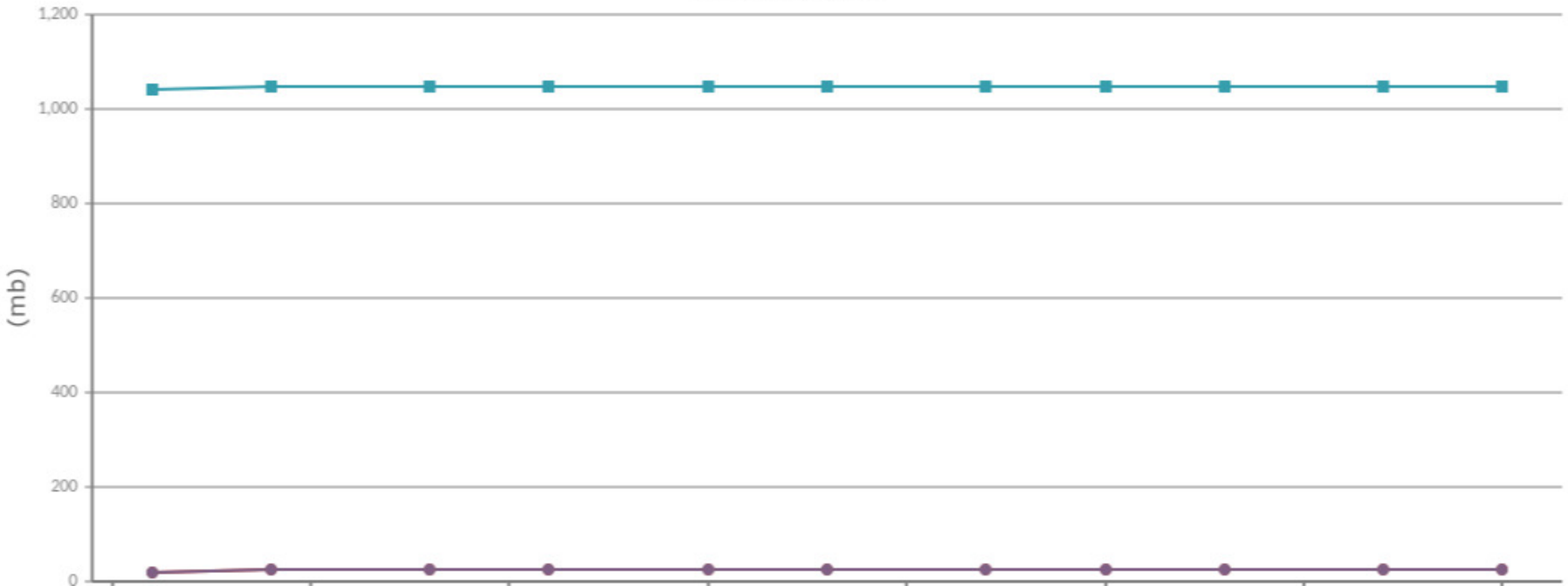
Young Gen
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
(kb)
11:44:30 am
11:44:35 am
11:44:40 am
11:44:45 am
11:44:50 am
11:44:55 am
11:45:00 am
11:45:05 am
Time UTC+0200
Young GC
Full GC

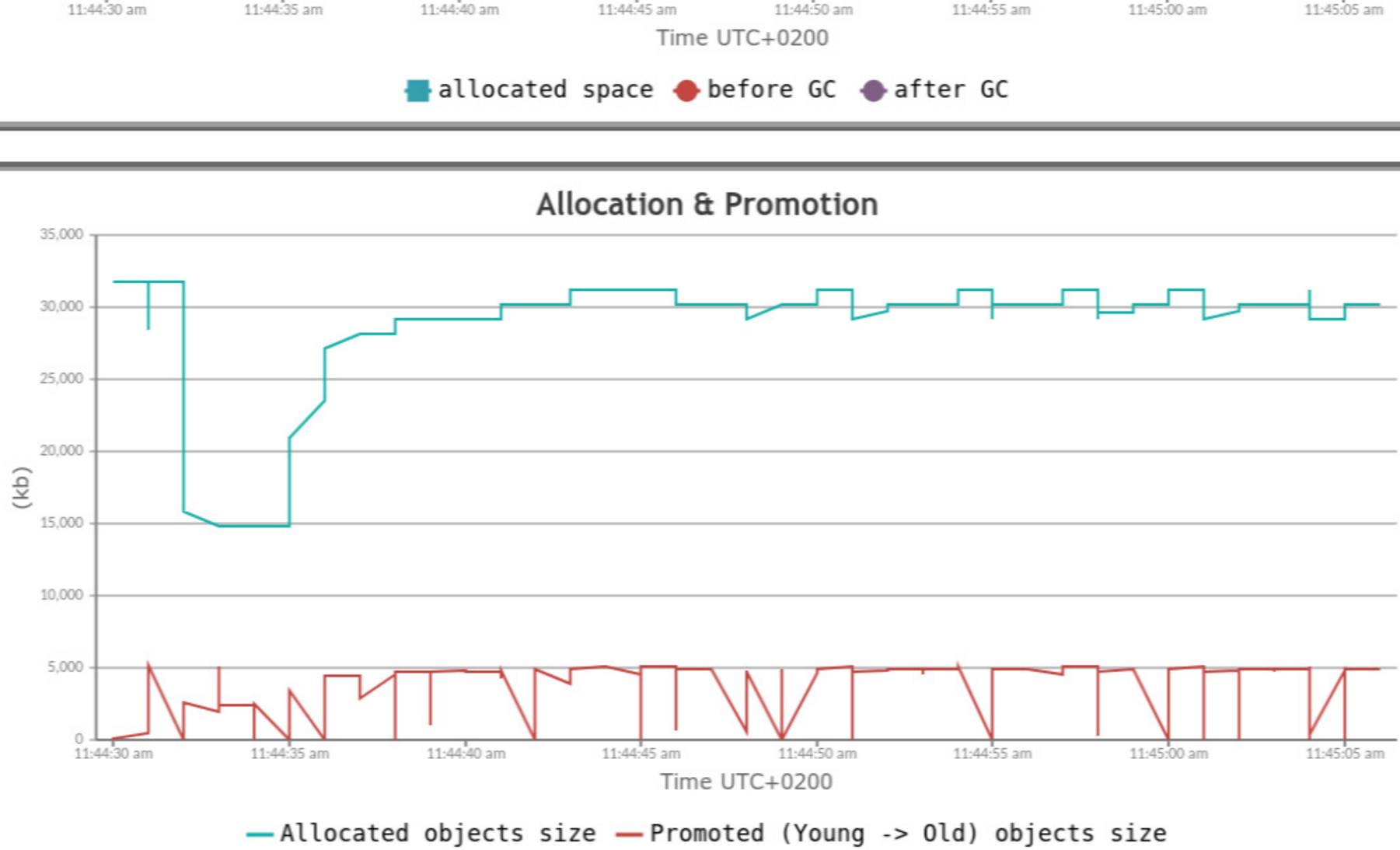


Old Gen



Meta Space





GC Statistics

Total GC stats

Total GC count	191
Total reclaimed bytes	4.73 gb
Total GC time	1 sec 120 ms
Avg GC time	6 ms
GC avg time std dev	9 ms
GC min/max time	0 / 60 ms
GC Interval avg time	191 ms

Minor GC stats

Minor GC count	180
Minor GC reclaimed	4.14 gb
Minor GC total time	740 ms
Minor GC avg time	4 ms
Minor GC avg time std dev	5 ms
Minor GC min/max time	0 / 20 ms
Minor GC Interval avg	203 ms

Full GC stats

Full GC Count	11
Full GC reclaimed	607.26 mb
Full GC total time	380 ms
Full GC avg time	35 ms
Full GC avg time std dev	12 ms
Full GC min/max time	20 ms / 60 ms
Full GC Interval avg	3 sec 386 ms

GC Pause Statistics

Pause Count	191
Pause total time	1 sec 120 ms
Pause avg time	6 ms
Pause avg time std dev	0.0
Pause min/max time	0 / 60 ms

Object Stats

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

Total created bytes	4.78 gb
Total promoted bytes	597.78 mb
Avg creation rate	134.19 mb/sec
Avg promotion rate	16.4 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

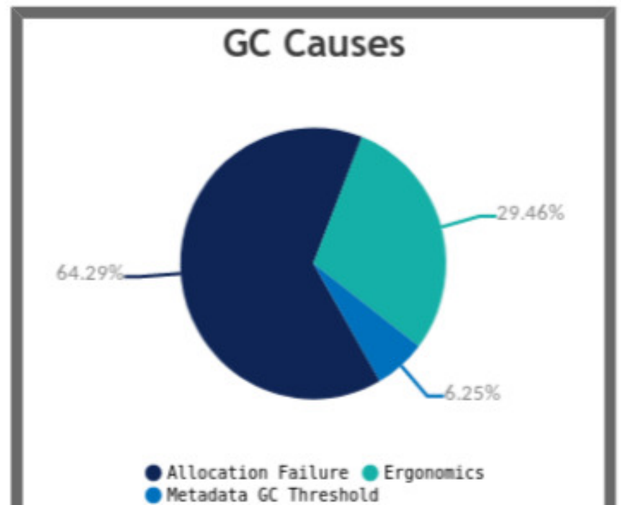
(To learn more about SafePoint duration, click here)

	Total Time	Avg Time	% of total duration
Total time for which app threads were stopped	1.167 secs	0.006 secs	3.241 %
Time taken to stop app threads	0.009 secs	0.0 secs	0.024 %

GC Causes

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

Cause	Count	Avg Time	Max Time	Total Time	Time %
Allocation Failure	179	4 ms	10 ms	720 ms	64.29%
Ergonomics	10	33 ms	60 ms	330 ms	29.46%
Metadata GC Threshold	2	35 ms	50 ms	70 ms	6.25%
Total	191	n/a	n/a	1 sec 120 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