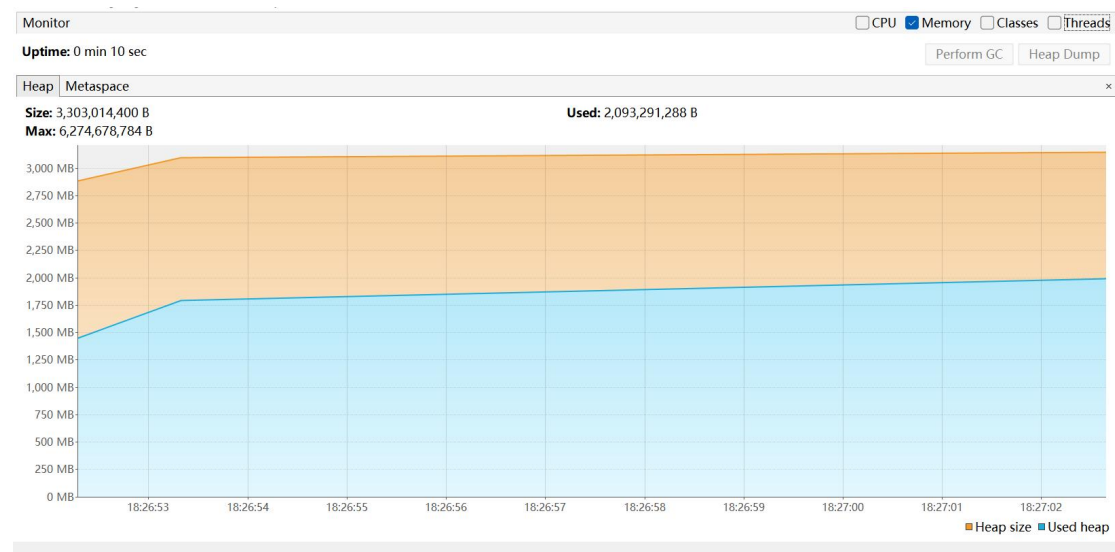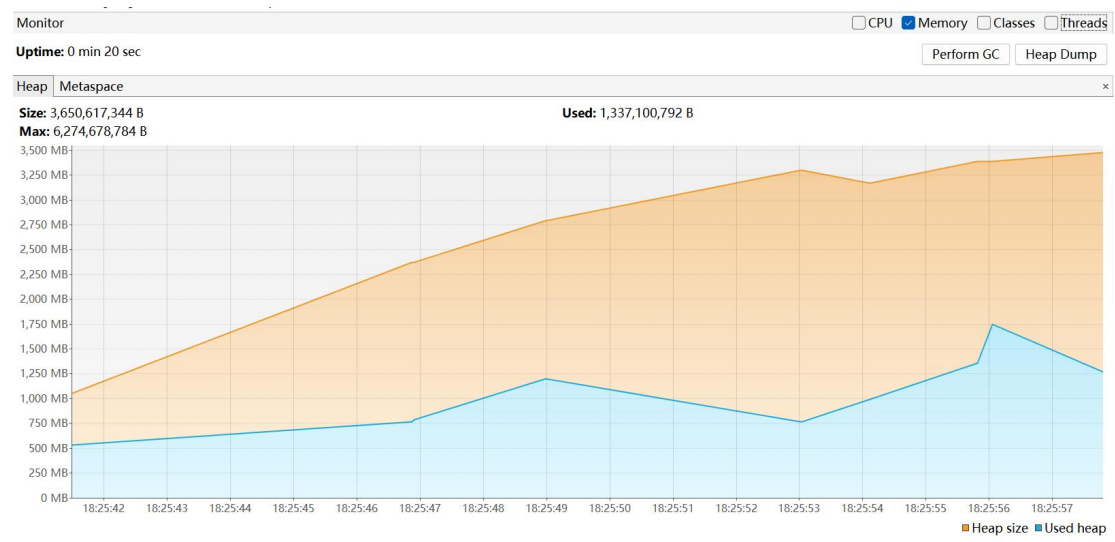Compare LinkList with ArrayList:

ArrayList:

LinkList:

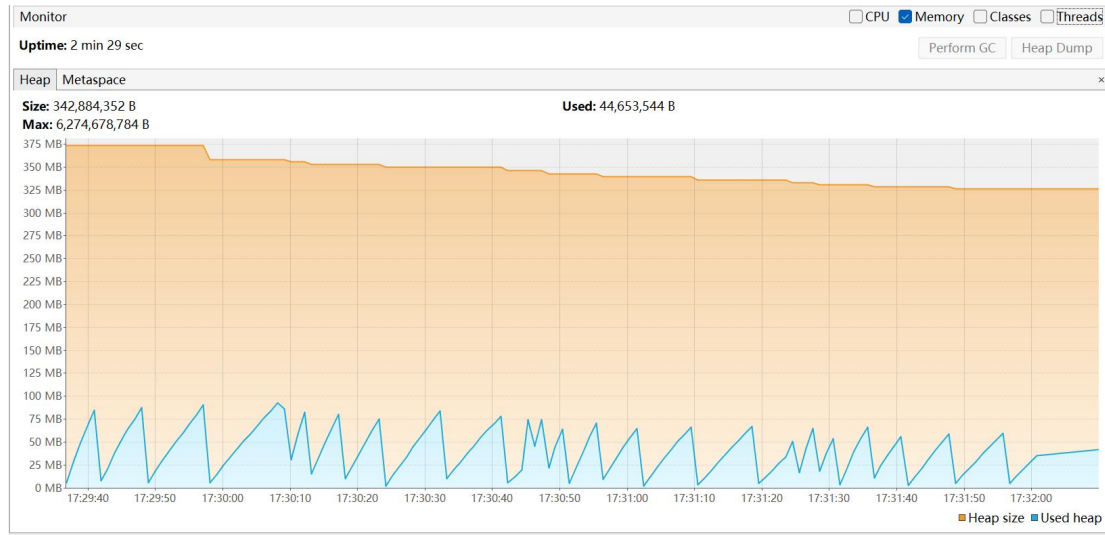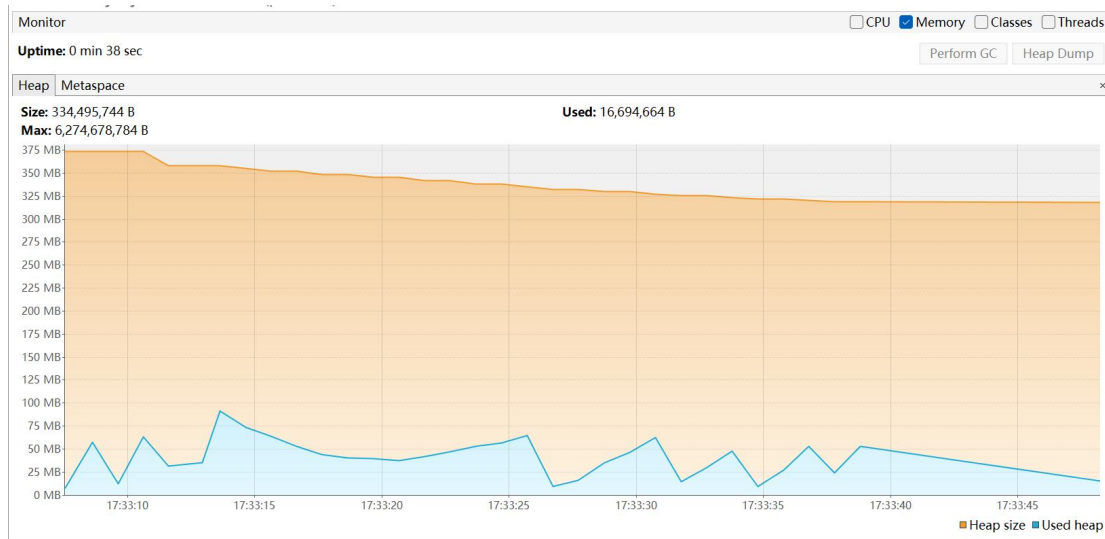Analysis result:

The ArrayList takes less time than LinkList, but it occupied more memory.

Compare ConsoleAppender with FileAppender:

ConsoleAppender:

FileAppender:

Analysis result:

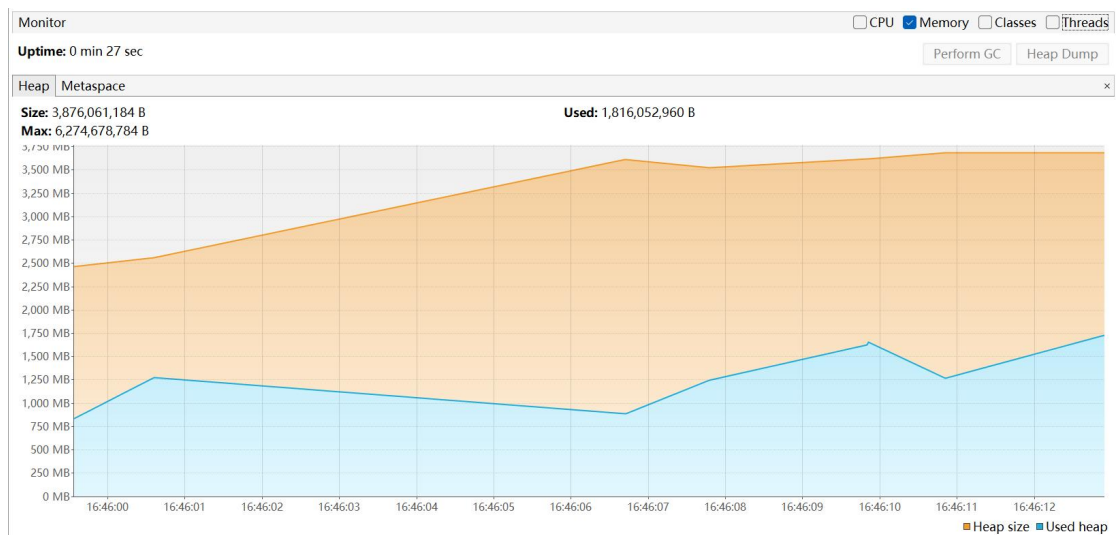ConsoleAppender takes more time than FileAppender and occupied more memory.

ConsoleAppender: The log content can be output in the console.
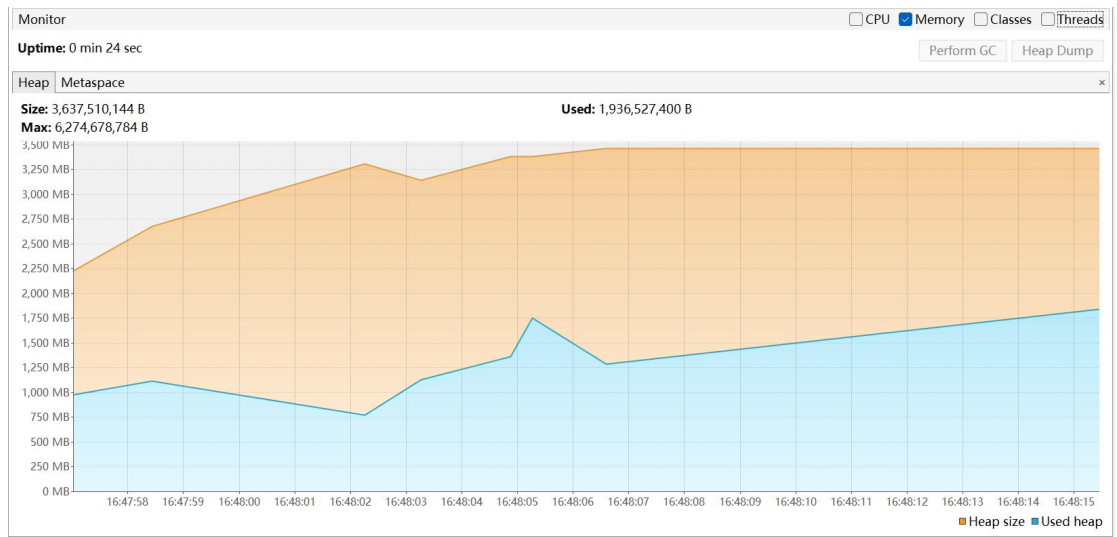
FileAppender: The log content can be output in the files.

Compare PatternLayout with VelocityLayout:
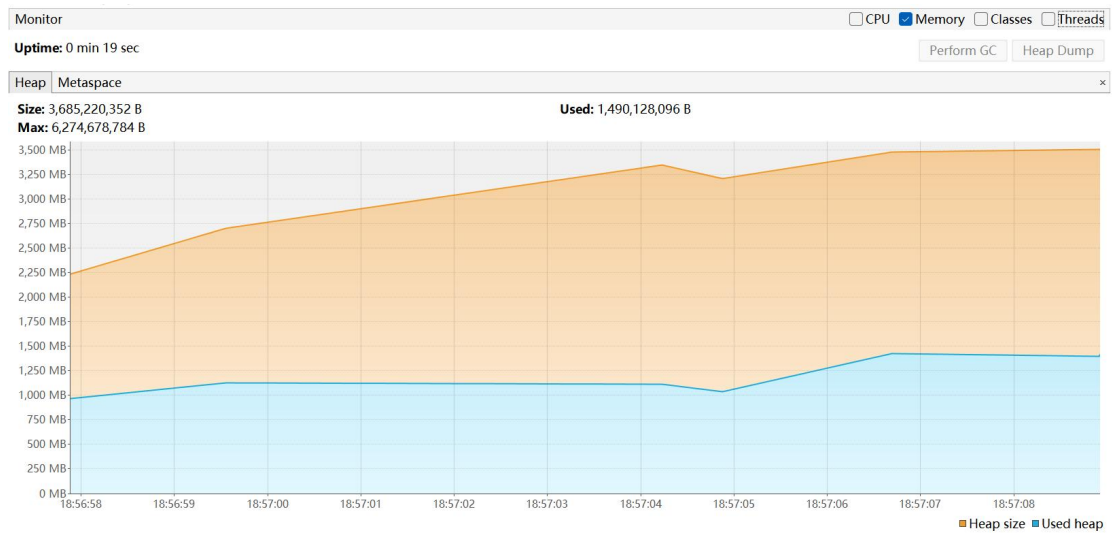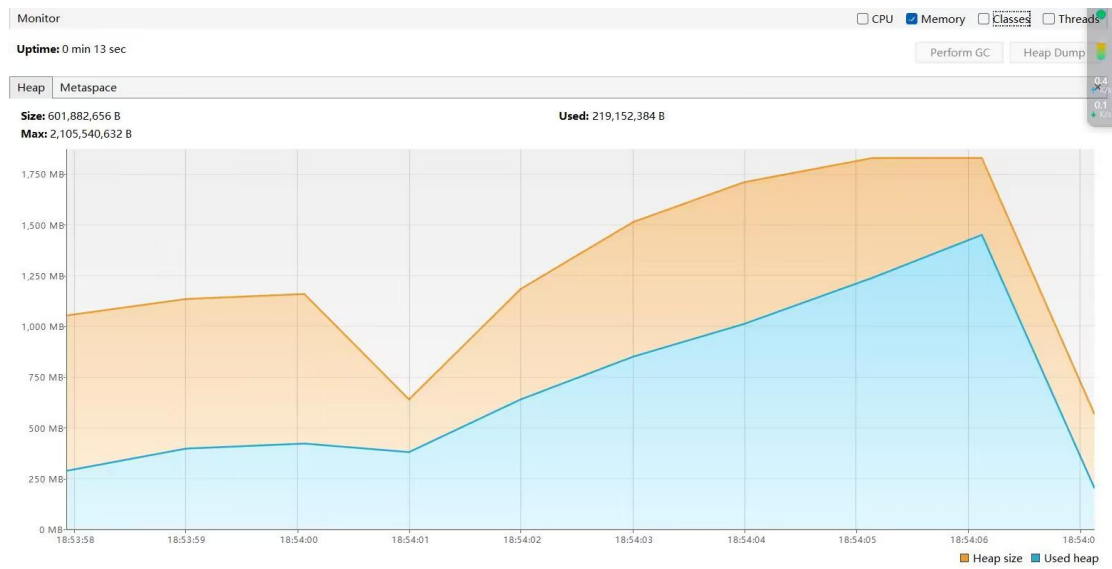
PatternLayout:

VelocityLayout:

Analysis result:

I write their formats in the same form, PatternLayout takes more time than VelocityLayout.

Analysis Max Size:

Before max size:

Monitor ☐CPU ☑Memory ☐Classes ☐Threads

Uptime: 0 min 19 sec Perform GC Heap Dump

Heap | Metaspace ×

Size: 3,685,220,352 B **Used:** 1,490,128,096 B
Max: 6,274,678,784 B

3,500 MB
3,250 MB
3,000 MB
2,750 MB
2,500 MB
2,250 MB
2,000 MB
1,750 MB
1,500 MB
1,250 MB
1,000 MB
750 MB
500 MB
250 MB
0 MB
18:56:58 18:56:59 18:57:00 18:57:01 18:57:02 18:57:03 18:57:04 18:57:05 18:57:06 18:57:07 18:57:08

■ Heap size ■ Used heap

After max size：

Monitor ☐CPU ☑Memory ☐Classes ☐Threads

Uptime: 0 min 13 sec Perform GC Heap Dump

Heap | Metaspace

Size: 601,882,656 B **Used:** 219,152,384 B
Max: 2,105,540,632 B

1,750 MB
1,500 MB
1,250 MB
1,000 MB
750 MB
500 MB
250 MB
0 MB
18:53:58 18:53:59 18:54:00 18:54:01 18:54:02 18:54:03 18:54:04 18:54:05 18:54:06 18:54:0

■ Heap size ■ Used heap

Analysis result：

Due to the long running time and useless data wasting too much memory, the virtual machine
starts to collect garbage, so it will drop. When it does not reach the MaxSize, it will keep
increasing. If it exceeds the set maximum limit, it will crash. After reaching MaxSize, he will clean
up the unnecessary data and only keep the needed data.