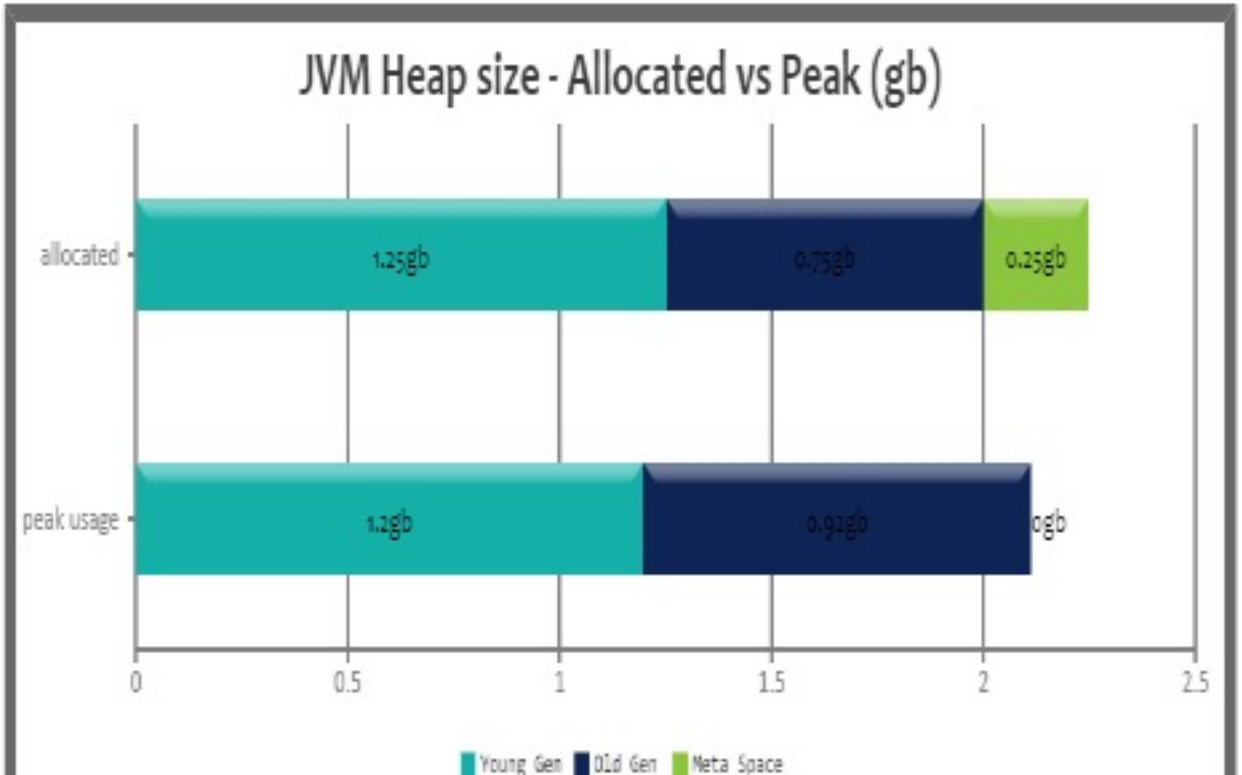
Analysis Report

Congratulations! Your application's GC activity is healthy.

JVM Heap Size

Generation	Allocated	Peak
Young Generation	1.25 gb	1.2 gb
Old Generation	767 mb	938.4 mb
Meta Space	256 mb	n/a
Young + Old + Meta space	2.25 gb	1.81 gb



Key Performance Indicators

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

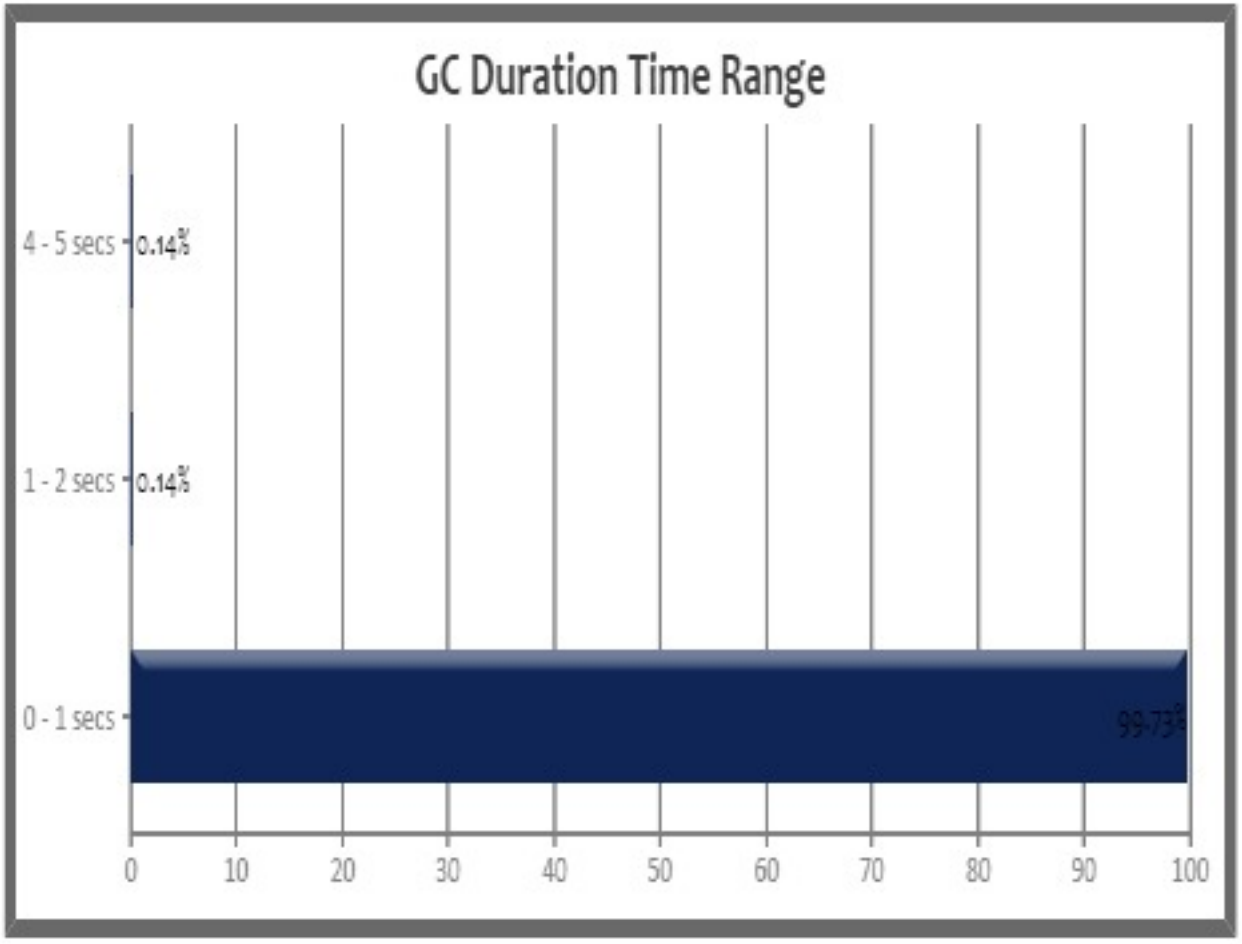
1. Throughput : 99.868%

2. Latency:

Avg Pause GC Time	142 ms
Max Pause GC Time	4 sec 100 ms

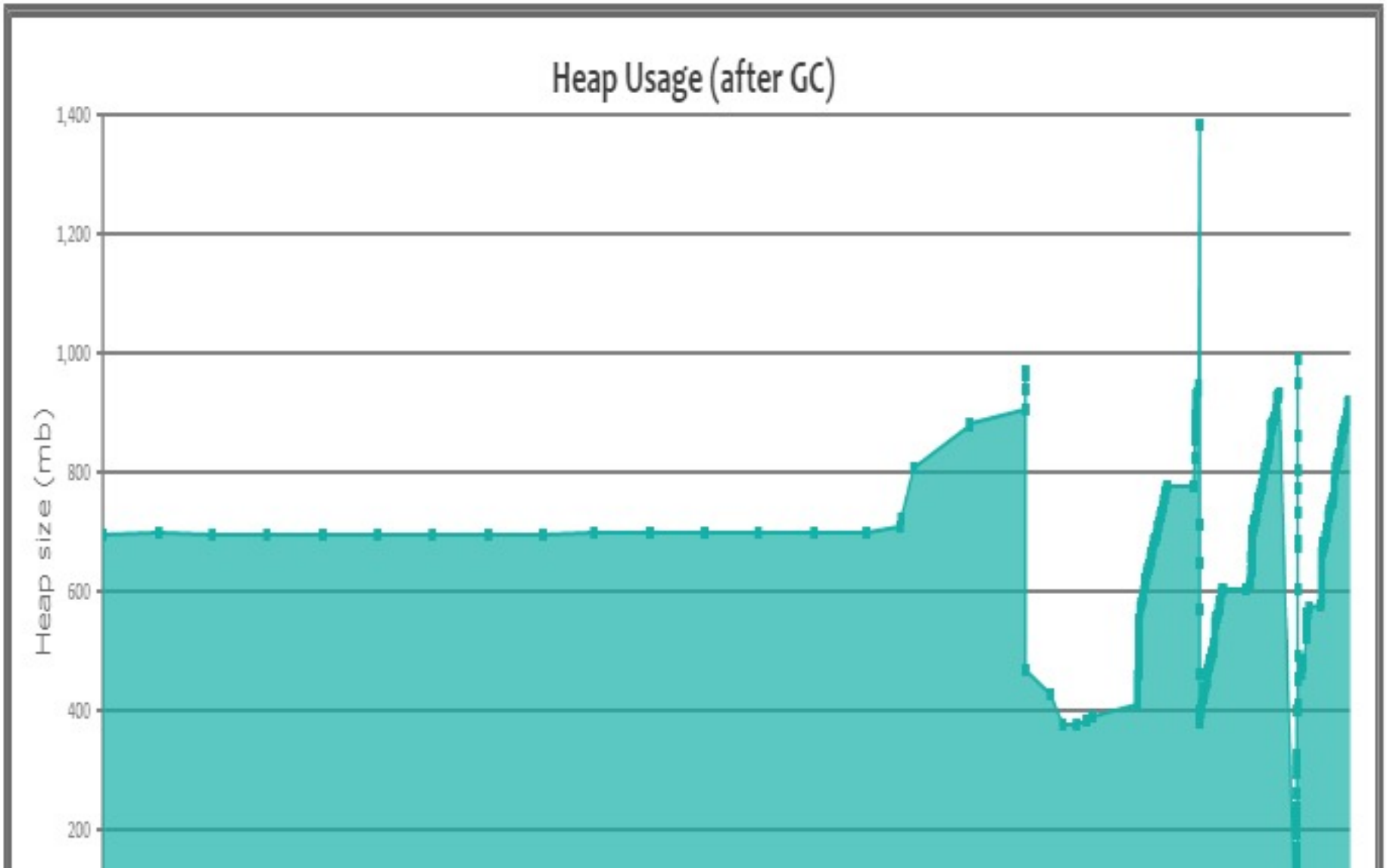
GC **Pause** Duration Time Range:

Duration (secs)	No. of GCs	Percentage
0 - 1	738	99.73%
1 - 2	1	99.865%
4 - 5	1	100.0%

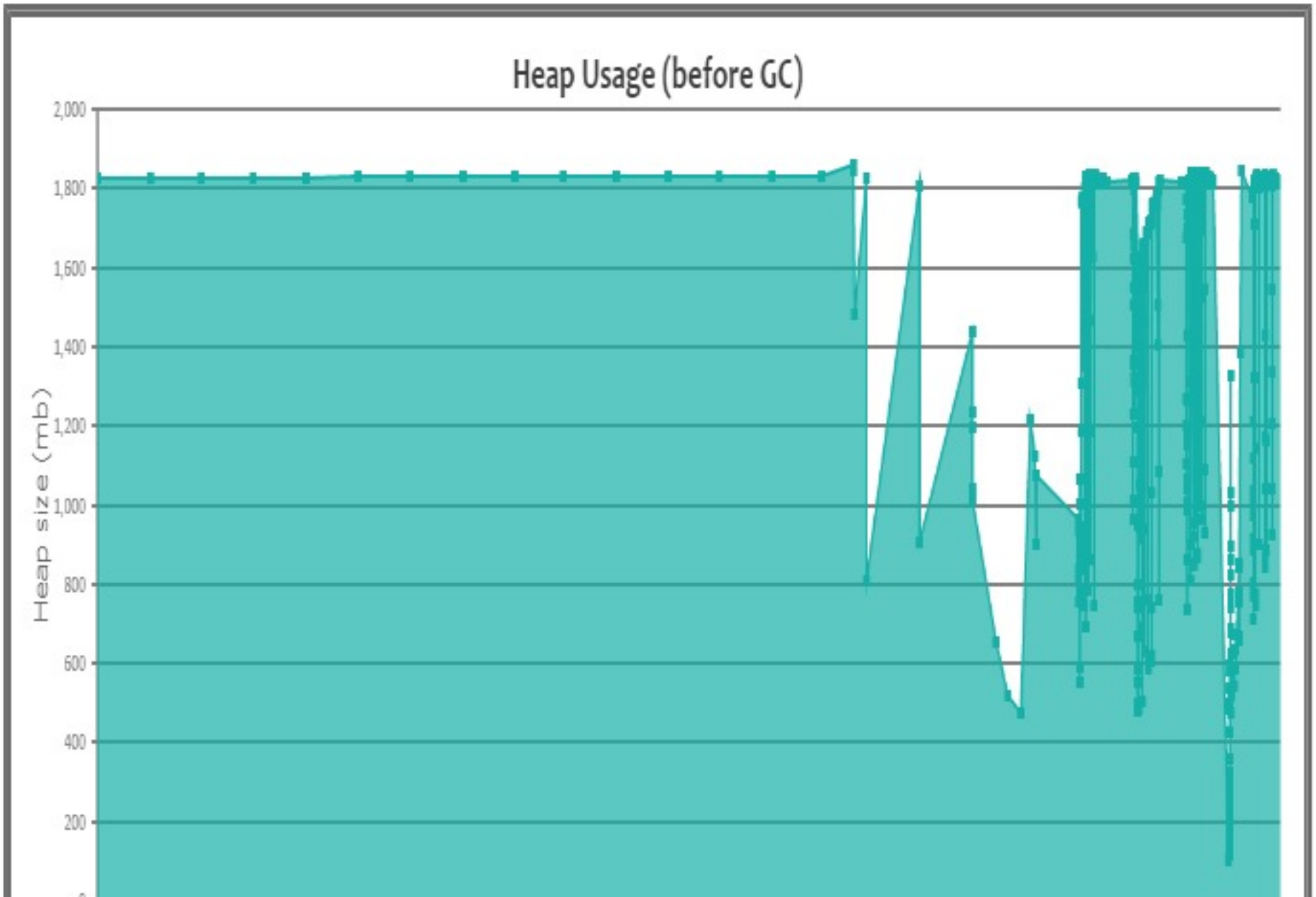


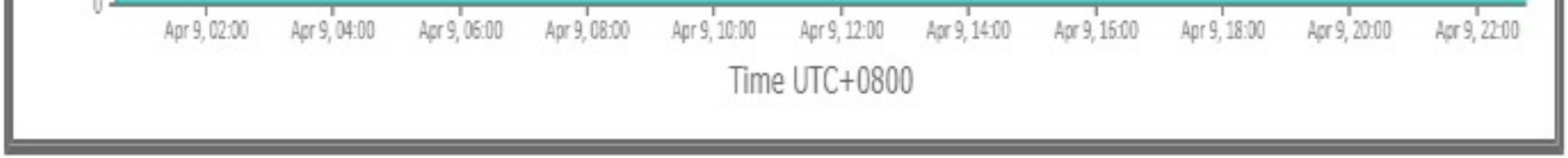
Interactive Graphs

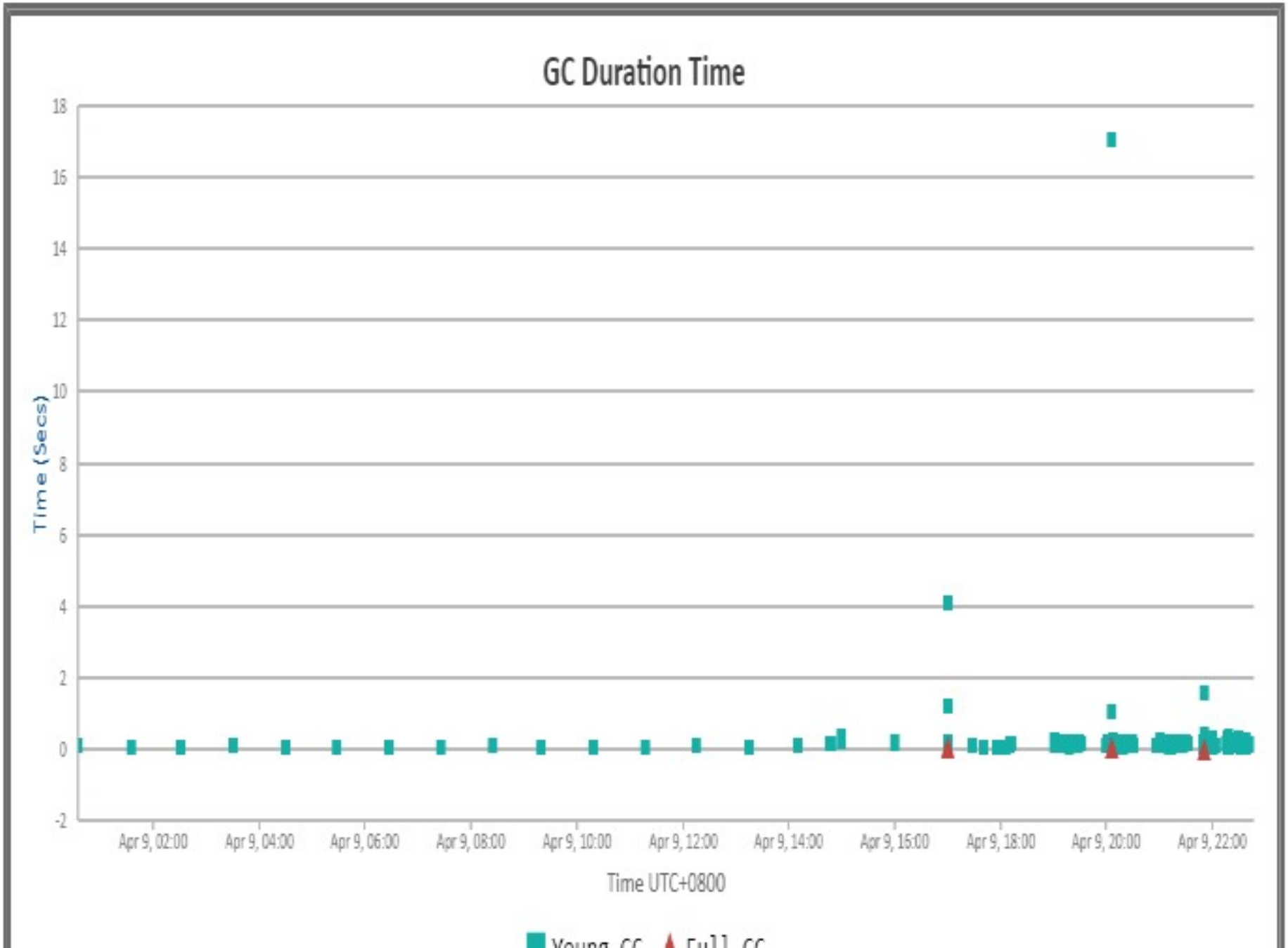
(All graphs are zoomable)

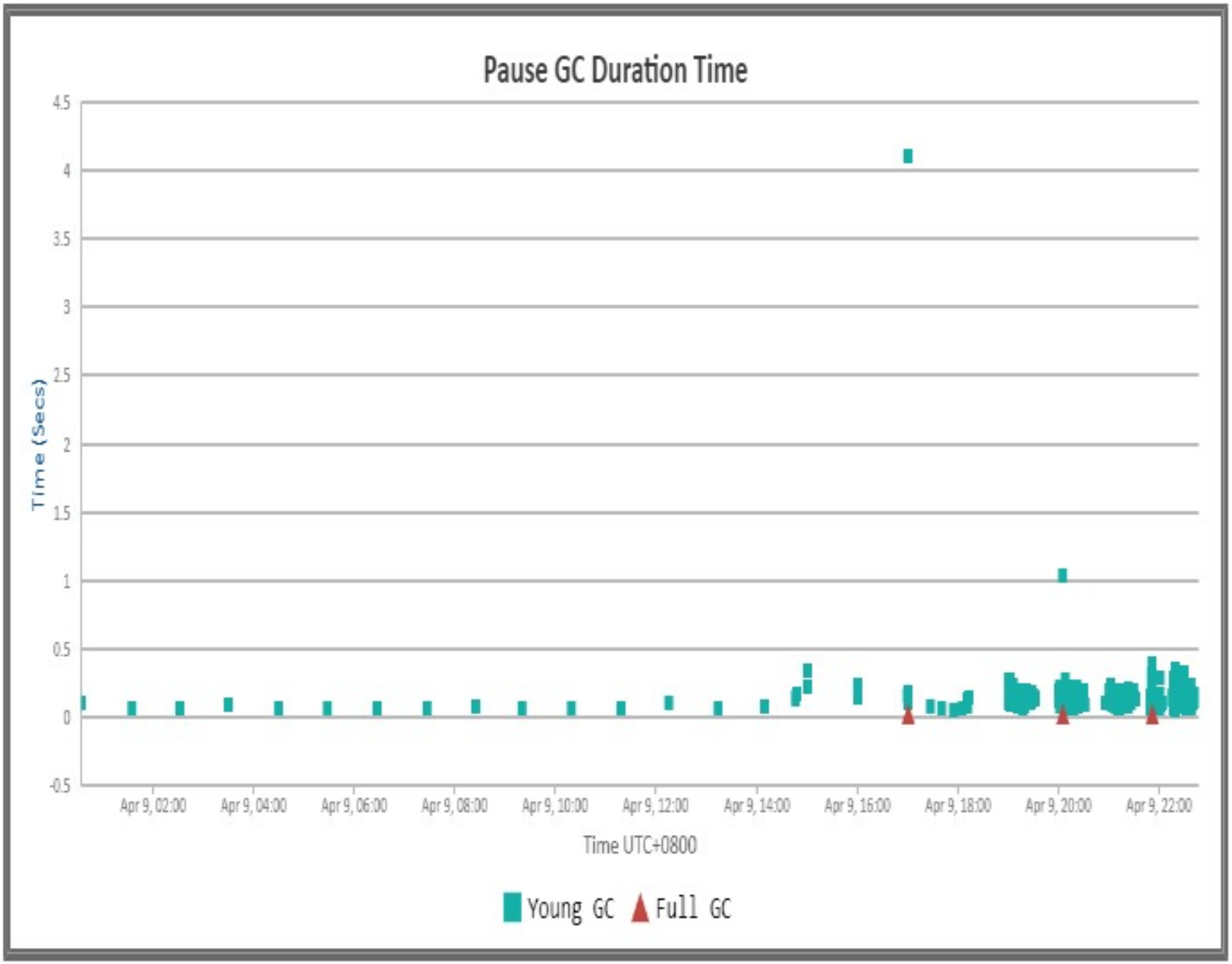


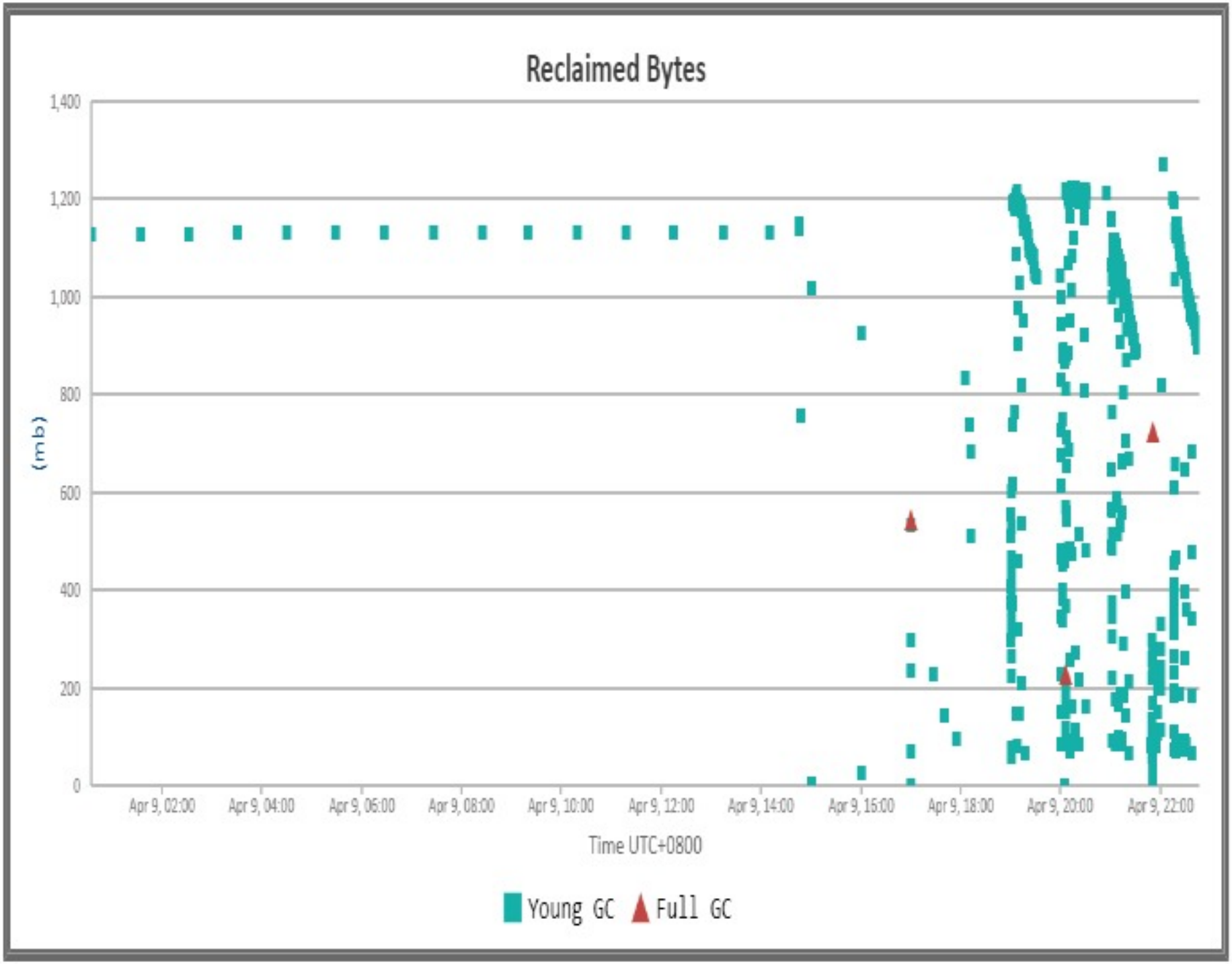

0
Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800

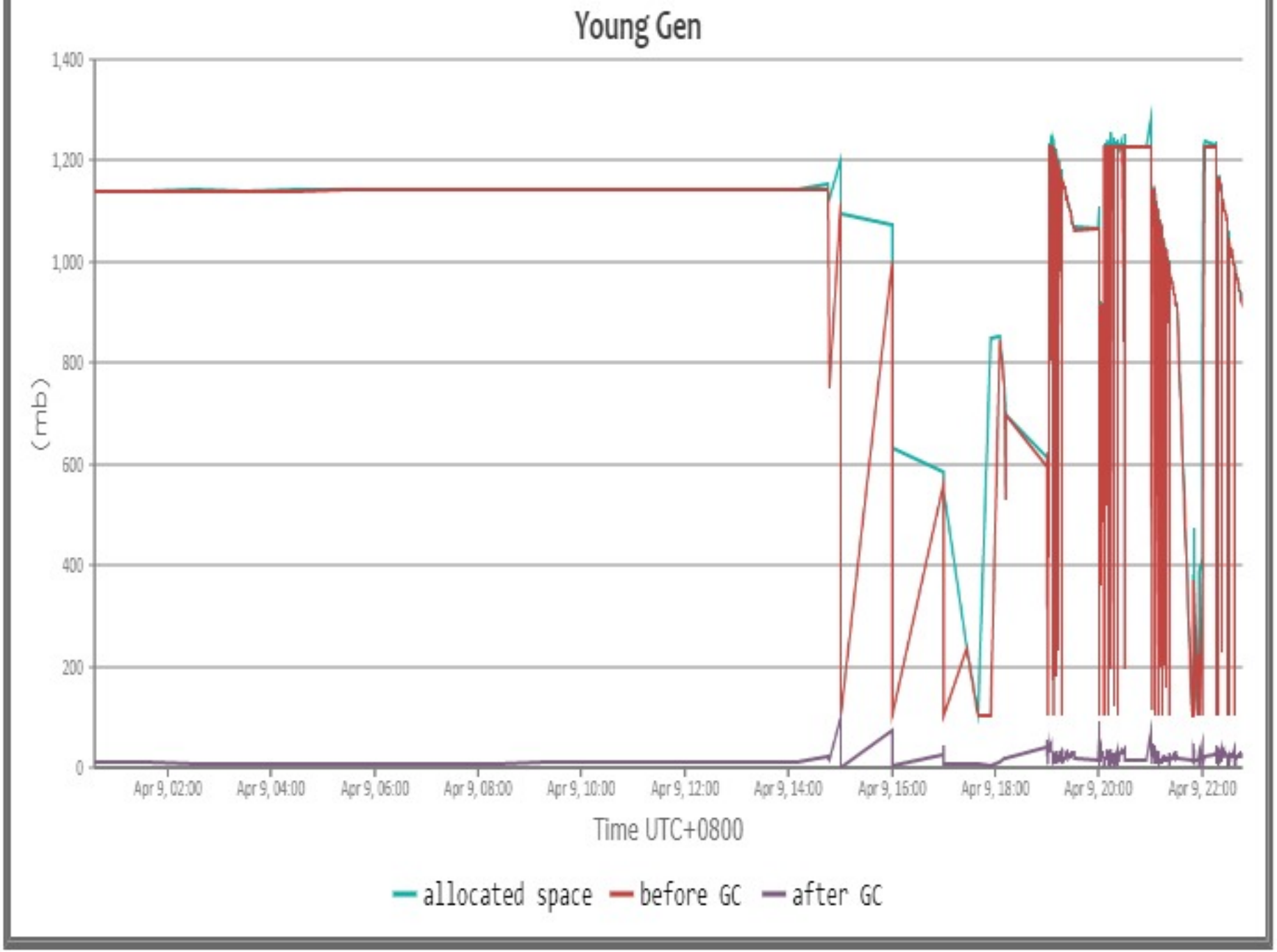

Heap Usage (before GC)
Heap size (mb)
2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200


Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800

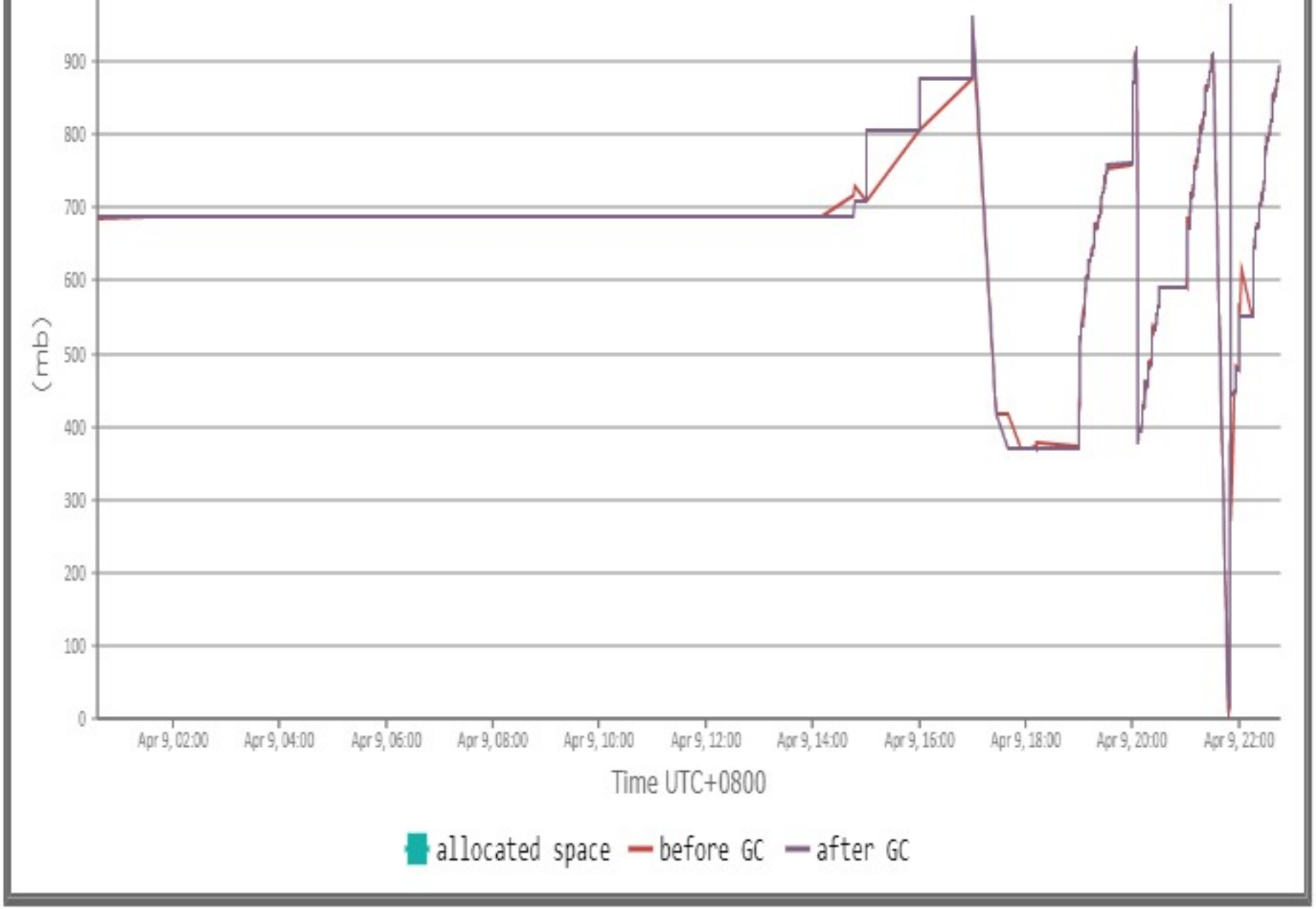

GC Duration Time
Time (Secs)
18
16
14
12
10
8
6
4
2
0
-2
Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800
Young GC
Full GC


Pause GC Duration Time
Time (Secs)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800
Young GC
Full GC

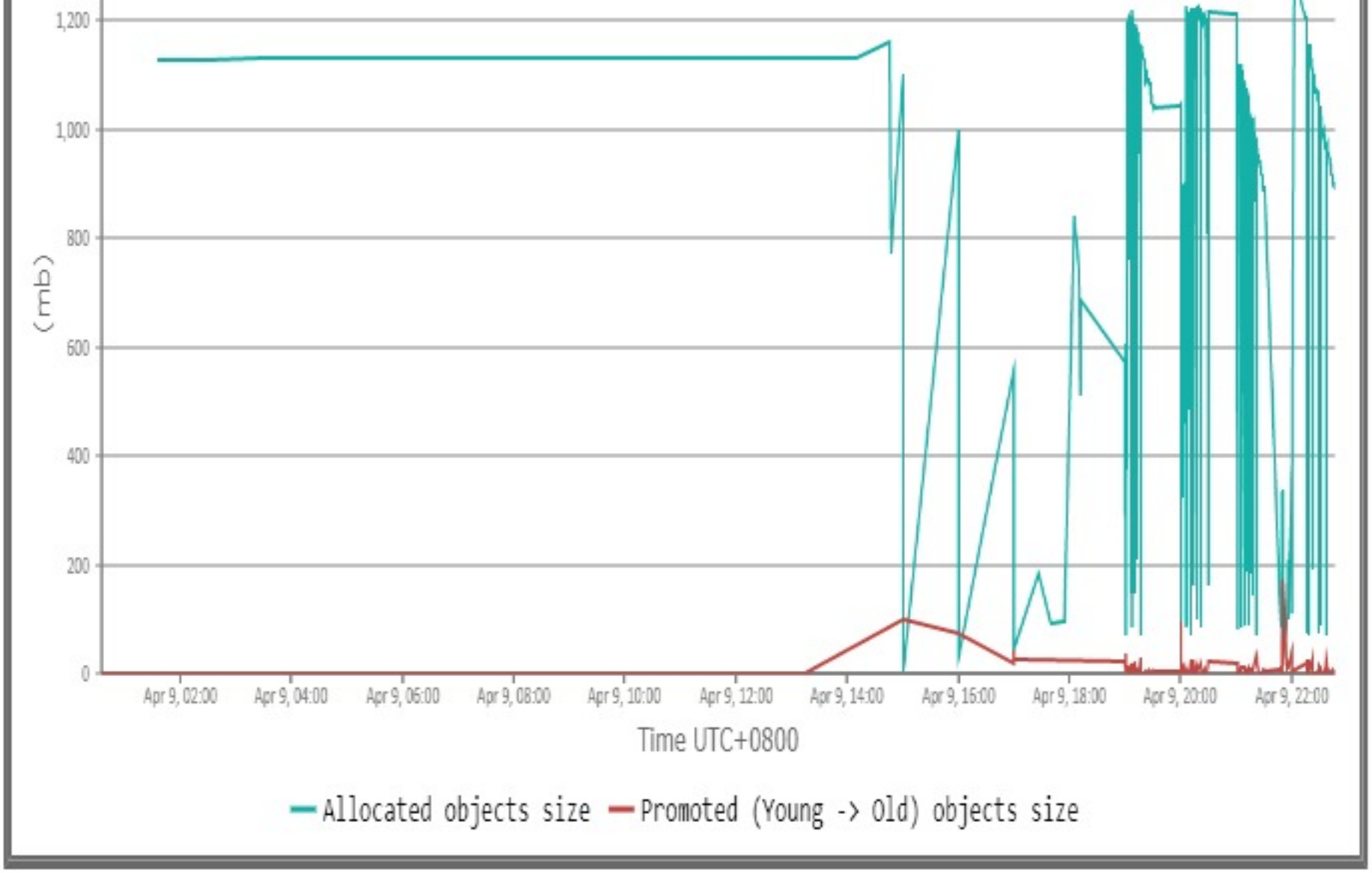

Reclaimed Bytes
1,400
1,200
1,000
800
600
400
200
0
(mb)
Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800
Young GC
Full GC


Young Gen
1,400
1,200
1,000
800
600
400
200
0
(mb)
Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800
allocated space
before GC
after GC

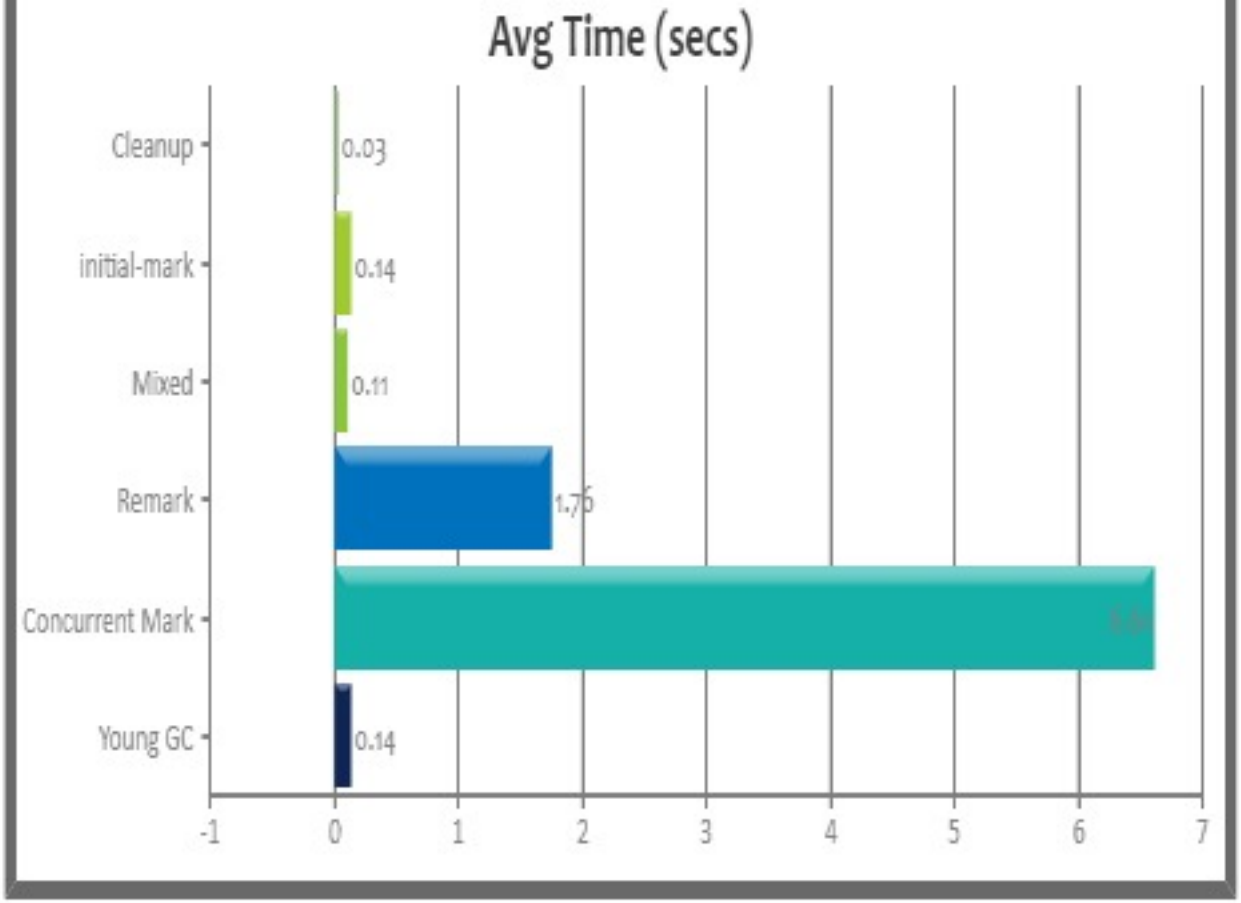

Old Gen
1,000

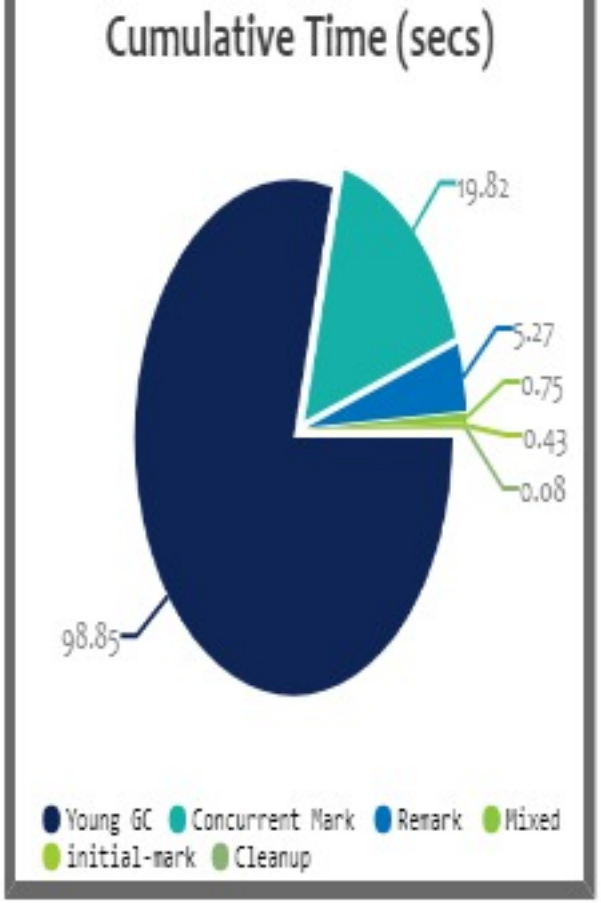

900
800
700
600
500
400
300
200
100
0
(mb)
Apr 9, 02:00
Apr 9, 04:00
Apr 9, 06:00
Apr 9, 08:00
Apr 9, 10:00
Apr 9, 12:00
Apr 9, 14:00
Apr 9, 16:00
Apr 9, 18:00
Apr 9, 20:00
Apr 9, 22:00
Time UTC+0800
allocated space
before GC
after GC


Allocation & Promotion
1,400



G1 Collection Phases Statistics

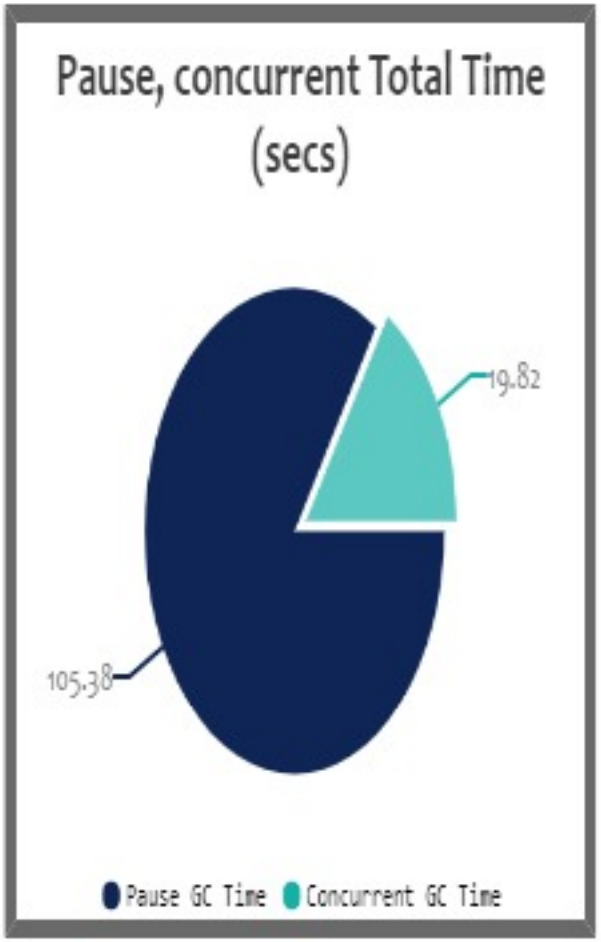


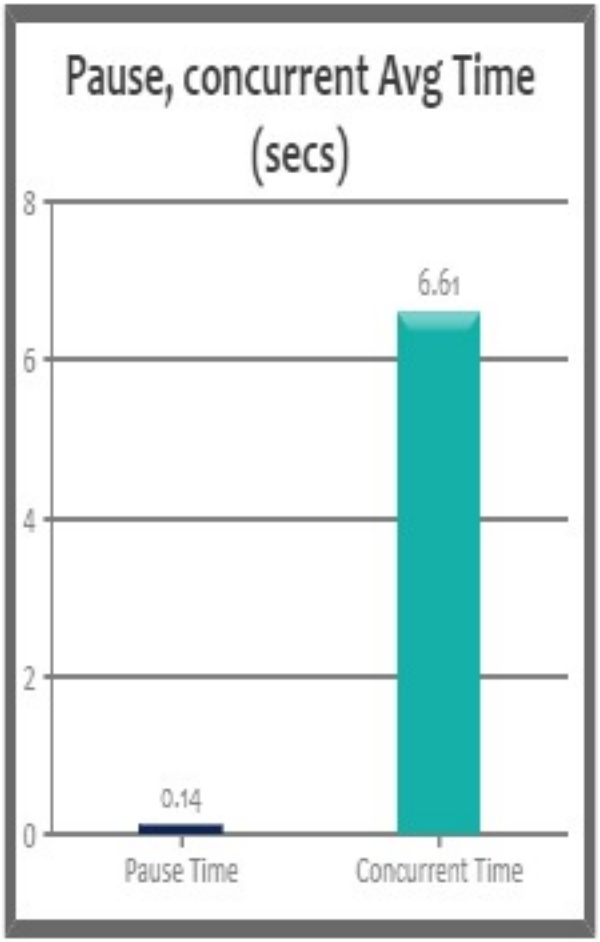


	Young GC	Concurrent Mark	Remark	Mixed	initial-mark	Cleanup	Total
Count	724	3	3	7	3	3	743
Total GC Time	1 min 38 sec 850 ms	19 sec 825 ms	5 sec 270 ms	750 ms	430 ms	80 ms	2 min 5 sec 205 ms
Avg GC Time	137 ms	6 sec 608 ms	1 sec 757 ms	107 ms	143 ms	27 ms	169 ms
Avg Time std dev	43 ms	7 sec 382 ms	1 sec 698 ms	35 ms	29 ms	5 ms	642 ms

Min/Max Time	0 7 390 ms	0 7 17 sec 47 ms	0 7 4 sec 100 ms	0 7 150 ms	0 7 180 ms	0 7 30 ms	0 7 17 sec 47 ms
Avg Interval Time	1 min 50 sec 525 ms	2 hrs 25 min 22 sec	2 hrs 25 min 22 sec	24 min 14 sec 470 ms	2 hrs 25 min 21 sec	2 hrs 25 min 20 sec	3 min 34 sec 943 ms

G1 GC Time





Pause Time

Total Time	1 min 45 sec 380 ms
Avg Time	142 ms
Std Dev Time	156 ms
Min Time	20 ms
Max Time	4 sec 100 ms

Concurrent Time

Total Time	19 sec 825 ms
Avg Time	6 sec 608 ms
Std Dev Time	7 sec 382 ms
Min Time	1 sec 217 ms
Max Time	17 sec 47 ms

Object Stats

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

Total created bytes	606.92 gb
Total promoted bytes	2.57 gb
Avg creation rate	7.78 mb/sec
Avg promotion rate	33 kb/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

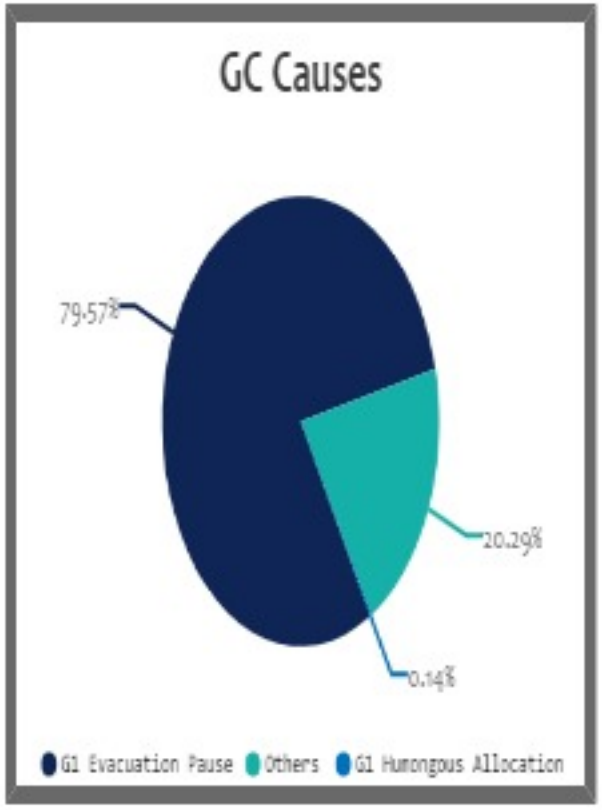
(To learn more about SafePoint duration, click here)

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 %
G1 Evacuation Pause	733	136 ms	393 ms	1 min 39 sec 625 ms	79.57%
Others	9	n/a	n/a	25 sec 401 ms	20.29%
G1 Humongous Allocation	1	179 ms	179 ms	179 ms	0.14%
Total	743	n/a	n/a	2 min 5 sec 205 ms	100.0%



Tenuring Summary

Not reported in the log.

Command Line Flags

-XX:CompressedClassSpaceSize=260046848 -XX:GCLogFileSize=3145728 -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/u03/home/csboa151/wildfly-12.0.0.Final/standalone/log/MemDump.hprof -XX:InitialHeapSize=2147483648 -XX:MaxHeapSize=2147483648 -XX:MaxMetaspaceSize=268435456 -XX:MetaspaceSize=268435456 -XX:NumberOfGCLogFiles=5 -XX:+PrintGC -XX:+PrintGCDateStamps -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:-TraceClassUnloading -XX:+UseCompressedClassPointers -XX:+UseCompressedOops -XX:+UseG1GC -XX:+UseGCLogFileRotation