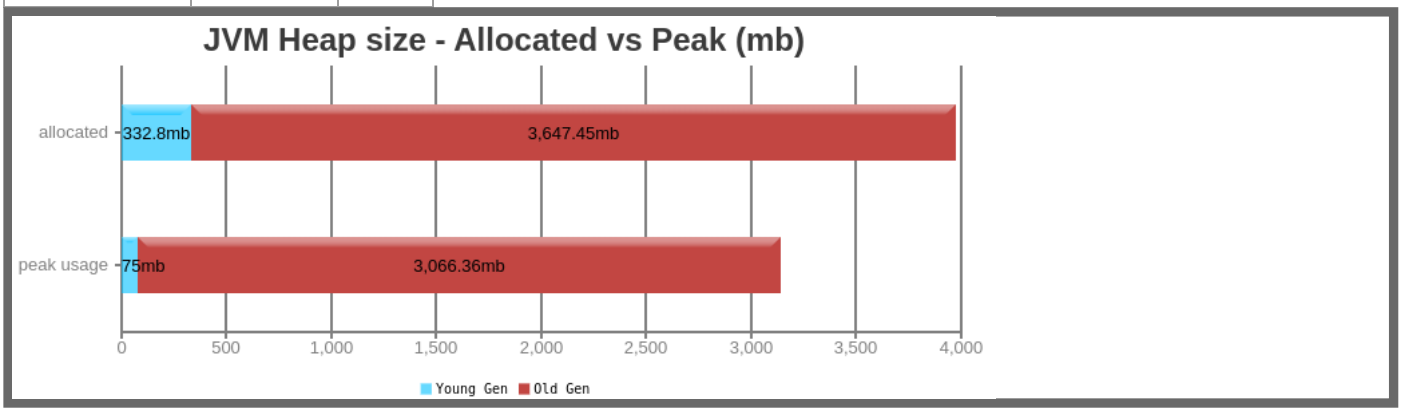
Analysis Report

JVM Heap Size

Generation	Allocated	Peak
Young Generation	332.8 mb	75 mb
Old Generation	3.56 gb	2.99 gb
Total	3.89 gb	3.07 gb



Key Performance Indicators

(Important section of the report. To learn more about KPIs, click here (https://blog.gceasy.io/2016/10/01/garbage-collection-kpi/))

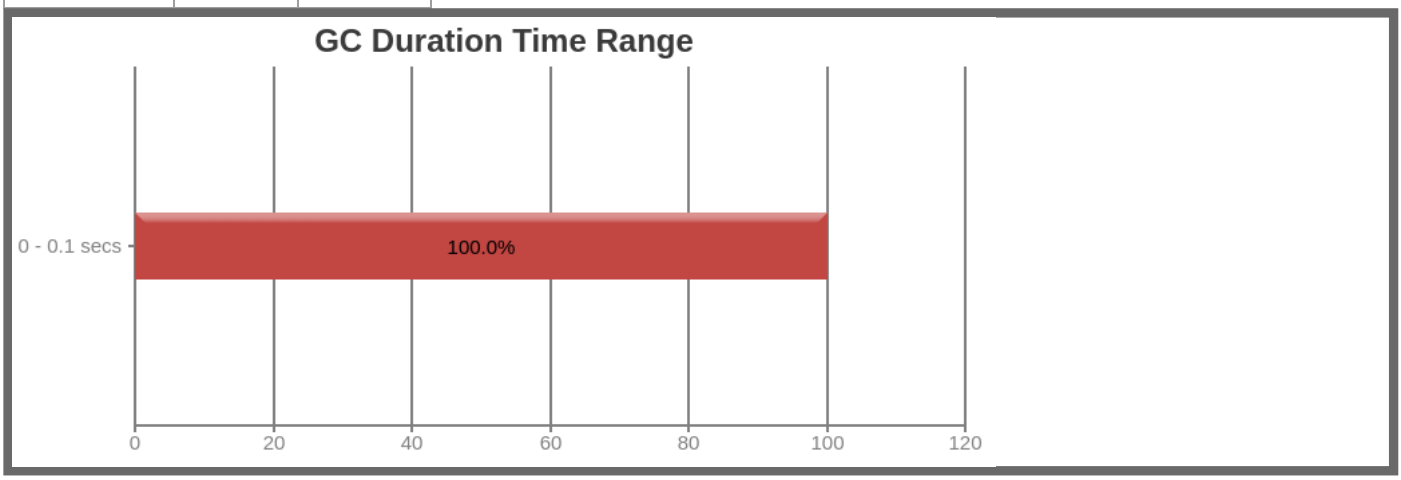
1. **Throughput : 58.898%**

2. **Latency:**

Avg Pause GC Time	**17 ms**
Max Pause GC Time	**40 ms**

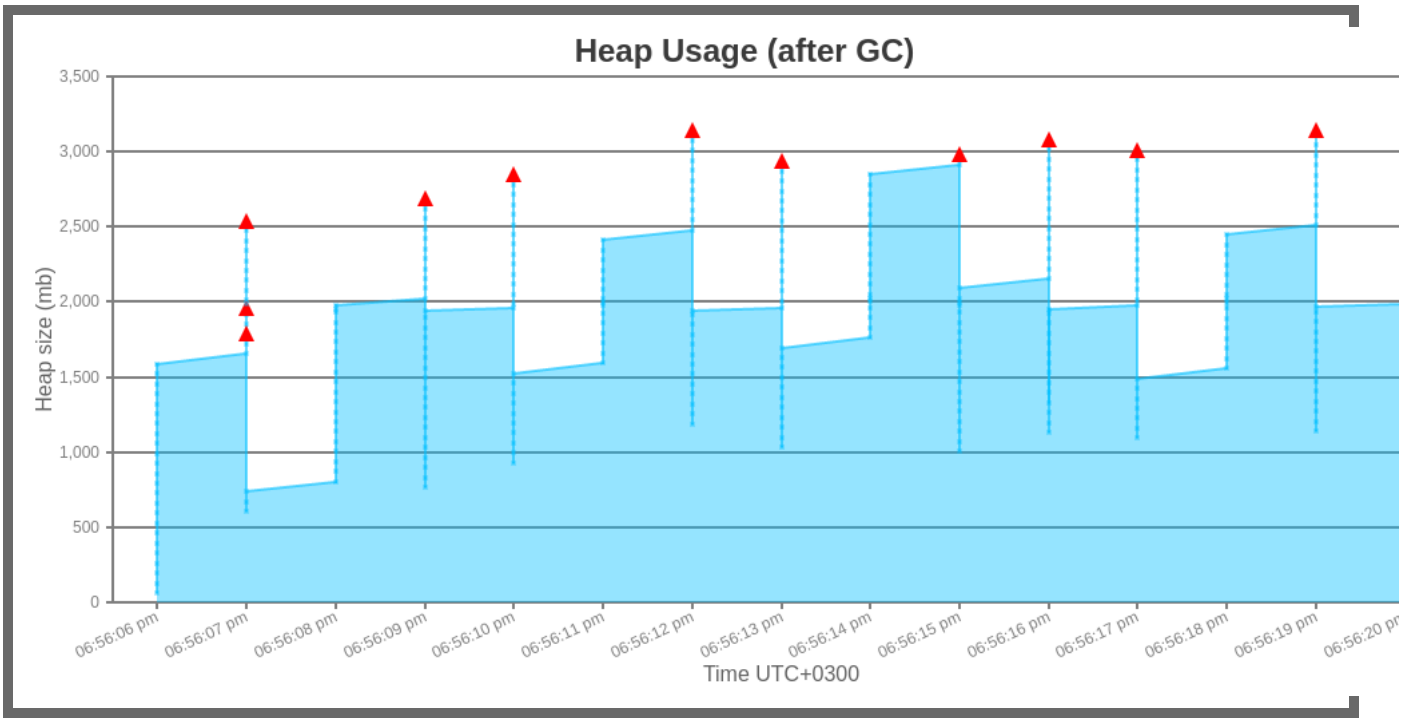
GC **Pause** Duration Time Range:

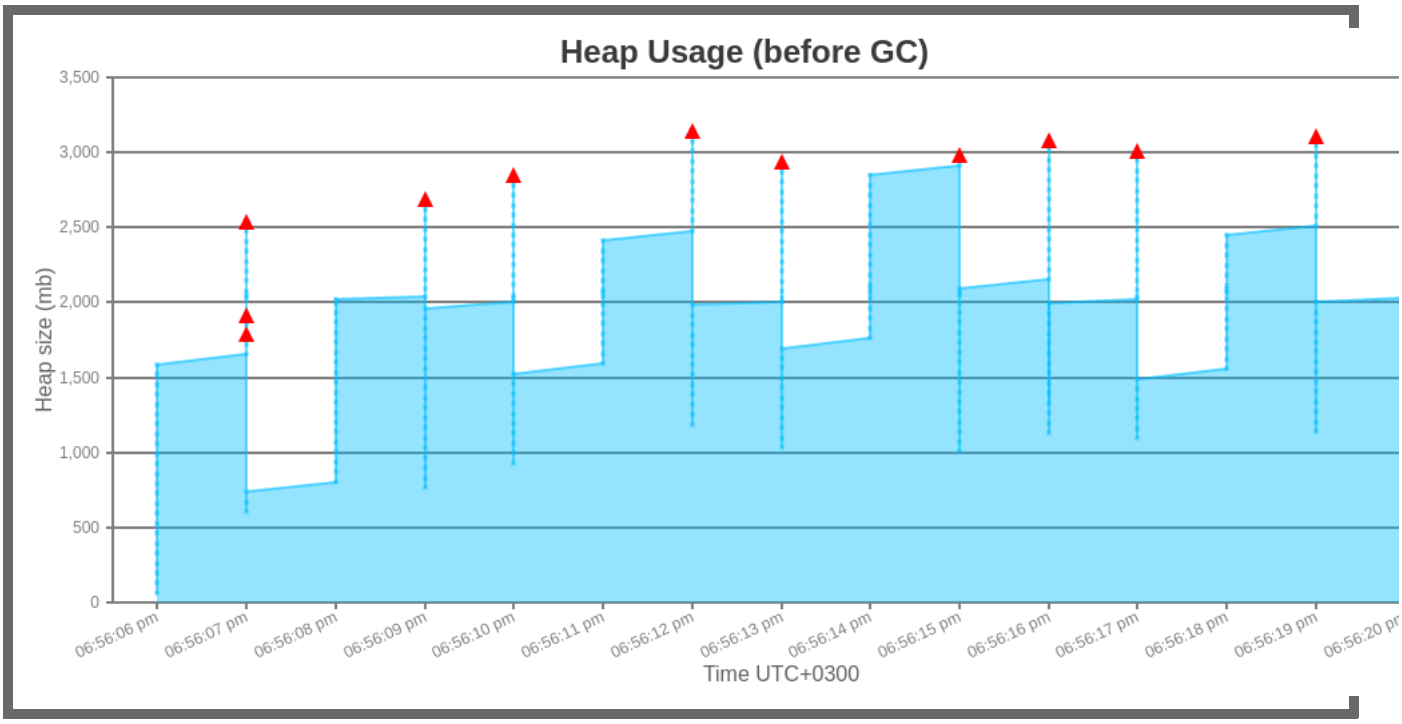
Duration (secs)	No. of GCs	Percentage
0 - 0.1	319	100.0%

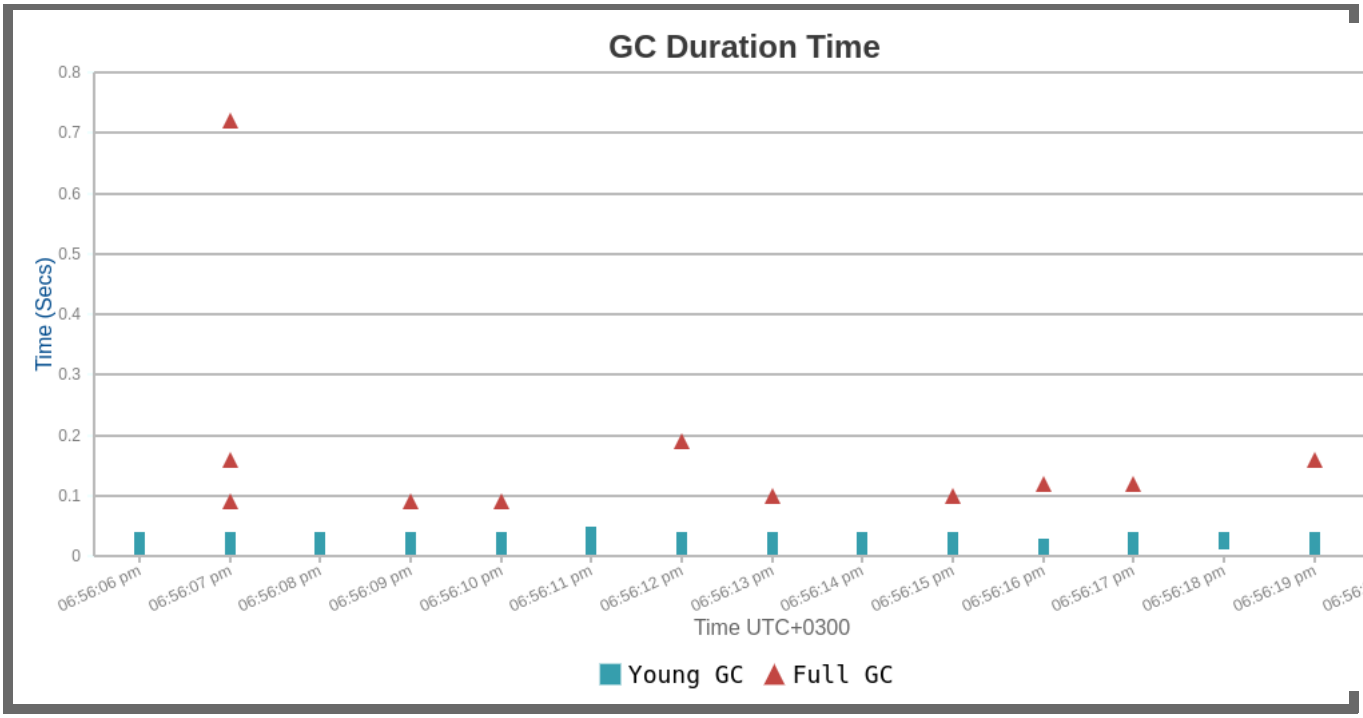


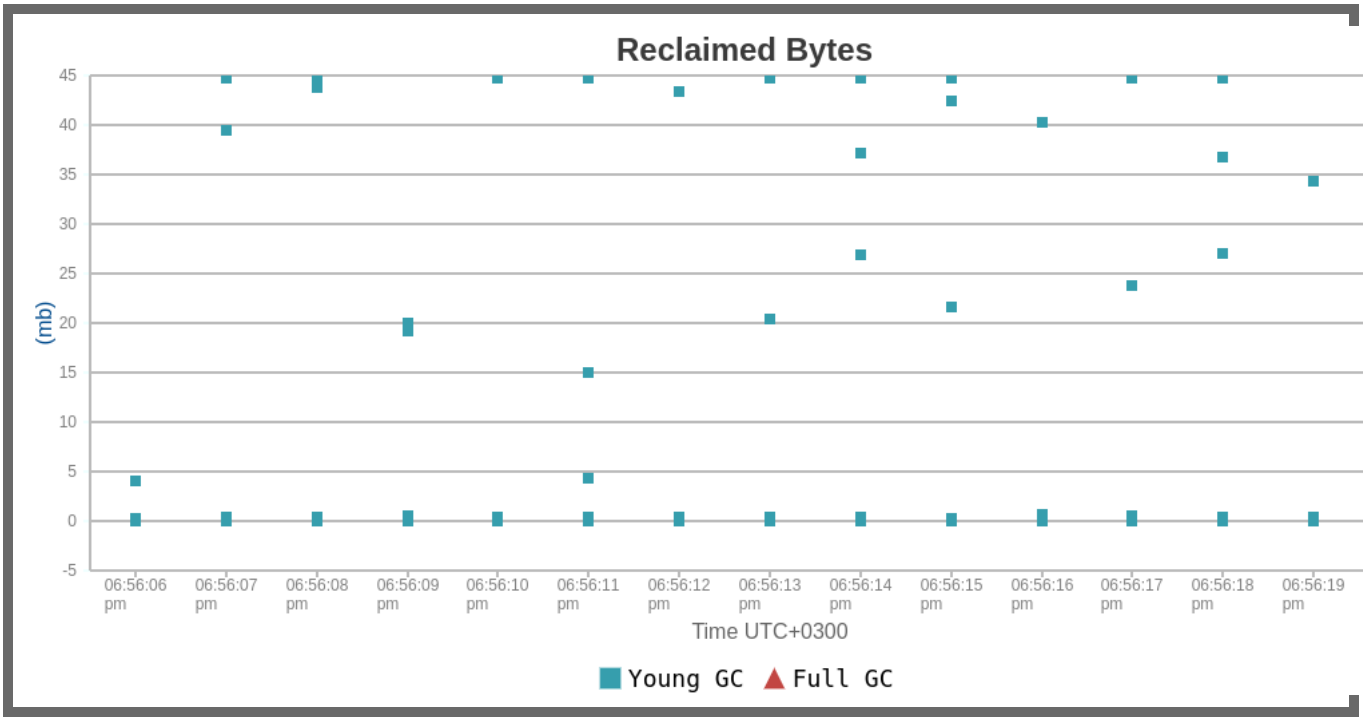
Interactive Graphs

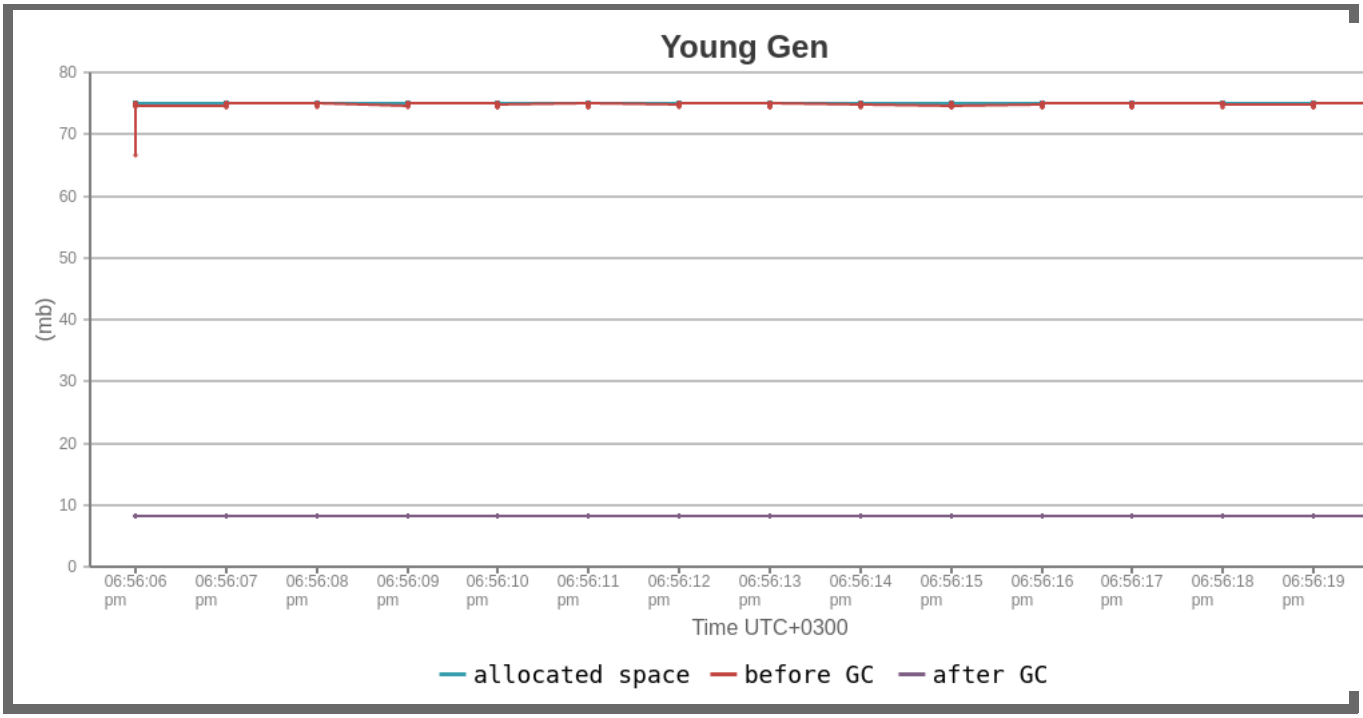
(All graphs are zoomable)

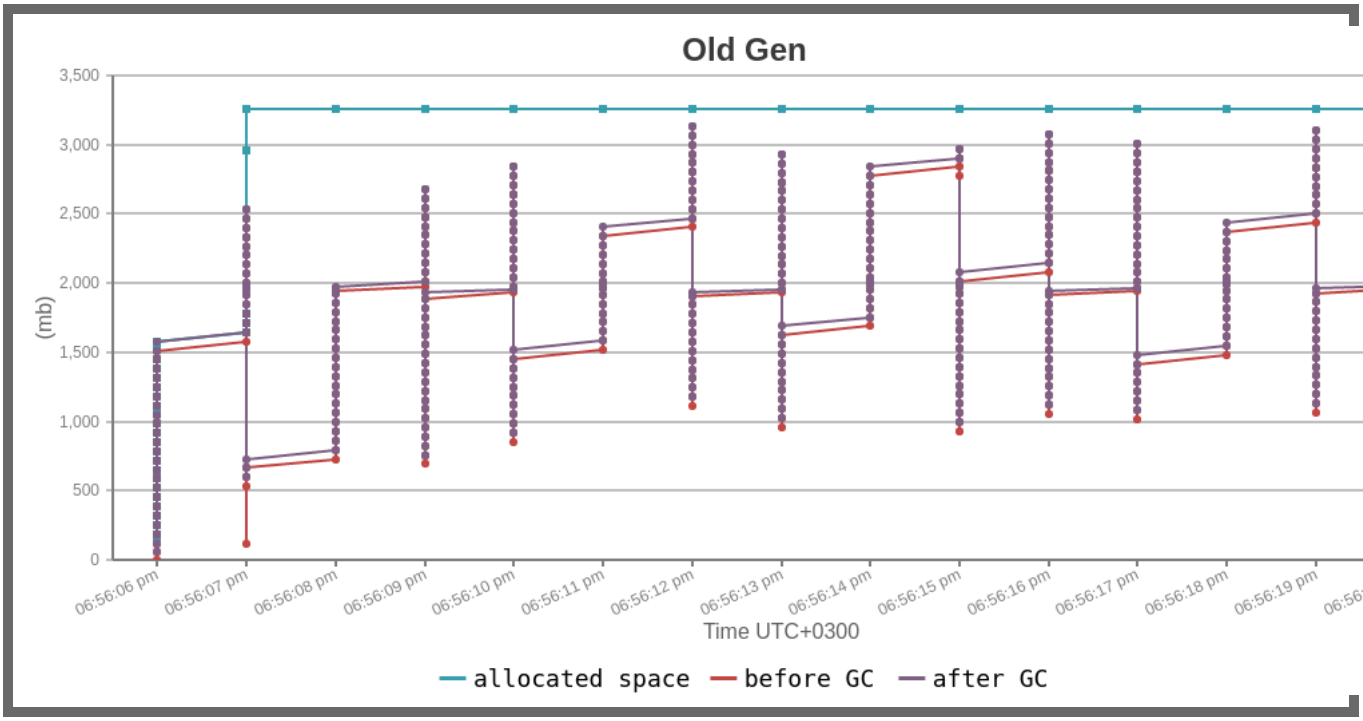


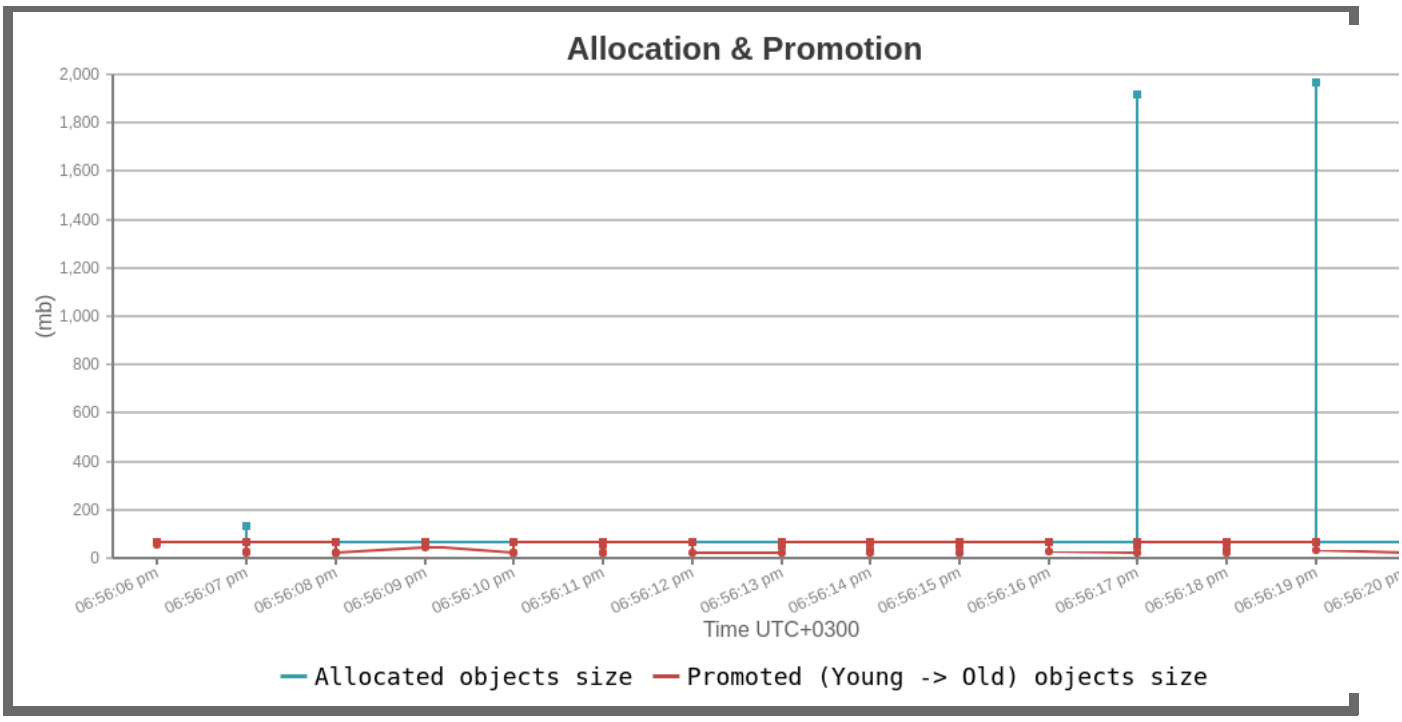



GC Duration Time
Time (Secs)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
06:56:06 pm
06:56:07 pm
06:56:08 pm
06:56:09 pm
06:56:10 pm
06:56:11 pm
06:56:12 pm
06:56:13 pm
06:56:14 pm
06:56:15 pm
06:56:16 pm
06:56:17 pm
06:56:18 pm
06:56:19 pm
06:56:20 pm
Time UTC+0300
Young GC
Full GC

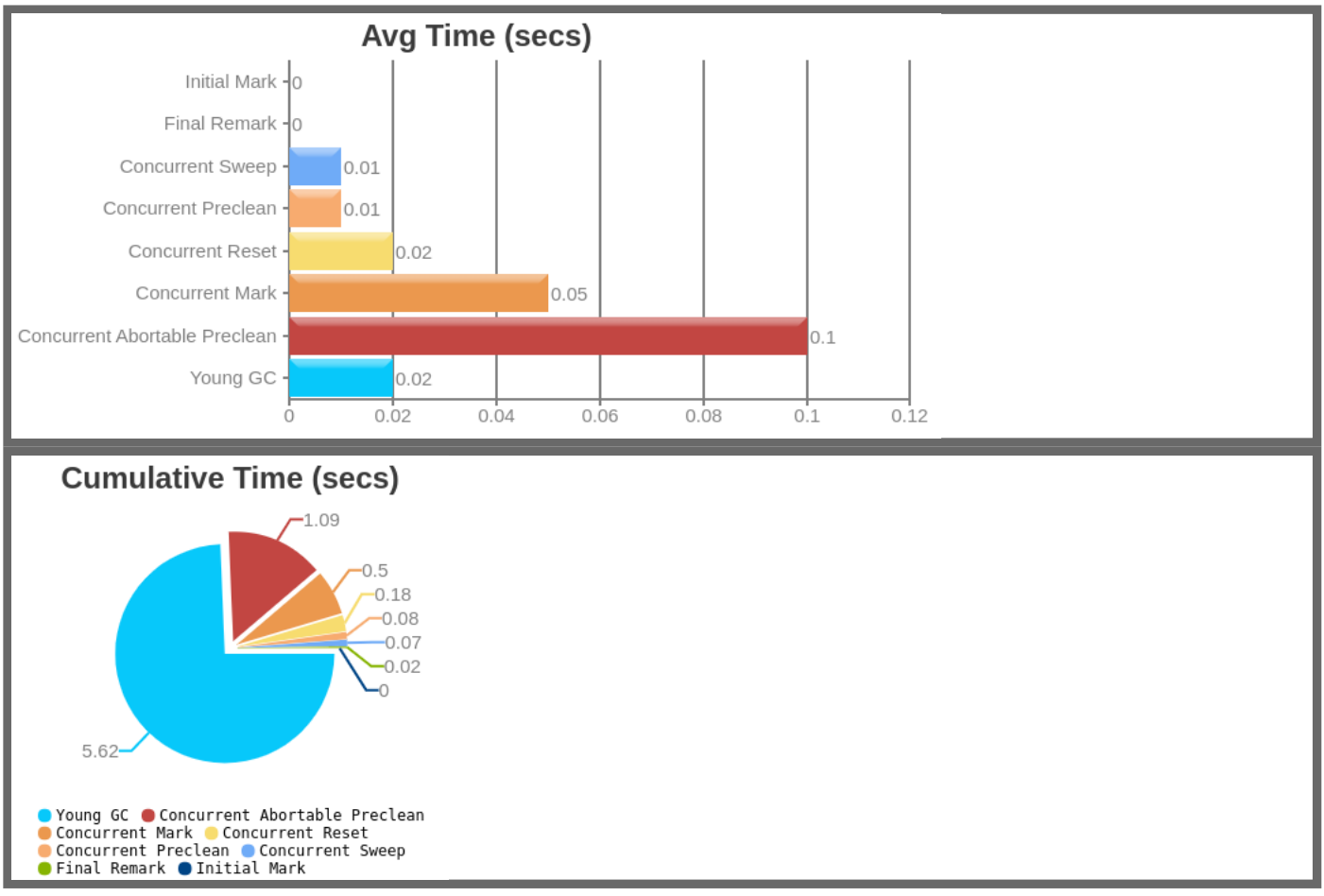

Reclaimed Bytes
(mb)
45
40
35
30
25
20
15
10
5
0
-5
06:56:06 pm
06:56:07 pm
06:56:08 pm
06:56:09 pm
06:56:10 pm
06:56:11 pm
06:56:12 pm
06:56:13 pm
06:56:14 pm
06:56:15 pm
06:56:16 pm
06:56:17 pm
06:56:18 pm
06:56:19 pm
Time UTC+0300
Young GC
Full GC


Young Gen
80
70
60
50
40
30
20
10
0
(mb)
06:56:06 pm
06:56:07 pm
06:56:08 pm
06:56:09 pm
06:56:10 pm
06:56:11 pm
06:56:12 pm
06:56:13 pm
06:56:14 pm
06:56:15 pm
06:56:16 pm
06:56:17 pm
06:56:18 pm
06:56:19 pm
Time UTC+0300
allocated space
before GC
after GC


Old Gen
3,500
3,000
2,500
2,000
1,500
1,000
500
0
(mb)
06:56:06 pm
06:56:07 pm
06:56:08 pm
06:56:09 pm
06:56:10 pm
06:56:11 pm
06:56:12 pm
06:56:13 pm
06:56:14 pm
06:56:15 pm
06:56:16 pm
06:56:17 pm
06:56:18 pm
06:56:19 pm
06:56:20 pm
Time UTC+0300
allocated space
before GC
after GC



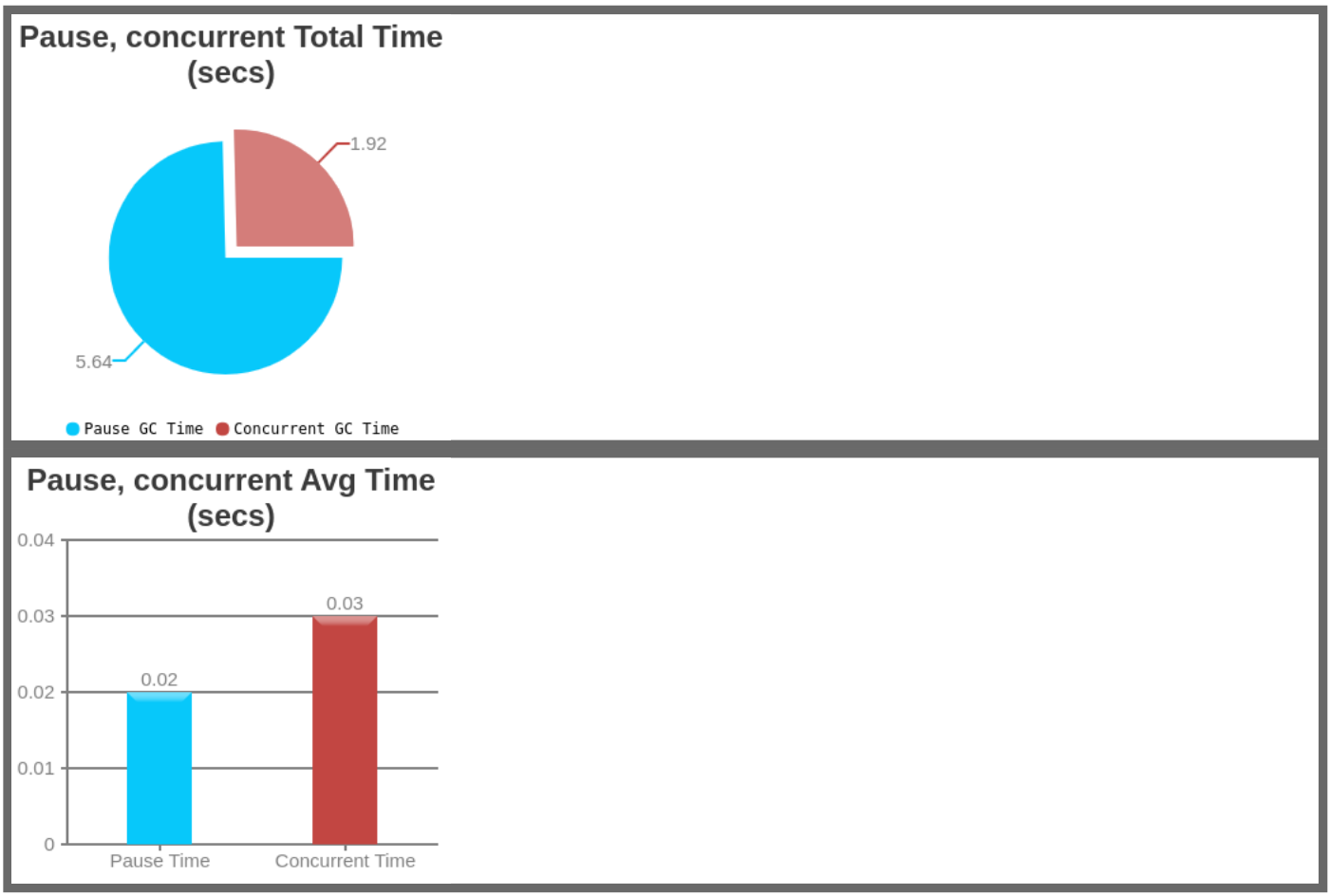
CMS Collection Phases Statistics



	Young GC	Concurrent Abortable Preclean	Concurrent Mark	Concurrent Reset	Concurrent Preclean	Concurrent Sweep	Final Remark	Initial Mark
Total Time	5 sec 620 ms	1 sec 90 ms	500 ms	180 ms	80 ms	70 ms	20 ms	0

Avg Time	18 ms	99 ms	45 ms	16 ms	7 ms	6 ms	2 ms	0
Std Dev Time	6 ms	180 ms	17 ms	10 ms	4 ms	5 ms	4 ms	0
Min Time	10 ms	20 ms	20 ms	10 ms	0	0	0	0
Max Time	40 ms	660 ms	80 ms	40 ms	10 ms	10 ms	10 ms	0
Count	317	11	11	11	11	11	11	11

CMS GC Time



Pause Time

Total Time	5 sec 640 ms
Avg Time	17 ms
Std Dev Time	7 ms
Min Time	0
Max Time	40 ms

Concurrent Time

Total Time	1 sec 920 ms

Avg Time	35 ms
Std Dev Time	88 ms
Min Time	0
Max Time	660 ms

Object Stats

(These are perfect micro-metrics (https://blog.gceasy.io/2017/05/30/improving-your-performance-reports/) to include in your performance reports)

Total created bytes	23.72 gb
Total promoted bytes	19.51 gb
Avg creation rate	1.73 gb/sec
Avg promotion rate	1.42 gb/sec

Memory Leak

No major memory leaks.

(**Note:** there are 8 flavours of OutOfMemoryErrors (https://tier1app.files.wordpress.com/2014/12/outofmemoryerror2.pdf). 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 (https://blog.gceasy.io/2016/12/22/total-time-for-which-application-threads-were-stopped/))

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 %
Allocation Failure	318	18 ms	110 ms	5 sec 730 ms	75.79%
Others	10	n/a	n/a	1 sec 830 ms	24.21%