Performance Analysis Report

Overview

This report aims to analyze and summarize the results of stress tests conducted on MemAppender, ConsoleAppender, and FileAppender. The purpose of the tests is to evaluate the performance of different log appenders when handling a large volume of log events, especially under various maxSize configurations. The test results will help us understand performance differences under different configurations and provide a basis for future optimizations.

Testing Environment

IDE: IntelliJ IDEA
Java Version: Java 21
Operating System: Windows 11
Performance Analysis Tool: IntelliJ Profiler
Test Tool: JUnit 5
Log Appenders: MemAppender, ConsoleAppender, FileAppender
Test Data Volume: 100, 1000, 10000 log messages
Test Type: Performance test, including time consumption and memory usage

Test Results

1. MemAppender Performance Test

Performance Comparison Using ArrayList and LinkedList

ArrayList MemAppender:
Maximum Log Count: 100
Discarded Log Count: 40800
Time Consumption: 559399 ns

LinkedList MemAppender:
Maximum Log Count: 100
Discarded Log Count: 41800
Time Consumption: 894400 ns

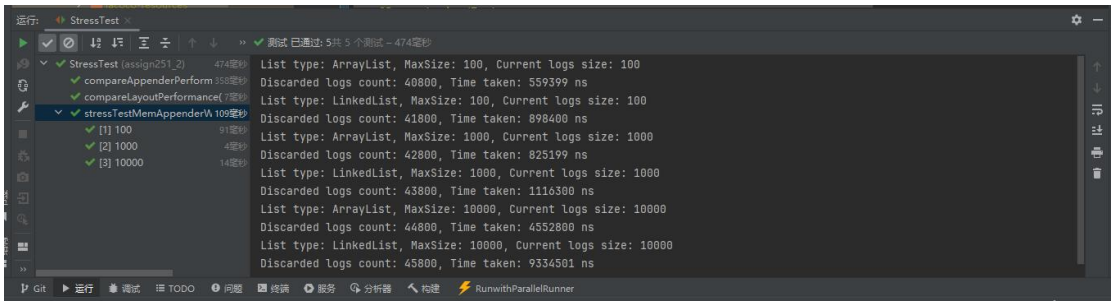
Comparison with Different maxSize Values

ArrayList MemAppender (maxSize=1000):
Discarded Log Count: 42800
Time Consumption: 825199 ns

LinkedList MemAppender (maxSize=1000):
Discarded Log Count: 43800
Time Consumption: 1116300 ns

ArrayList MemAppender (maxSize=10000):
Discarded Log Count: 44800
Time Consumption: 4552800 ns

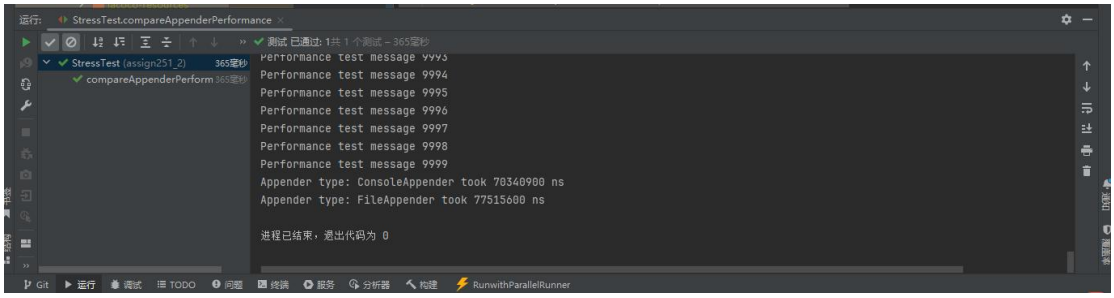
LinkedList MemAppender (maxSize=10000):
Discarded Log Count: 45800
Time Consumption: 9334501 ns



2. ConsoleAppender and FileAppender Performance Test

ConsoleAppender:
Time Consumption: 70340900 ns

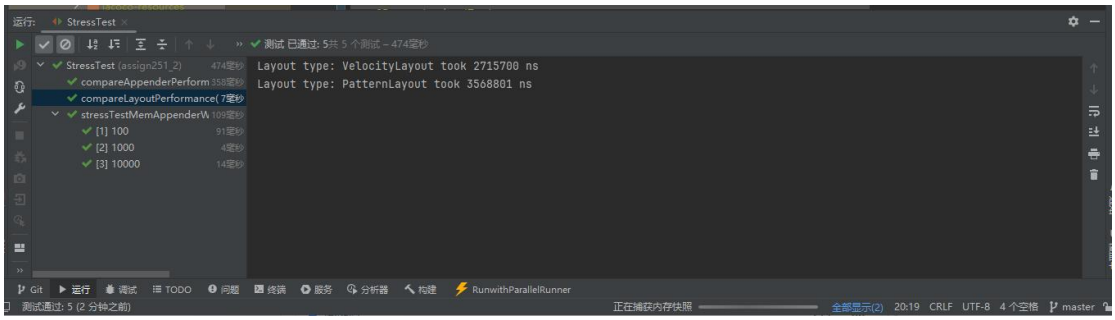
FileAppender:
Time Consumption: 77515600 ns



3. Performance Comparison of Different Layouts

VelocityLayout:
Time Consumption: 2715700 ns

PatternLayout:
Time Consumption: 3568801 ns



Conclusion

ArrayList provides better performance than LinkedList when used as the underlying data structure for MemAppender, especially when the log count reaches the maximum limit.
As maxSize increases, the time consumption of MemAppender also increases, but ArrayList still outperforms LinkedList.

ConsoleAppender and FileAppender show significantly lower performance when handling a large volume of logs compared to MemAppender, with FileAppender being the slowest, likely due to disk I/O overhead.

VelocityLayout is more efficient in formatting log messages than PatternLayout.

Recommendations

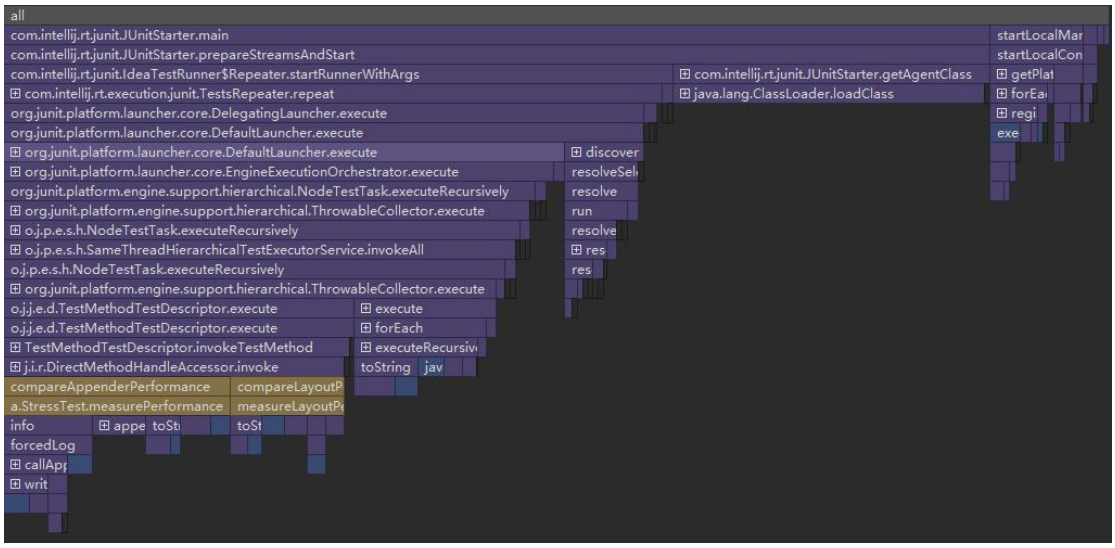
For scenarios requiring high-performance logging, it is recommended to use MemAppender with ArrayList.

For applications that need to persist logs to disk, the performance impact of FileAppender should be considered, and optimizations should be made when necessary.

In practical applications, the choice of log appender and layout should be based on specific needs, taking into account the impact of maxSize configuration on performance.

Appendix

Memory Allocation Chart: Shows memory usage at different test stages.



Time Allocation Chart: Shows time consumption at different test stages