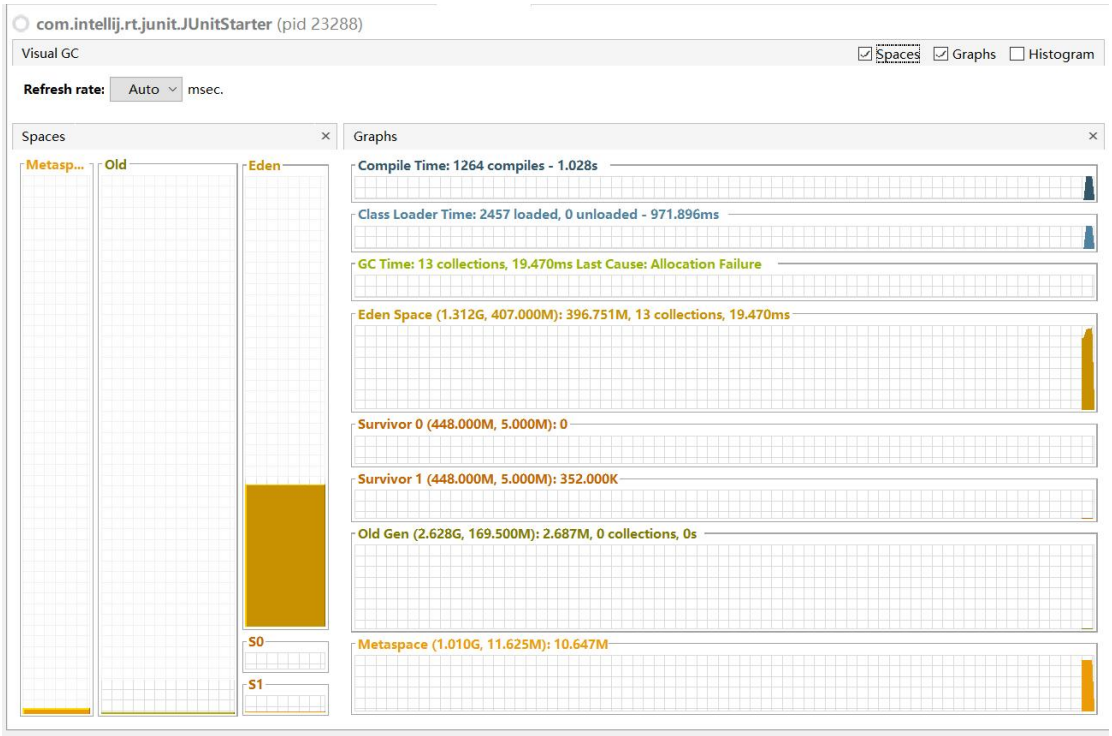
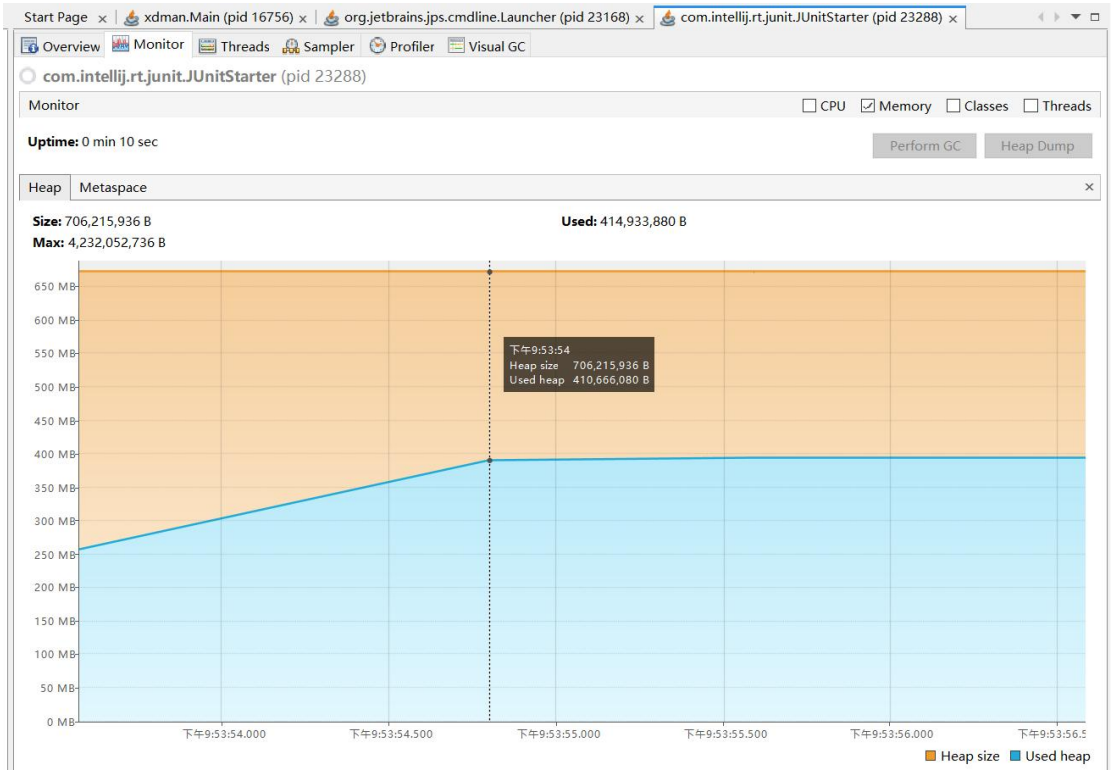
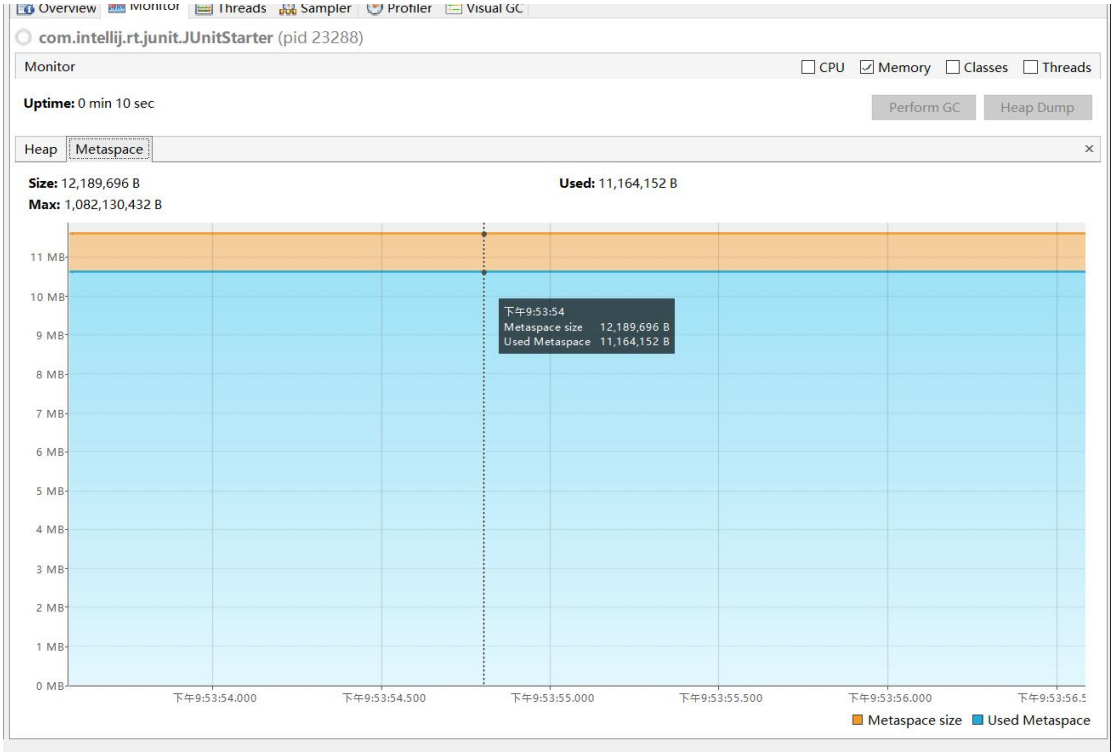
Compare the performance between MemAppender using a LinkedList

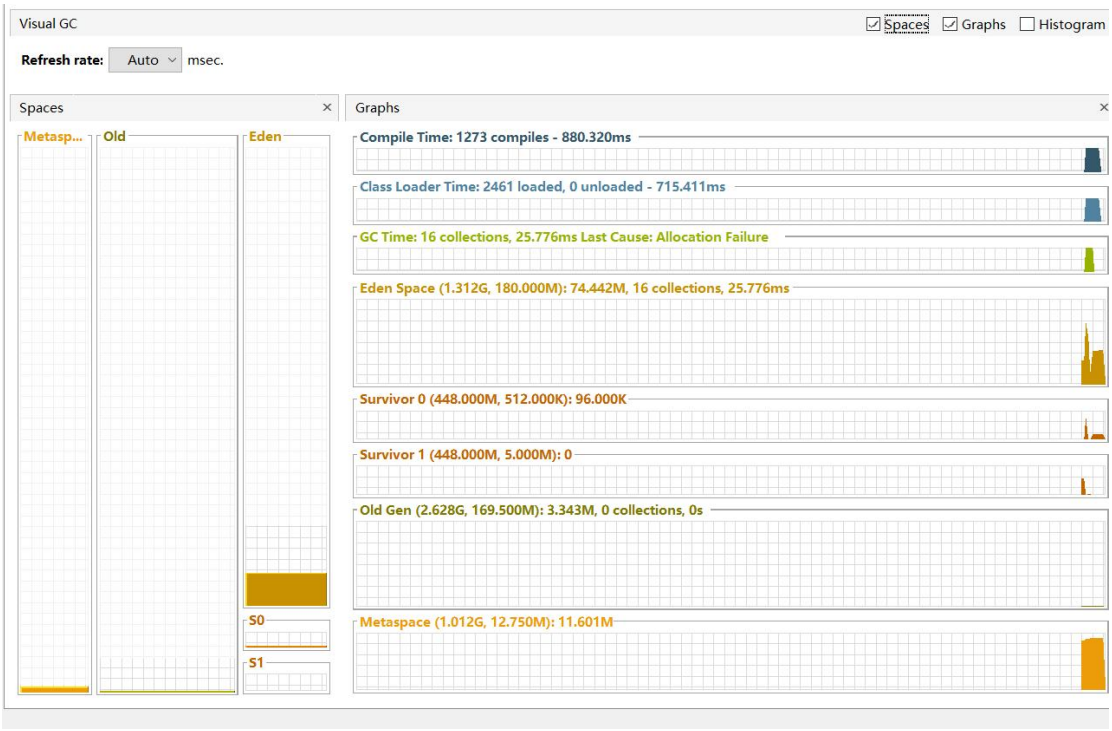
testLinkedlist()

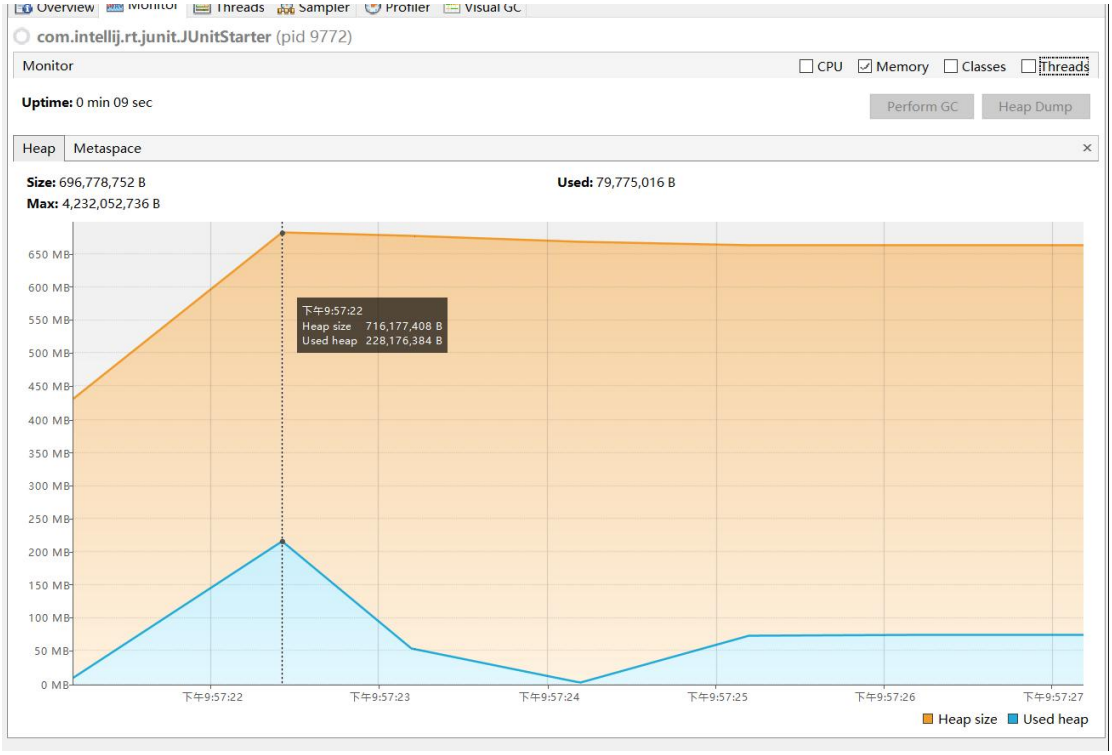


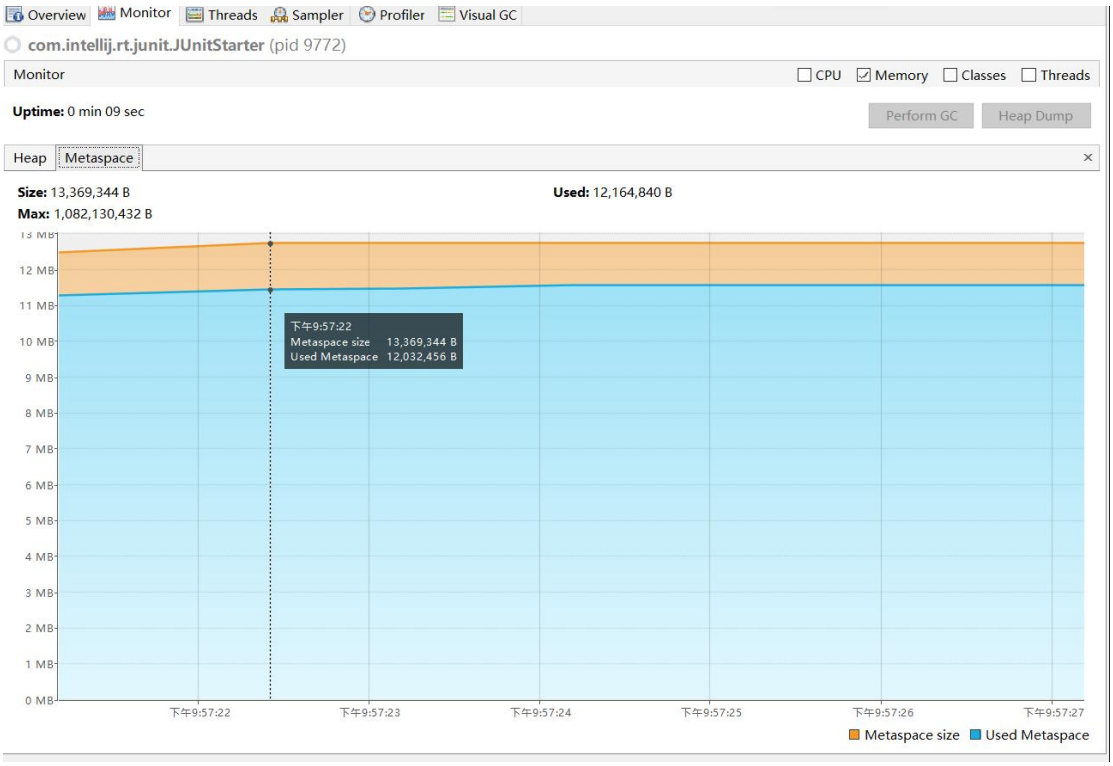




testArraylist()







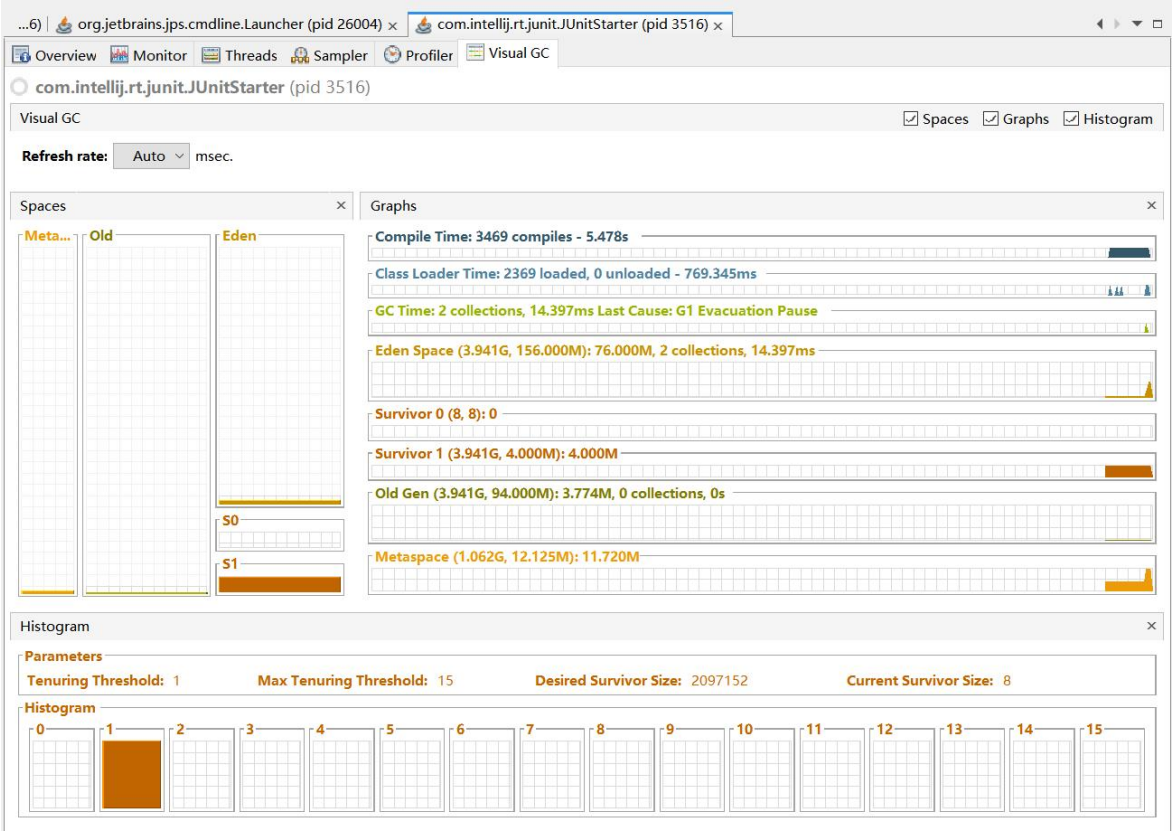
Report：

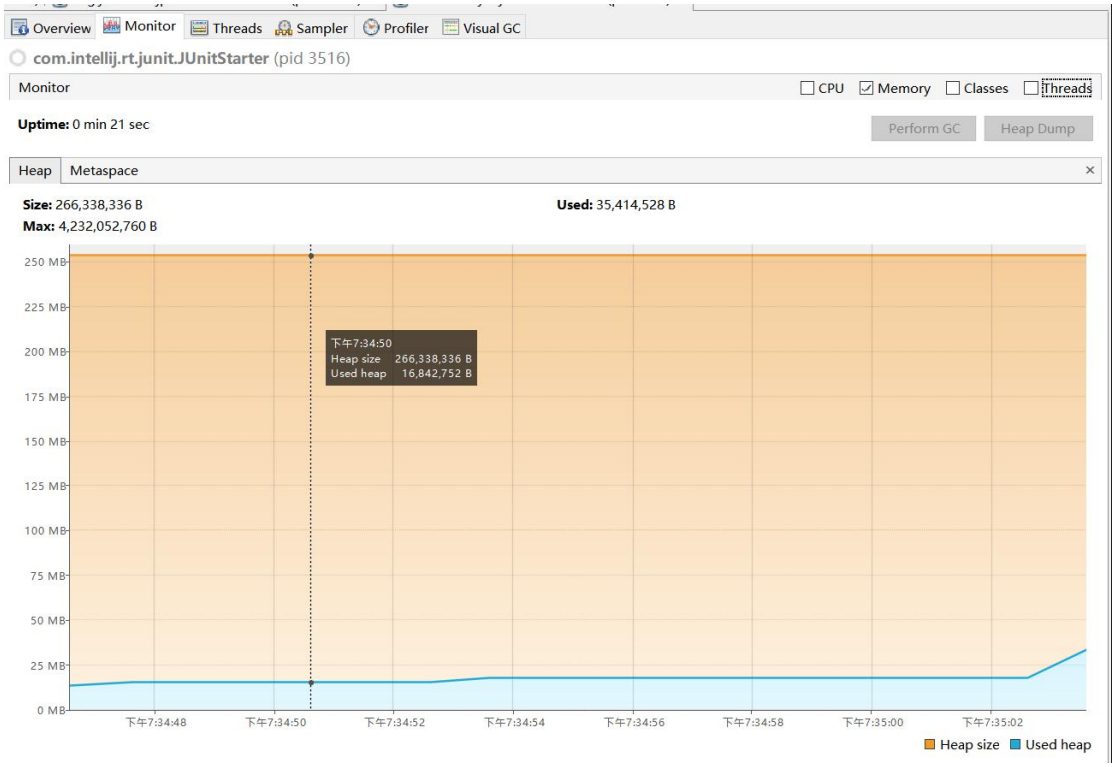
1.If MemAppender stores LoggingEvents in a Arraylist, it will use less time than storing them in a Linkedlist when the Logger records the same number of logs（When the loggingEvents' number is up to a point, this is going to happen to above situation.However,if the loggingEvents' number isn't up to a point, the Linkedlist is more faster than the ArrayList).

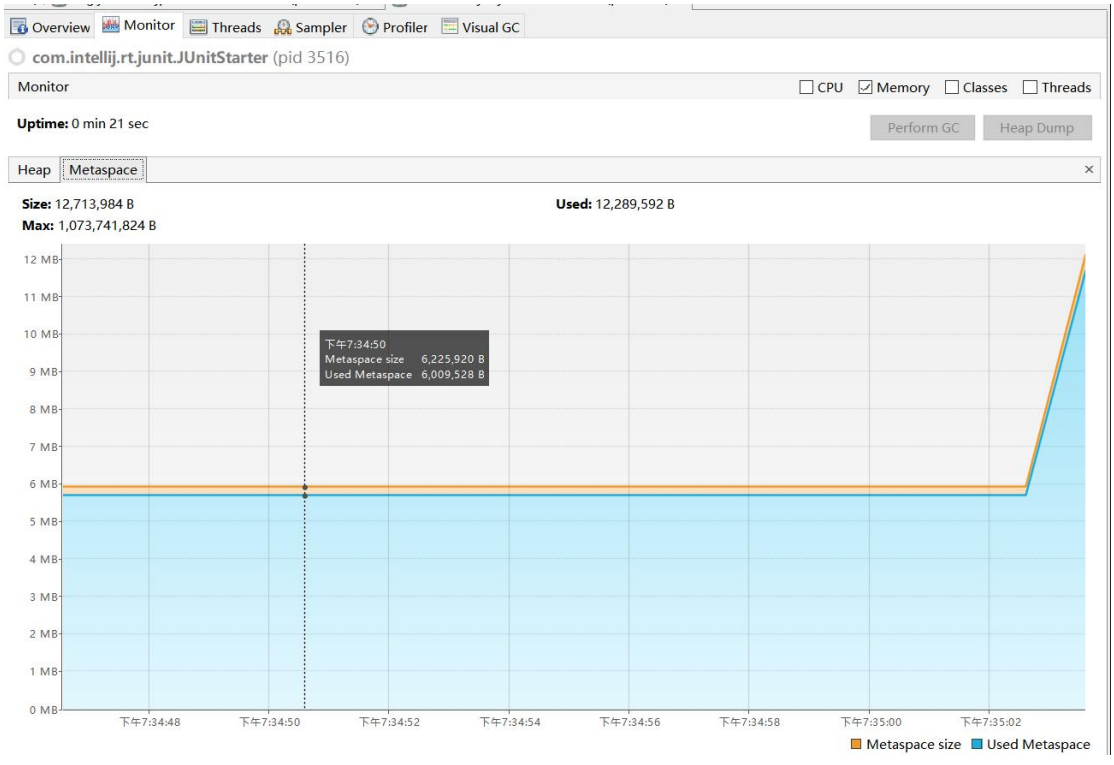
2.The LinkedList consume more memory than the Arraylist when they contain the same number of data(the same type).

Compare the performance between ConsoleAppender and FileAppender

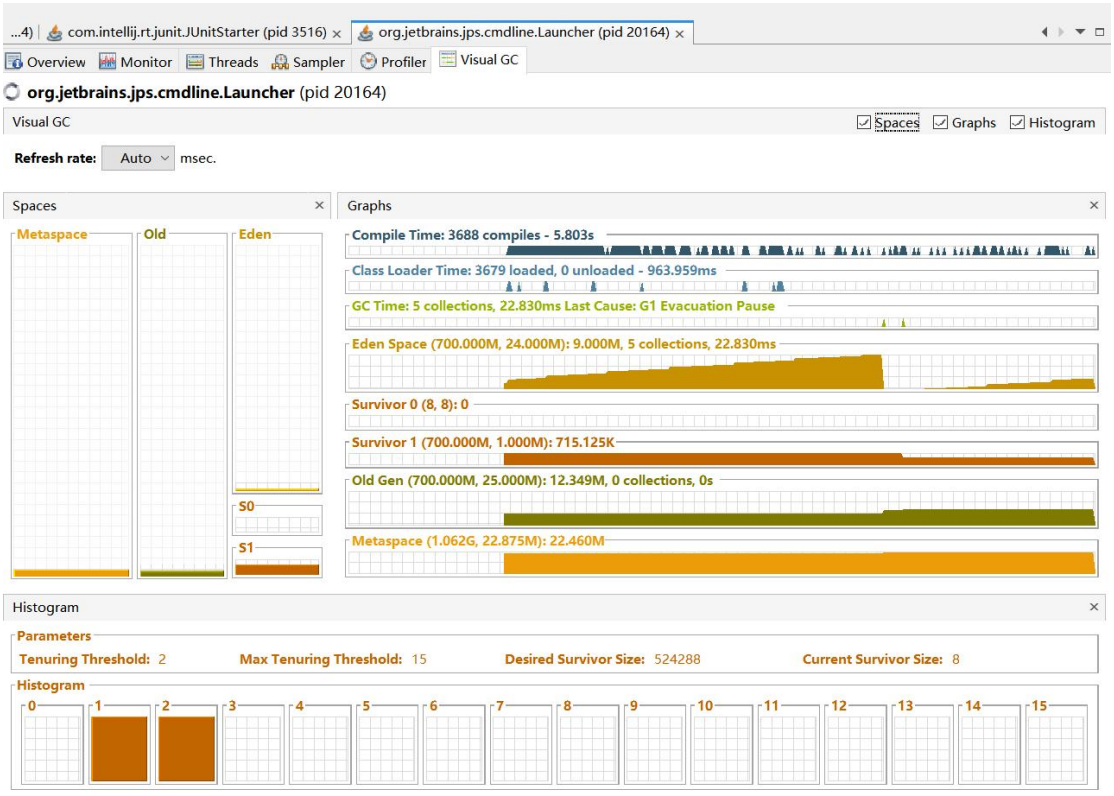
ConsoleAppender

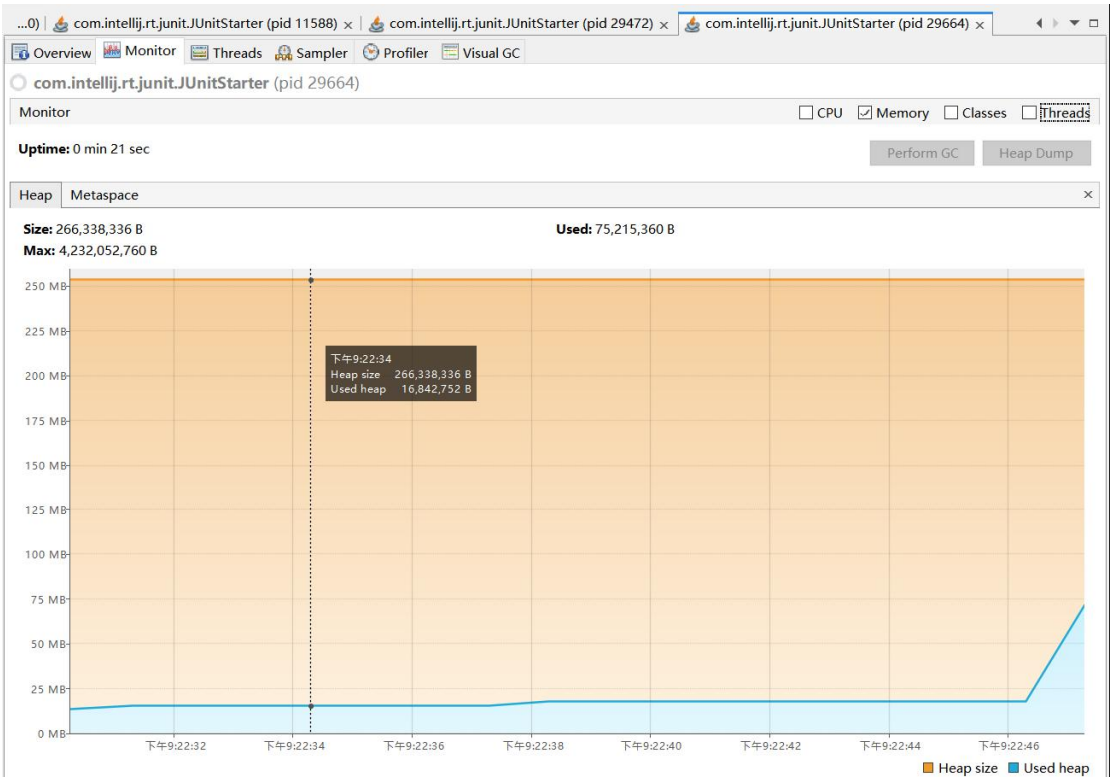


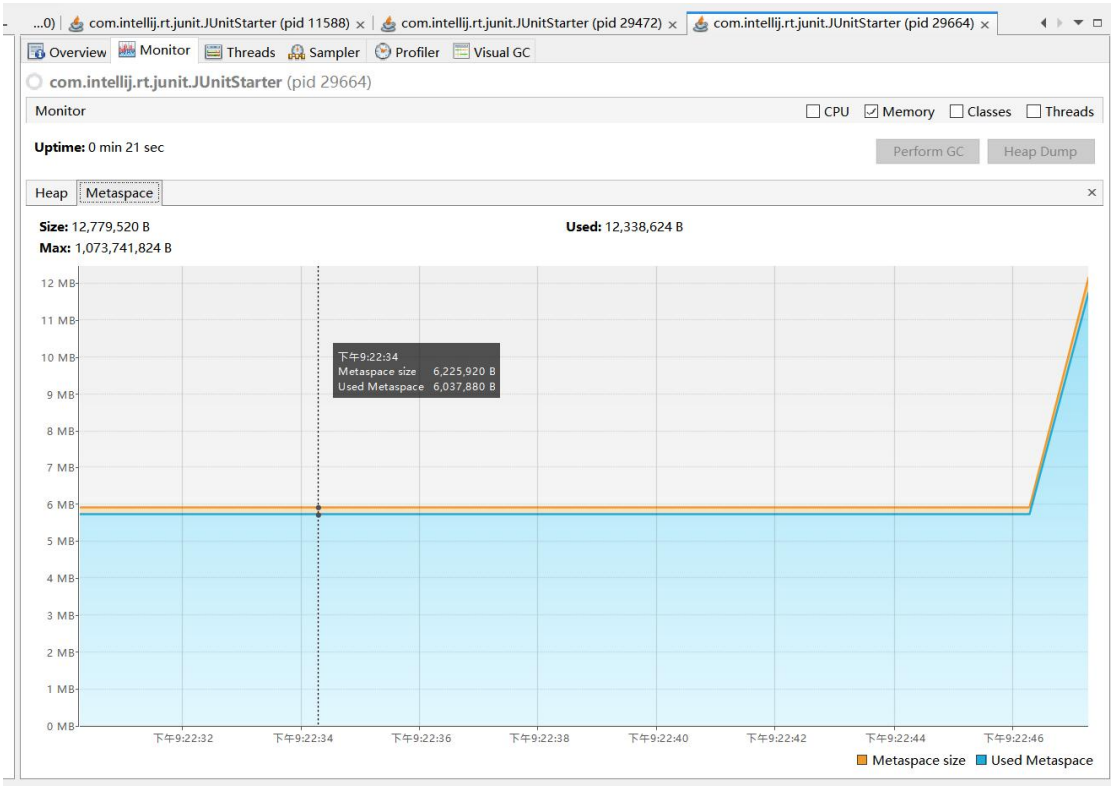




FileAppender


...4) | com.intellij.rt.junit.JUnitStarter (pid 3516) × org.jetbrains.jps.cmdline.Launcher (pid 20164) ×
Overview Monitor Threads Sampler Profiler Visual GC
org.jetbrains.jps.cmdline.Launcher (pid 20164)
Visual GC
Spaces Graphs Histogram
Refresh rate: Auto msec.
Spaces
Metaspace
Old
Eden
S0
S1
Graphs
Compile Time: 3688 compiles - 5.803s
Class Loader Time: 3679 loaded, 0 unloaded - 963.959ms
GC Time: 5 collections, 22.830ms Last Cause: G1 Evacuation Pause
Eden Space (700.000M, 24.000M): 9.000M, 5 collections, 22.830ms
Survivor 0 (8, 8): 0
Survivor 1 (700.000M, 1.000M): 715.125K
Old Gen (700.000M, 25.000M): 12.349M, 0 collections, 0s
Metaspace (1.062G, 22.875M): 22.460M
Histogram
Parameters
Tenuring Threshold: 2
Max Tenuring Threshold: 15
Desired Survivor Size: 524288
Current Survivor Size: 8
Histogram
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


...0) | com.intellij.rt.junit.JUnitStarter (pid 11588) × com.intellij.rt.junit.JUnitStarter (pid 29472) × com.intellij.rt.junit.JUnitStarter (pid 29664) ×
Overview Monitor Threads Sampler Profiler Visual GC
com.intellij.rt.junit.JUnitStarter (pid 29664)
Monitor
CPU Memory Classes Threads
Uptime: 0 min 21 sec
Perform GC Heap Dump
Heap Metaspace
Size: 266,338,336 B
Max: 4,232,052,760 B
Used: 75,215,360 B
250 MB
225 MB
200 MB
175 MB
150 MB
125 MB
100 MB
75 MB
50 MB
25 MB
0 MB
下午9:22:34
Heap size 266,338,336 B
Used heap 16,842,752 B
下午9:22:32 下午9:22:34 下午9:22:36 下午9:22:38 下午9:22:40 下午9:22:42 下午9:22:44 下午9:22:46
Heap size Used heap



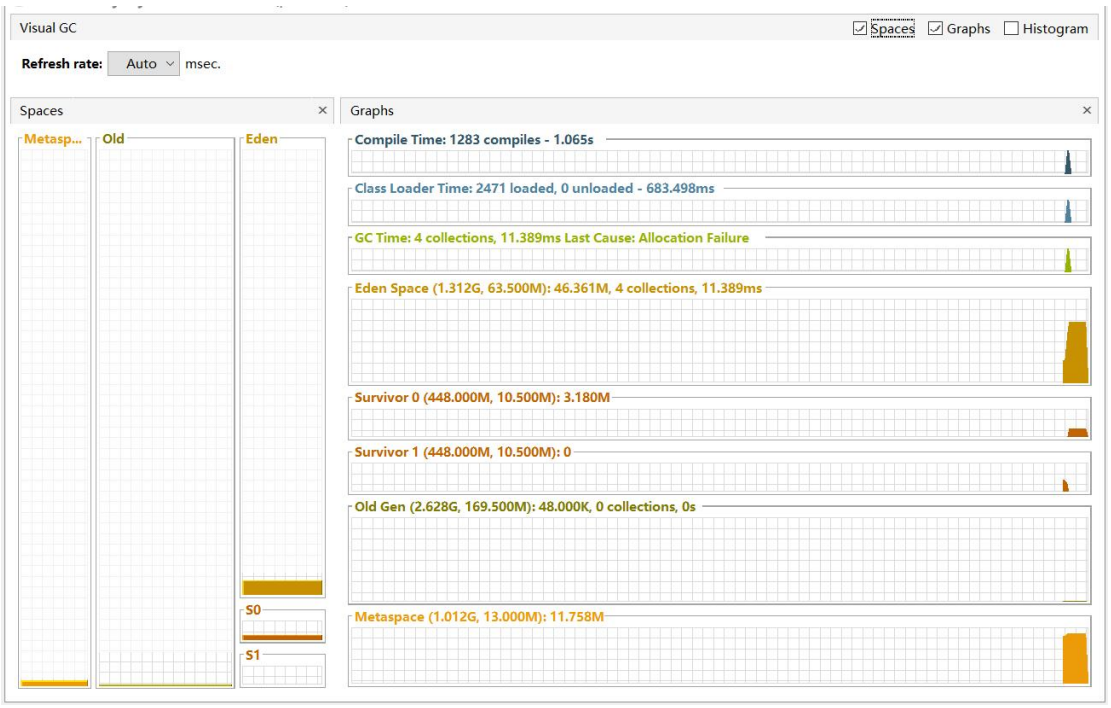
Report：

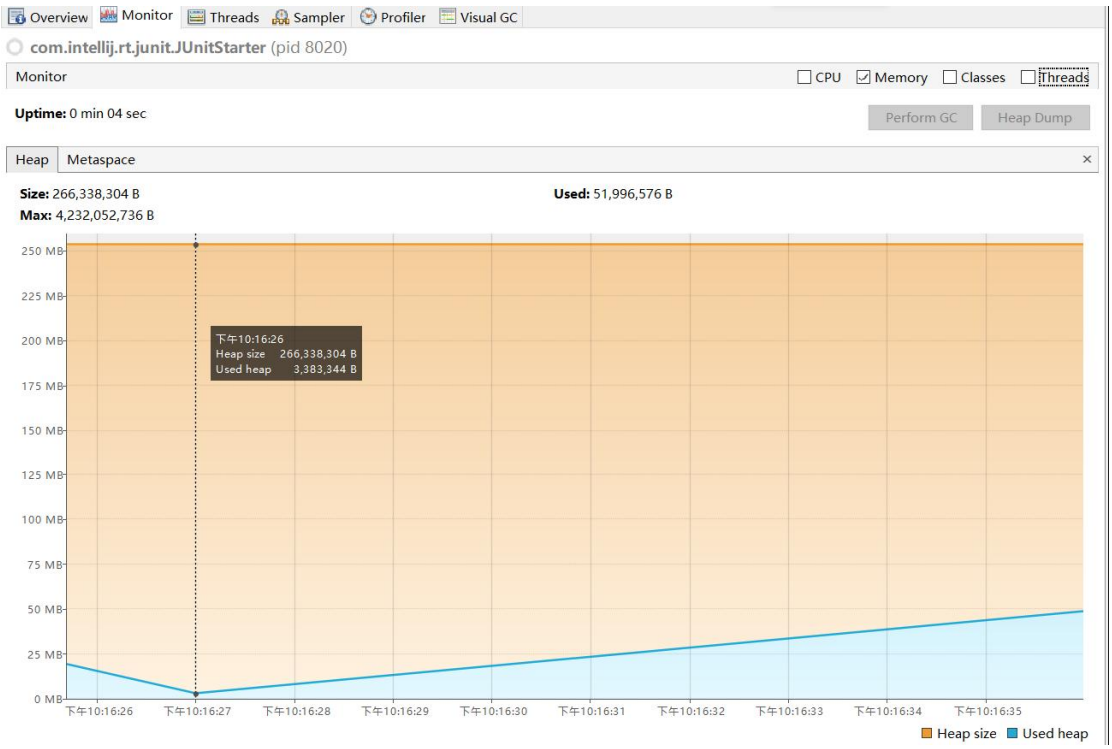
1.ConsoleAppender consumes less time than FileAppender when they are used to output the same content(logs) with a Logger.

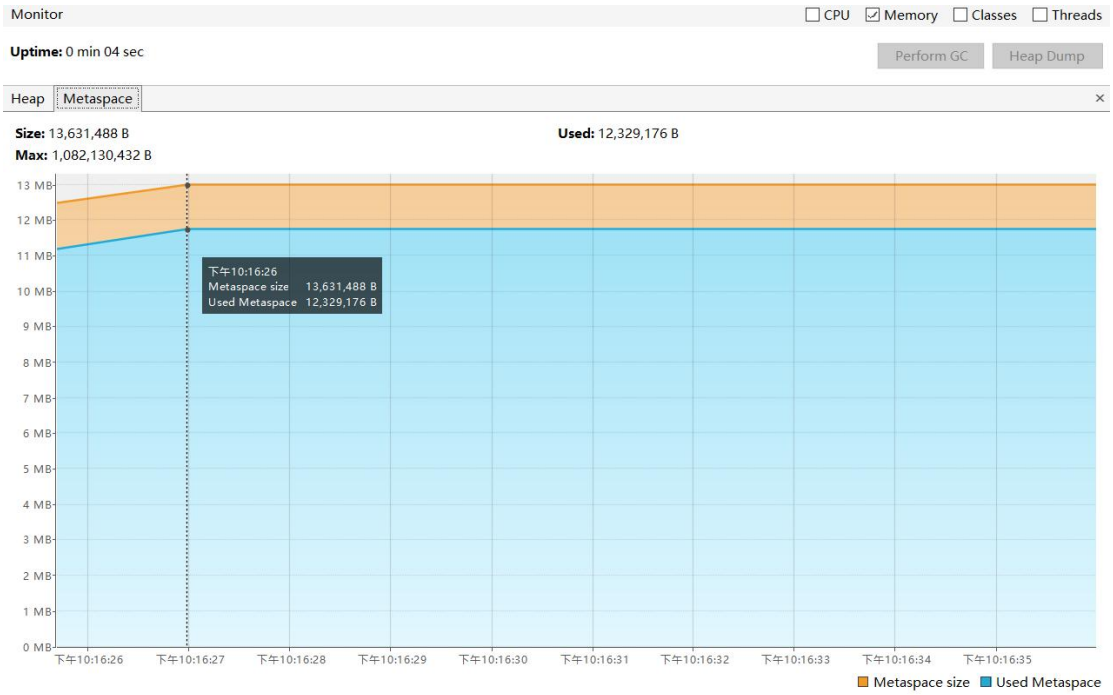
2.FileAppender consumes more memory than ConsoleAppender when they are used to output the same content(logs) with a Logger.

Compare the performance between PatternLayout and VelocityLayout

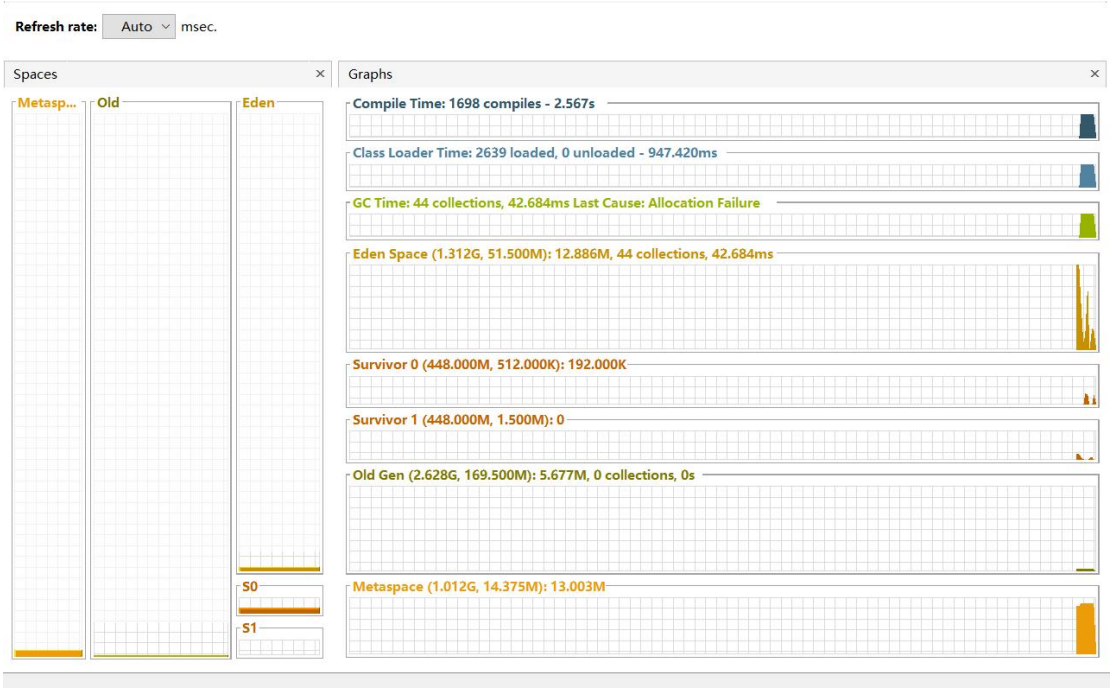
PatternLayout


Visual GC
Spaces
Graphs
Histogram
Refresh rate:
Auto
msec.
Spaces
Graphs
Metasp...
Old
Eden
S0
S1
Compile Time: 1283 compiles - 1.065s
Class Loader Time: 2471 loaded, 0 unloaded - 683.498ms
GC Time: 4 collections, 11.389ms Last Cause: Allocation Failure
Eden Space (1.312G, 63.500M): 46.361M, 4 collections, 11.389ms
Survivor 0 (448.000M, 10.500M): 3.180M
Survivor 1 (448.000M, 10.500M): 0
Old Gen (2.628G, 169.500M): 48.000K, 0 collections, 0s
Metaspace (1.012G, 13.000M): 11.758M


Overview
Monitor
Threads
Sampler
Profiler
Visual GC
com.intellij.rt.junit.JUnitStarter (pid 8020)
Monitor
CPU
Memory
Classes
Threads
Uptime: 0 min 04 sec
Perform GC
Heap Dump
Heap
Metaspace
Size: 266,338,304 B
Max: 4,232,052,736 B
Used: 51,996,576 B
250 MB
225 MB
200 MB
175 MB
150 MB
125 MB
100 MB
75 MB
50 MB
25 MB
0 MB
下午10:16:26
Heap size 266,338,304 B
Used heap 3,383,344 B
下午10:16:26
下午10:16:27
下午10:16:28
下午10:16:29
下午10:16:30
下午10:16:31
下午10:16:32
下午10:16:33
下午10:16:34
下午10:16:35
Heap size
Used heap



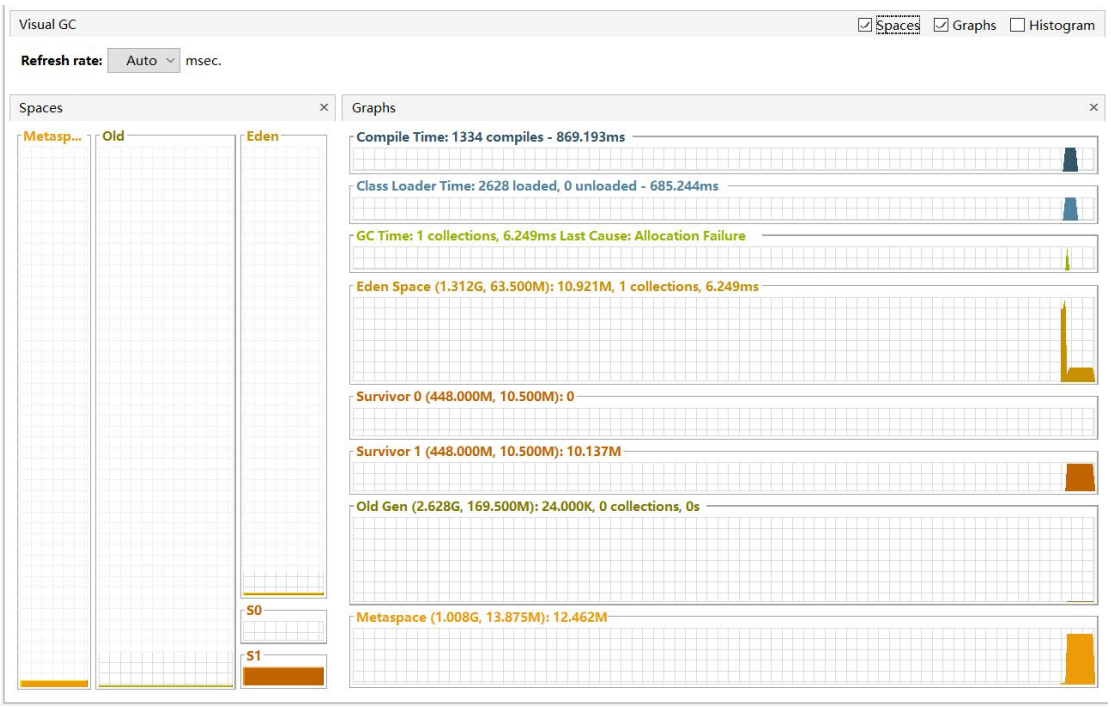
VelocityLayout

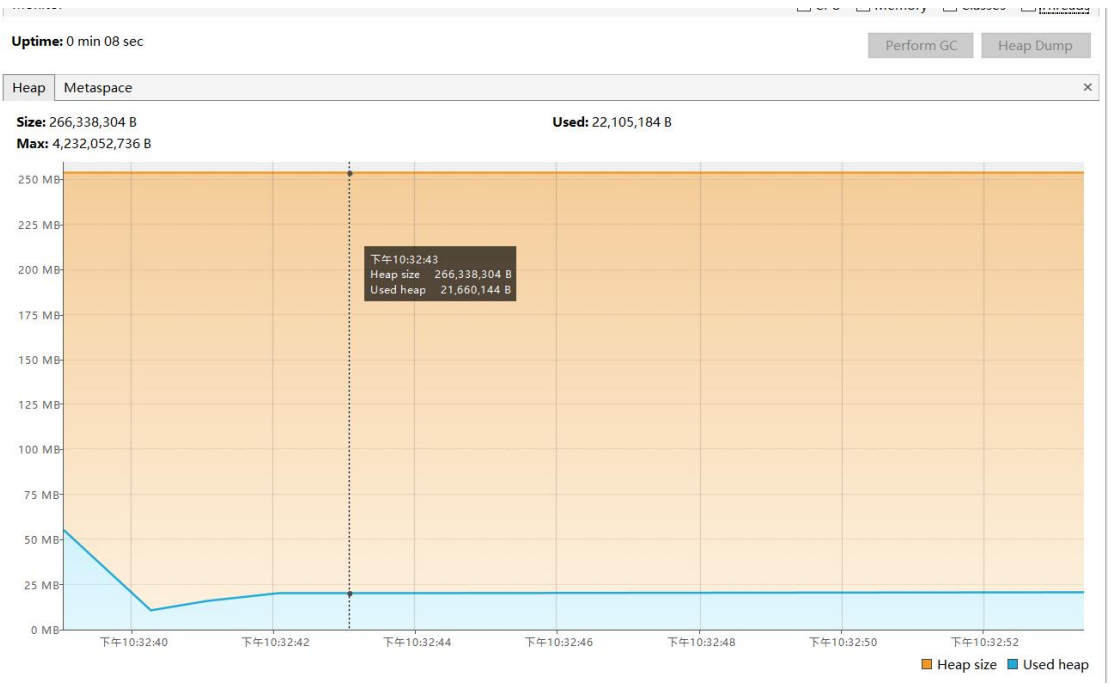


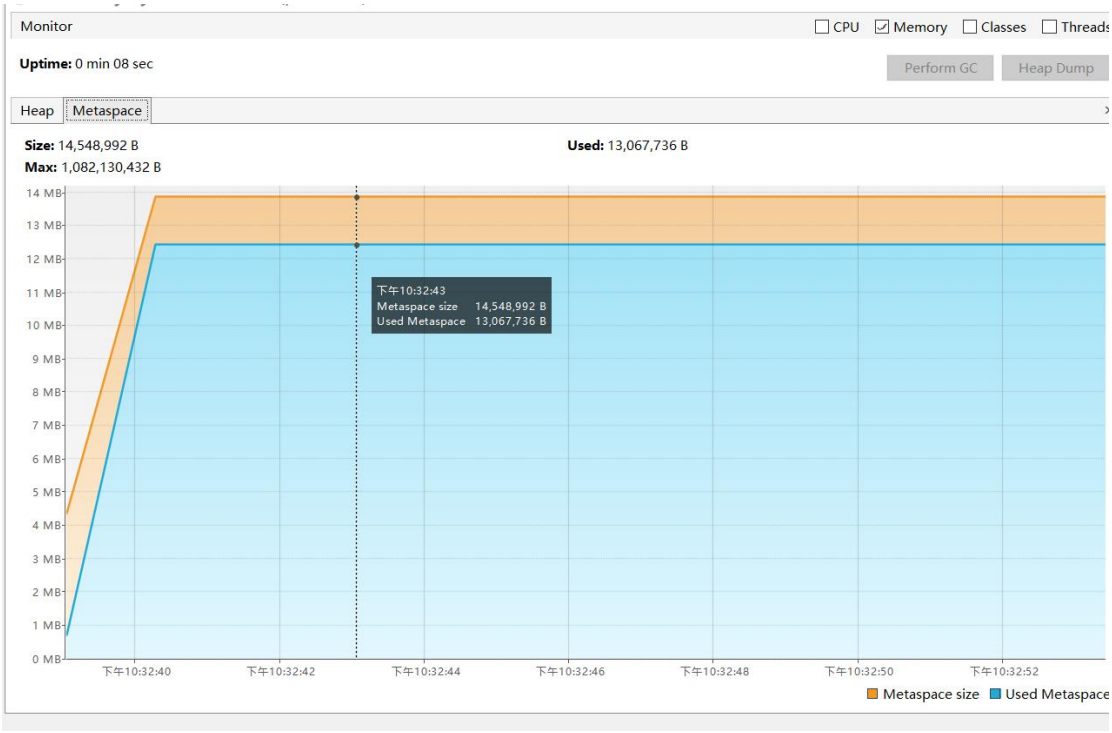
Report：

PatternLayout consumes less time and memory than VelocityLayout when they are used to output the same content(logs) with a logger.

Test performance before and after **maxSize** has been reached, and with different maxSize values.


Visual GC
Spaces
Graphs
Histogram
Refresh rate:
Auto
msec.
Spaces
Graphs
Metasp...
Old
Eden
S0
S1
Compile Time: 1334 compiles - 869.193ms
Class Loader Time: 2628 loaded, 0 unloaded - 685.244ms
GC Time: 1 collections, 6.249ms Last Cause: Allocation Failure
Eden Space (1.312G, 63.500M): 10.921M, 1 collections, 6.249ms
Survivor 0 (448.000M, 10.500M): 0
Survivor 1 (448.000M, 10.500M): 10.137M
Old Gen (2.628G, 169.500M): 24.000K, 0 collections, 0s
Metaspace (1.008G, 13.875M): 12.462M


Uptime: 0 min 08 sec
Perform GC
Heap Dump
Heap
Metaspace
Size: 266,338,304 B
Max: 4,232,052,736 B
Used: 22,105,184 B
250 MB
225 MB
200 MB
175 MB
150 MB
125 MB
100 MB
75 MB
50 MB
25 MB
0 MB
下午10:32:43
Heap size 266,338,304 B
Used heap 21,660,144 B
下午10:32:40
下午10:32:42
下午10:32:44
下午10:32:46
下午10:32:48
下午10:32:50
下午10:32:52
Heap size
Used heap



Report：

If we set a maxSize and a logs' number that have been over the maxSize , it will consume more time and memory than it don't reach the maxSize.