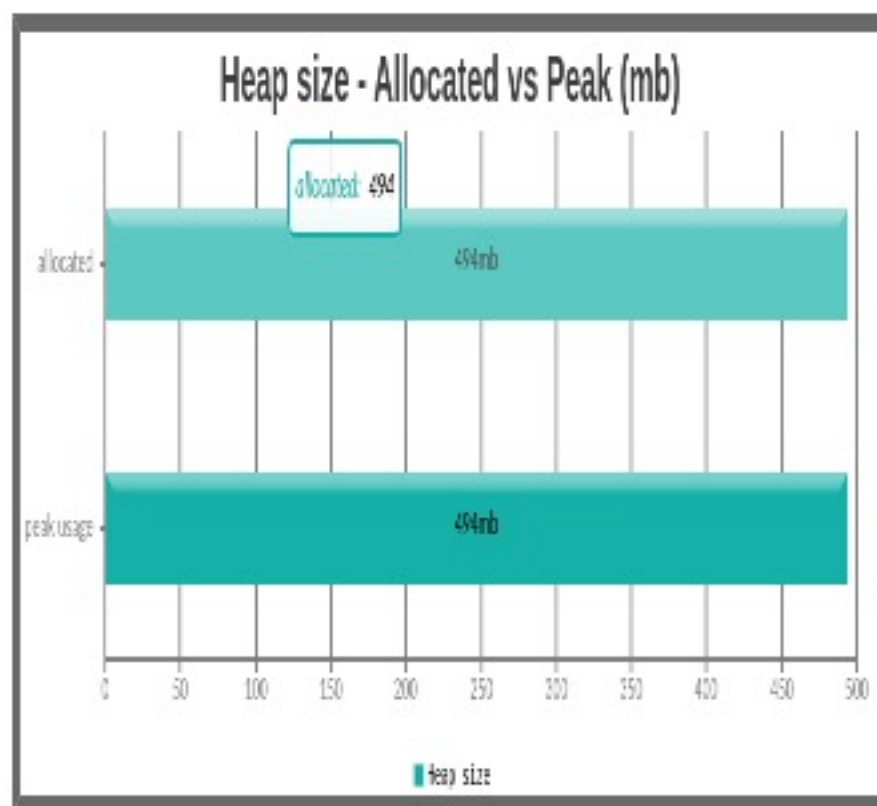


JVM Heap Size



Allocated Size: 494 mb

Peak Size: 494 mb

Key Performance Indicators

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

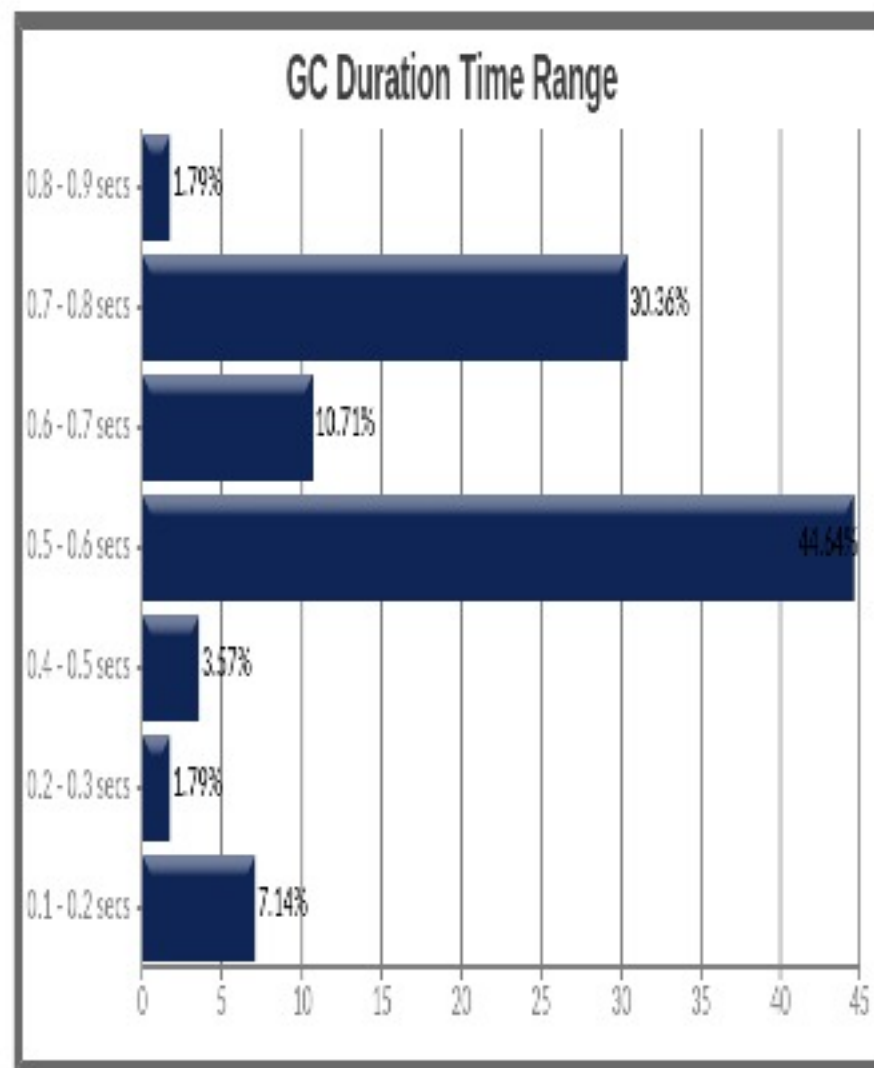
1 Throughput 88.385%

2 Latency:

Avg Pause GC Time	609 ms
Max Pause GC Time	876 ms

GC Pause Duration Time Range

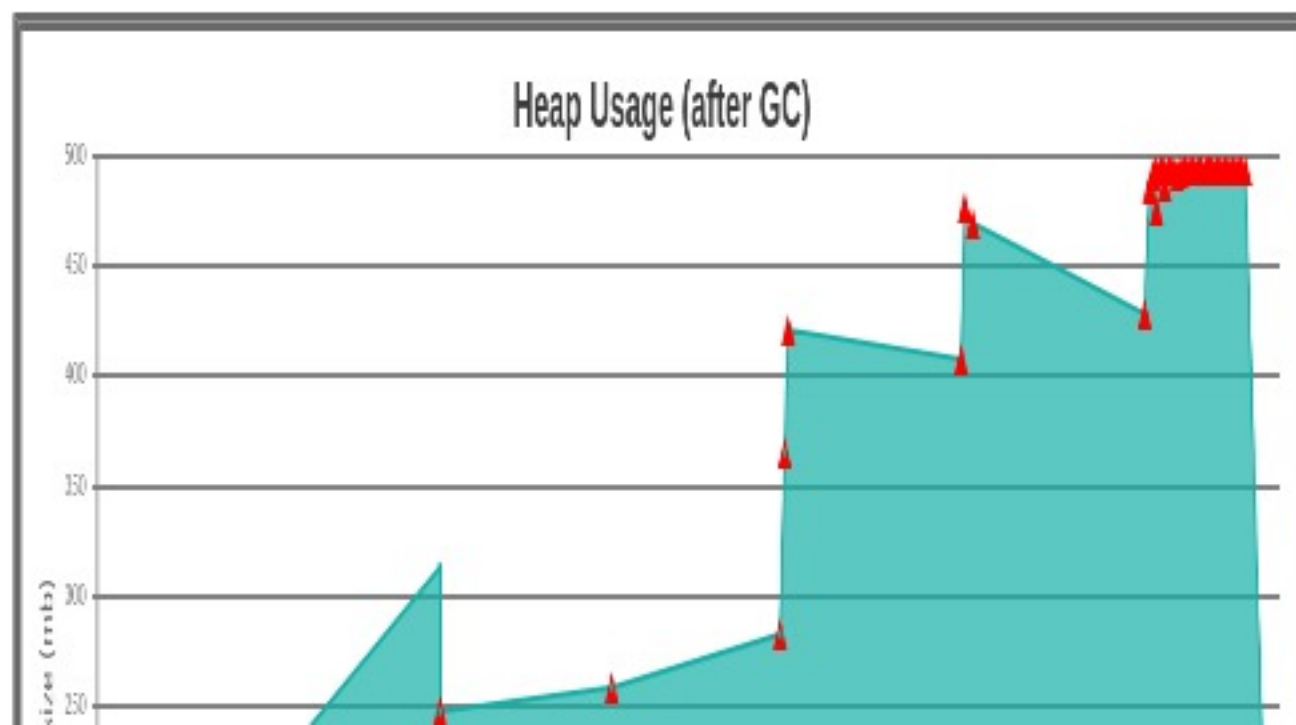
Duration (secs)	No. of GCs	Percentage
0.1 - 0.2	4	7.143%
0.2 - 0.3	1	8.929%
0.4 - 0.5	2	12.5%

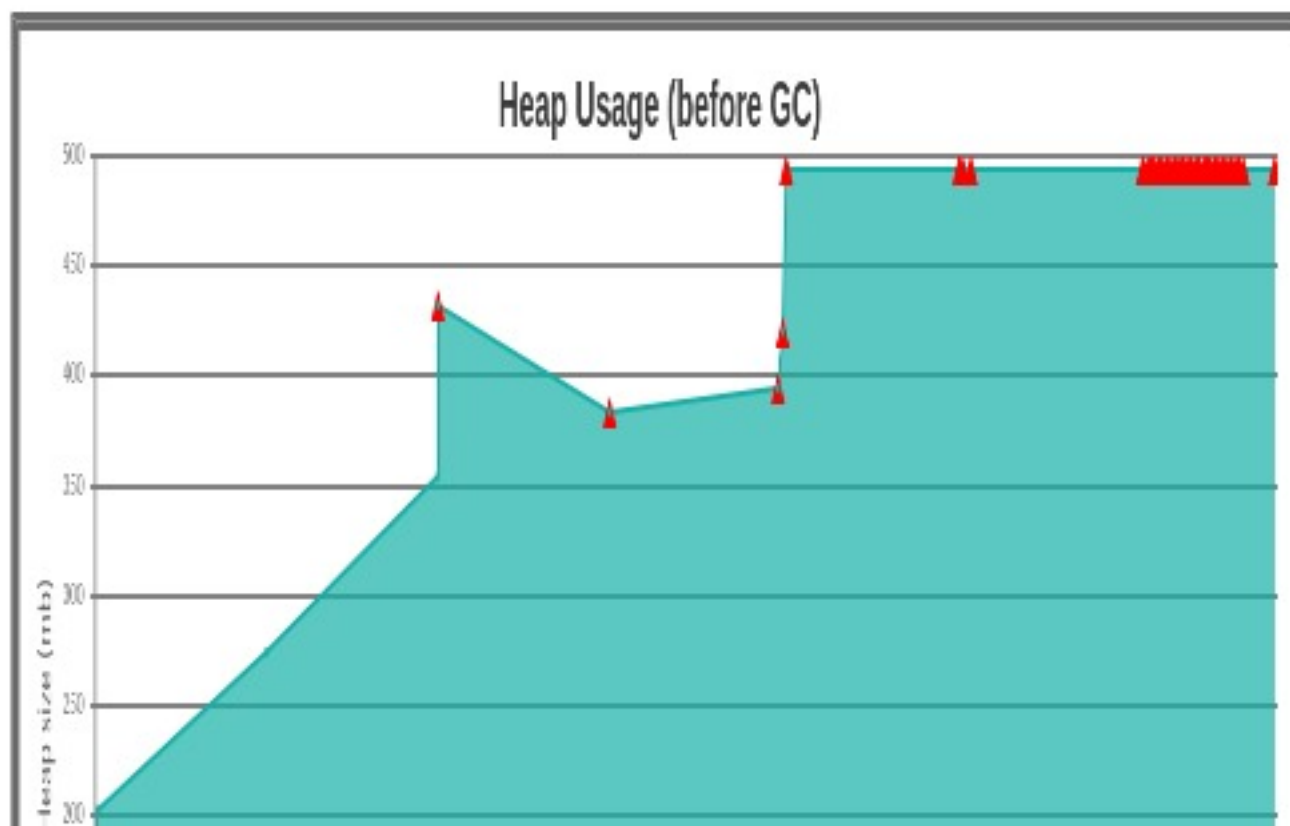


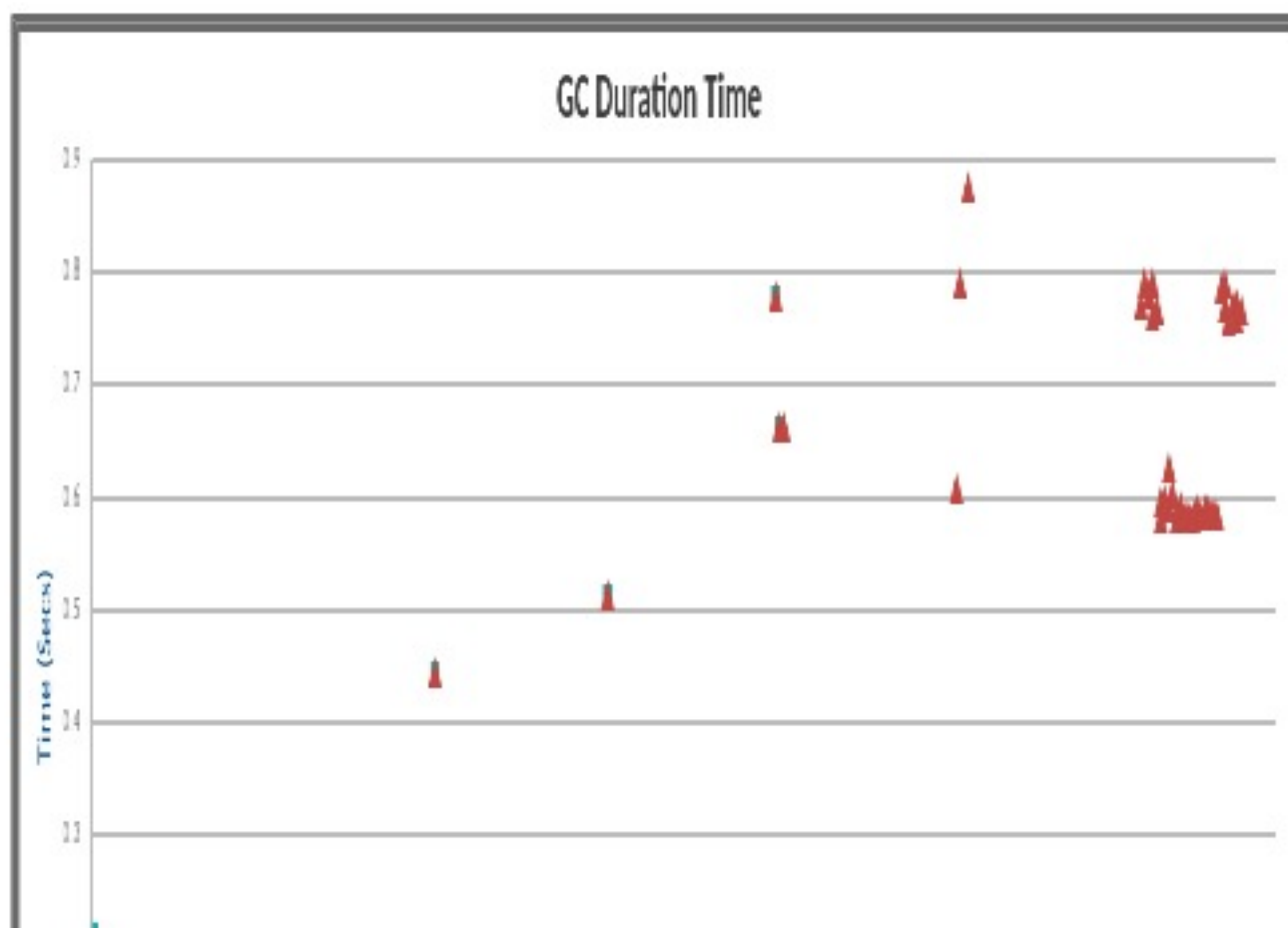
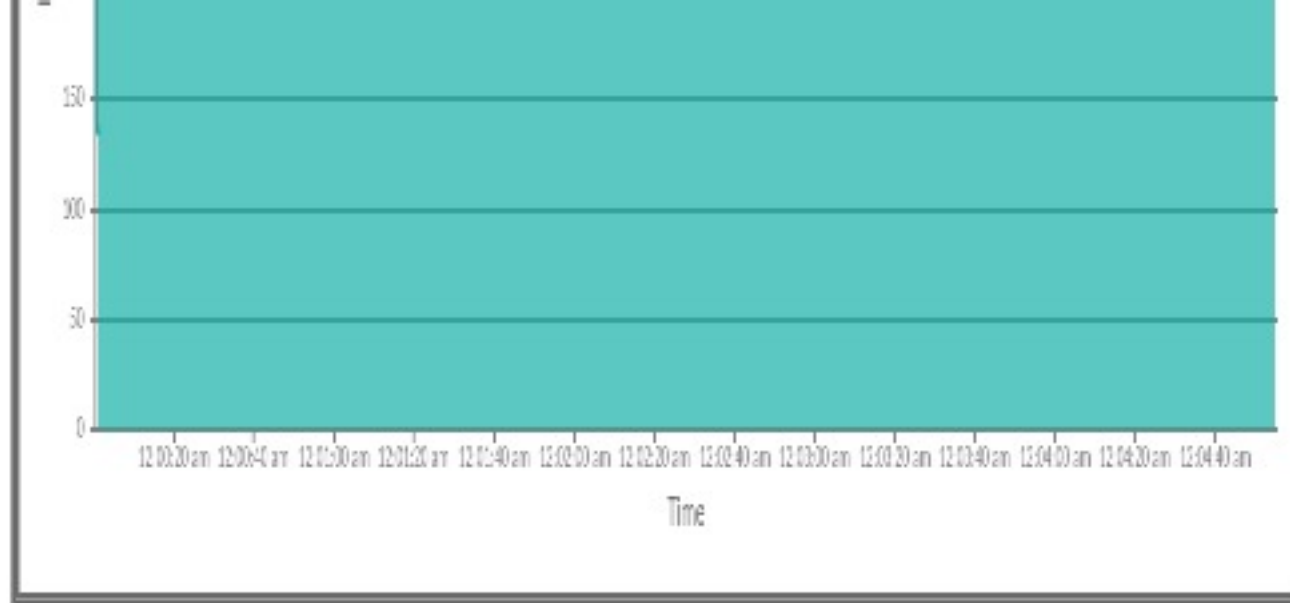
0.5 - 0.6	25	57.143%
0.6 - 0.7	6	67.857%
0.7 - 0.8	17	98.214%
0.8 - 0.9	1	100.0%

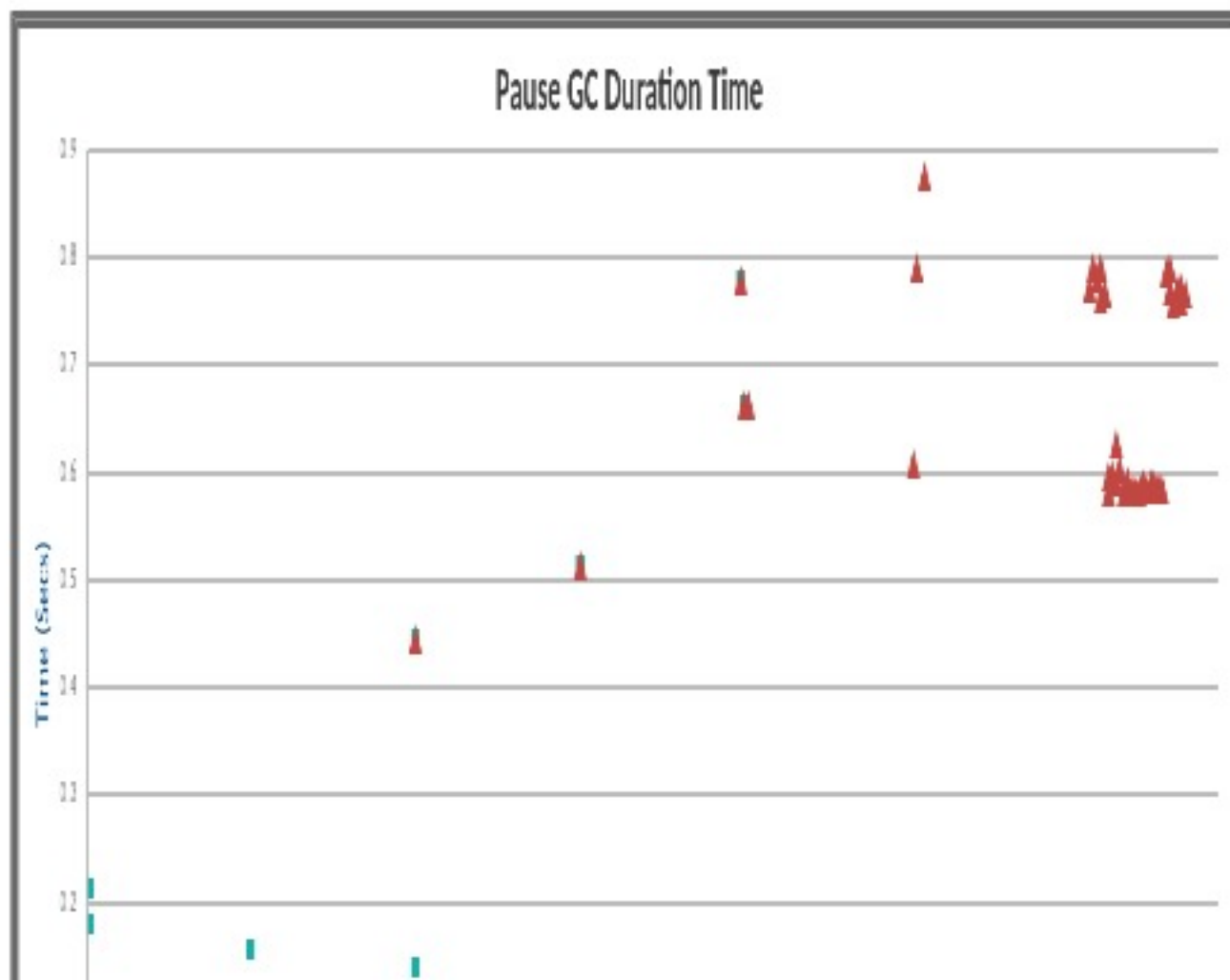
Interactive Graphs

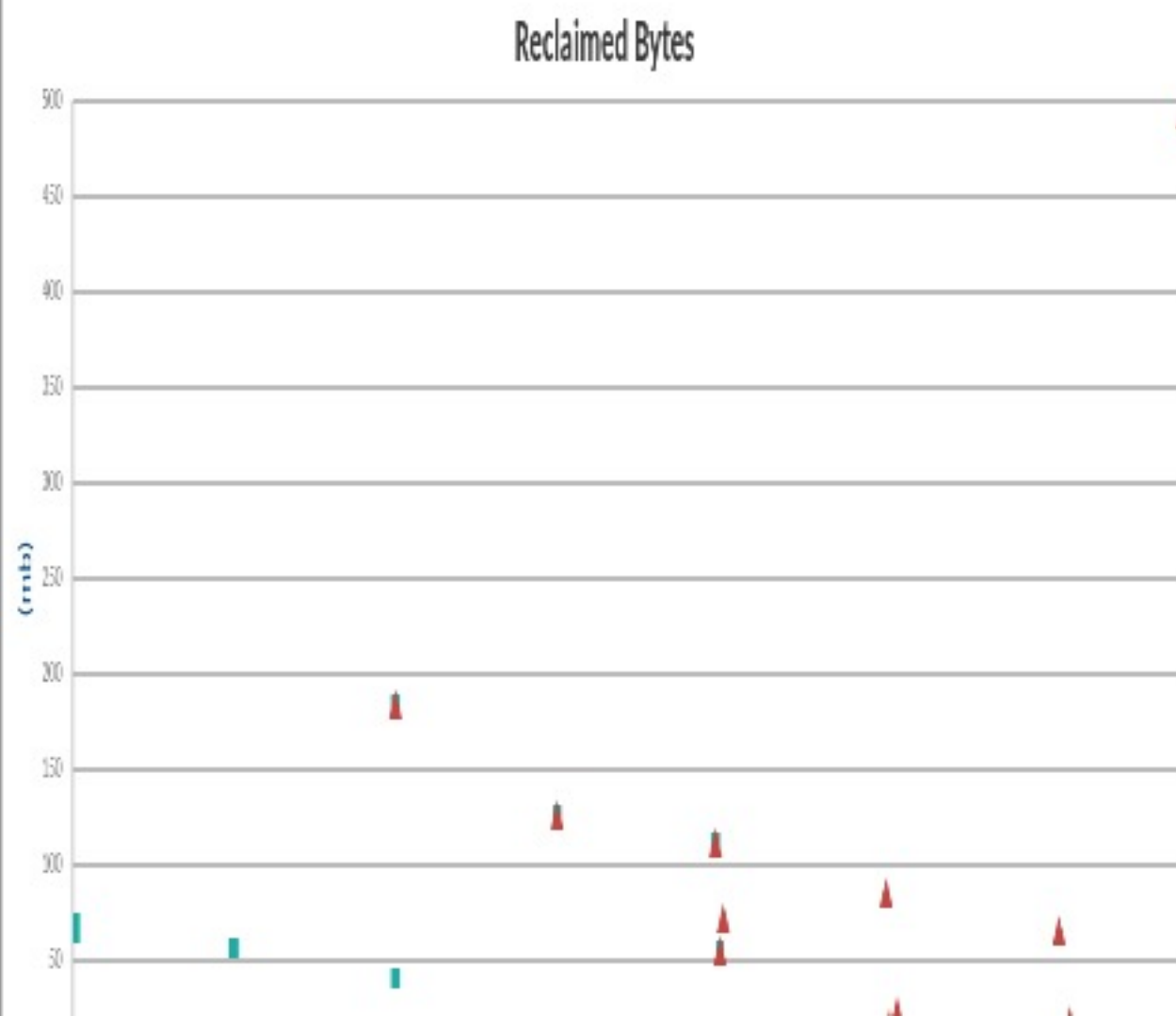
(All graphs are zoomable)

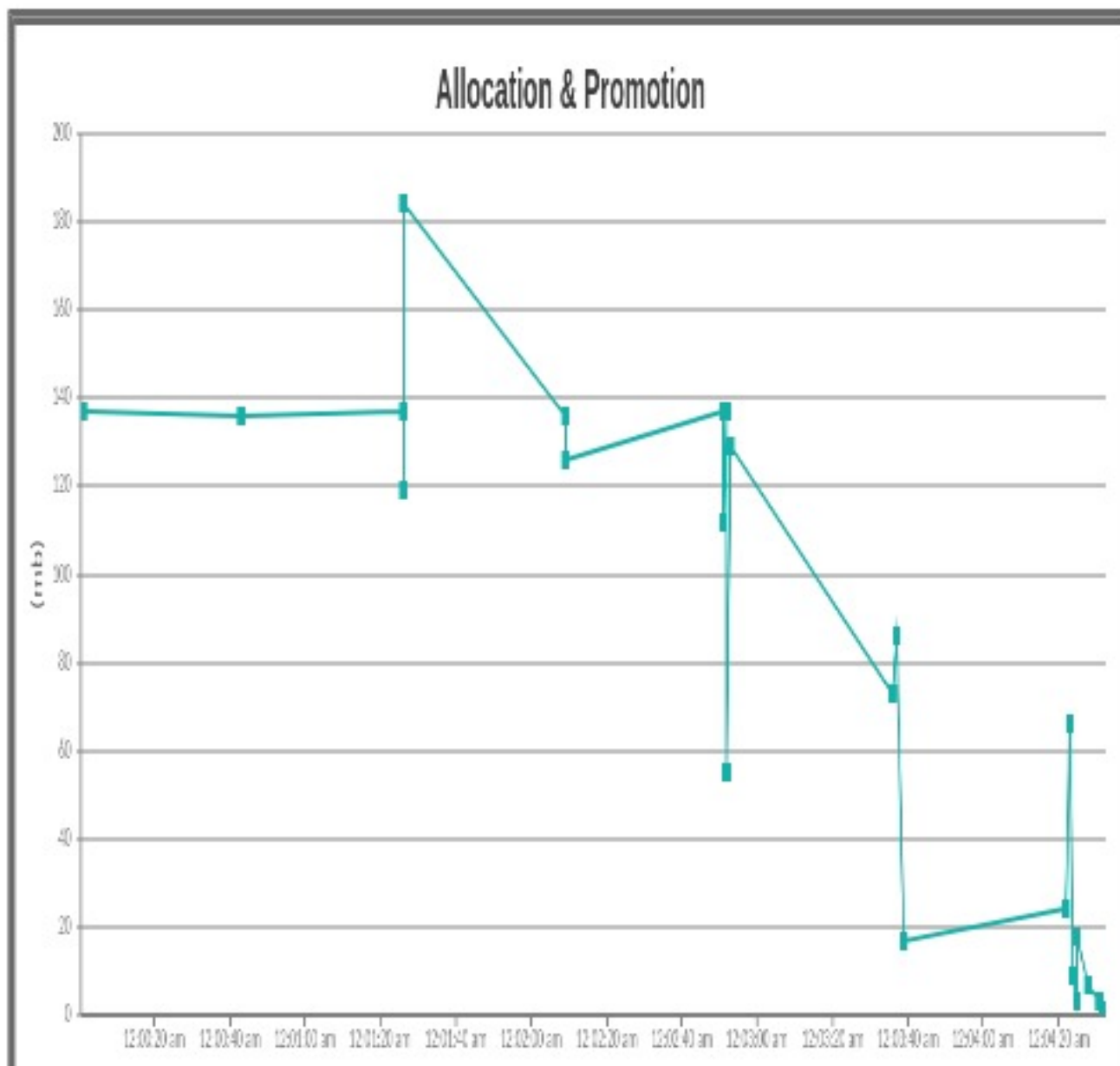








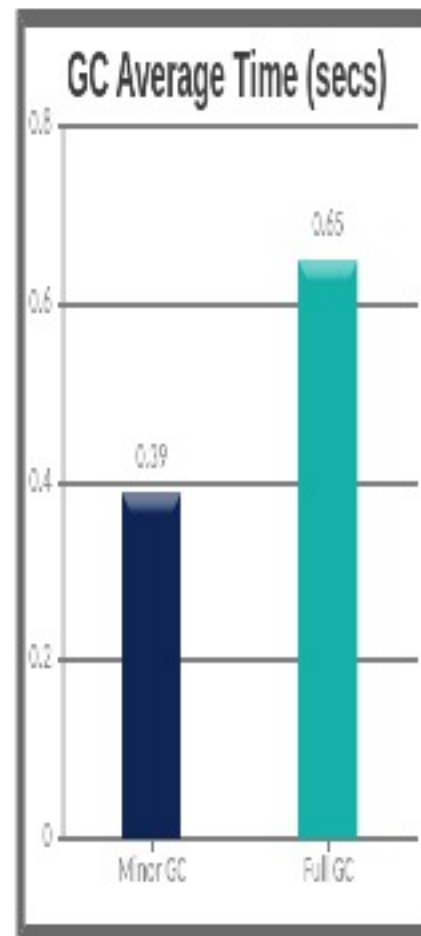
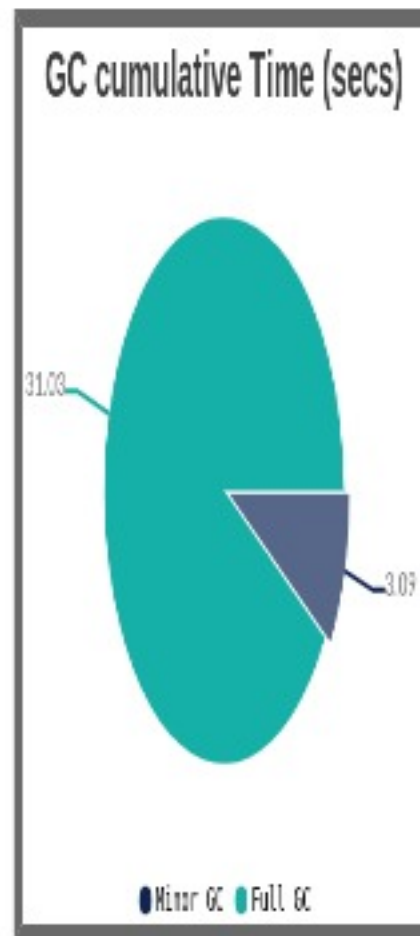
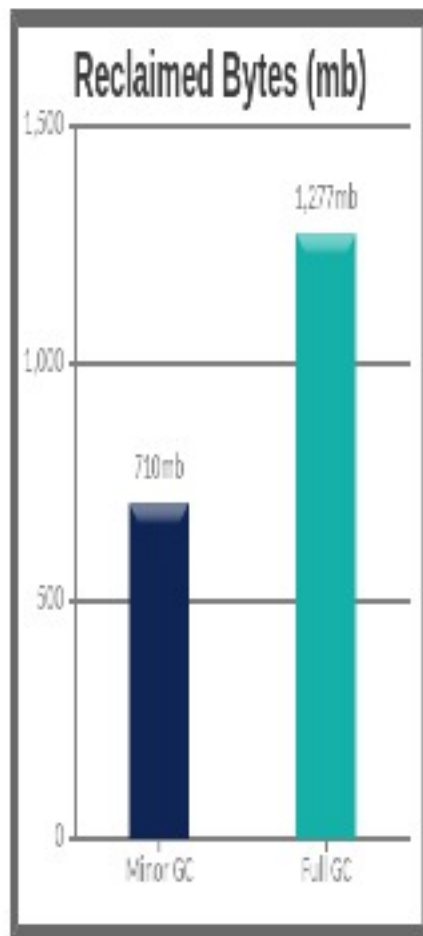




Time

■ Allocated objects size ■ Promoted (Young -> Old) objects size

GC Statistics ?



Total GC stats

Total GC count ⓘ	56
Total reclaimed bytes ⓘ	1.94 gb
Total GC time ⓘ	34 sec 122 ms
Avg GC time ⓘ	609 ms
GC avg time std dev	172 ms
GC min/max time	106 ms / 876 ms
GC Interval avg time ⓘ	5 sec 341 ms

Minor GC stats

Minor GC count	8
Minor GC reclaimed ⓘ	710 mb
Minor GC total time	3 sec 92 ms
Minor GC avg time ⓘ	387 ms
Minor GC avg time std dev	234 ms
Minor GC min/max time	139 ms / 779 ms
Minor GC Interval avg ⓘ	24 sec 466 ms

Full GC stats

Full GC Count	48
Full GC reclaimed ⓘ	1.25 gb
Full GC total time	31 sec 30 ms
Full GC avg time ⓘ	646 ms
Full GC avg time std dev	126 ms
Full GC min/max time	106 ms / 876 ms
Full GC Interval avg ⓘ	4 sec 437 ms

GC Pause Statistics

Pause Count	56
Pause total time	34 sec 122 ms
Pause avg time ⓘ	609 ms
Pause avg time std dev	0.0

Pause min/max time	106 ms / 876 ms
--------------------	-----------------

⚙️ Object Stats

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

Total created bytes ⓘ	1.94 gb
Total promoted bytes ⓘ	n/a
Avg creation rate ⓘ	6.77 mb/sec
Avg promotion rate ⓘ	n/a

💧 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

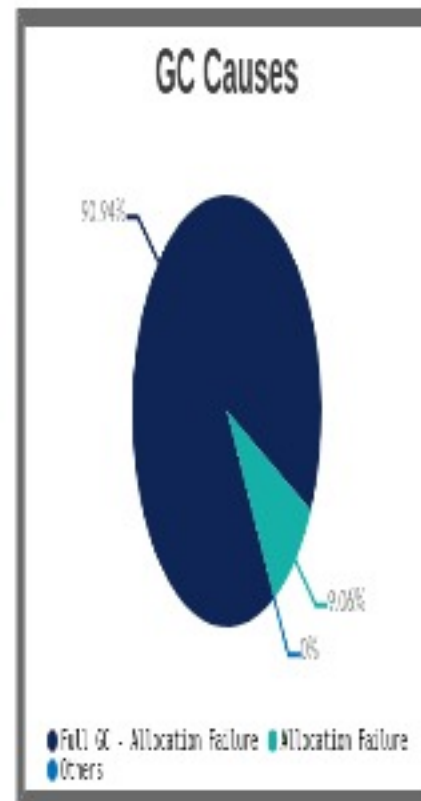
(To learn more about SafePoint duration, [click here](#))

Not Reported in the log.

GC Causes

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

Cause	Count	Avg Time	Max Time	Total Time	Time %
Full GC - Allocation Failure 🚫	48	646 ms	876 ms	31 sec 30 ms	90.94%
Allocation Failure 🚫	8	387 ms	779 ms	3 sec 92 ms	9.06%
Others	n/a	n/a	n/a	0	0.0%
Total	56	n/a	n/a	34 sec 122 ms	100.0%



🔄 Tenuring Summary 📄

Not reported in the log.

Command Line Flags

Not reported in the log.

