

# **Отчёт по нагрузочному тестированию сервиса “book”**

# Оглавление

Цели и результаты тестирования.....	3
Цели тестирования.....	3
Результаты тестирования .....	3
Проведенные тесты.....	3
Максимальная производительность .....	3
Подтверждение максимальной производительности.....	5
Тест надёжности.....	6
Промежуточные выводы.....	7
Дополнительные тесты .....	7
Локализация утечки памяти.....	8
Тест без утечки памяти .....	9
Поиск блокировок в сервисе.....	9

# Цели и результаты тестирования

## Цели тестирования

- 1) Определить максимальную производительность.
- 2) Определить надёжность на 100% профиля.

№	Операция	Интенсивность	%
1	Информация о книге	30 в секунду.	75%
2	Изменить цену	6 в секунду	15%
3	Список всех книг	2 в секунду	5%
4	Удалить книгу	1 в секунду	2,5%
5	Добавить книгу	1 в секунду	2,5%

- 3) Определить, где в сервисе возникают блокировки.
- 4) Локализовать утечку памяти, определить, что в ней хранится.

## Результаты тестирования

- 1) Максимальная производительность составляет, около 1040 запросов в секунду.

№	Операция	Интенсивность	%
1	Информация о книге	780 в секунду.	75%
2	Изменить цену	156 в секунду	15%
3	Список всех книг	52 в секунду	5%
4	Удалить книгу	26 в секунду	2,5%
5	Добавить книгу	26 в секунду	2,5%

- 2) Постоянный рост занимаемой памяти, при длительном тесте.
- 3) В сервисе присутствует утечка памяти(`oldBooksPrices` `java.util.concurrent.ConcurrentHashMap`) таблица с данными книг.
- 4) Блокировки возникают на 352 строке кода.

-

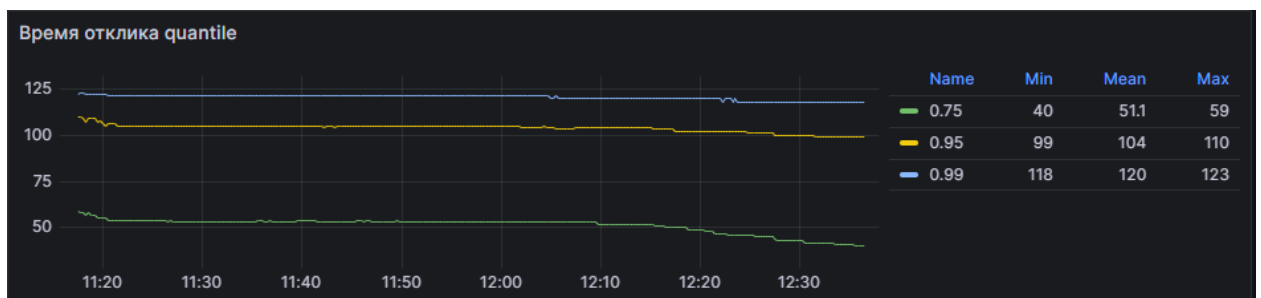
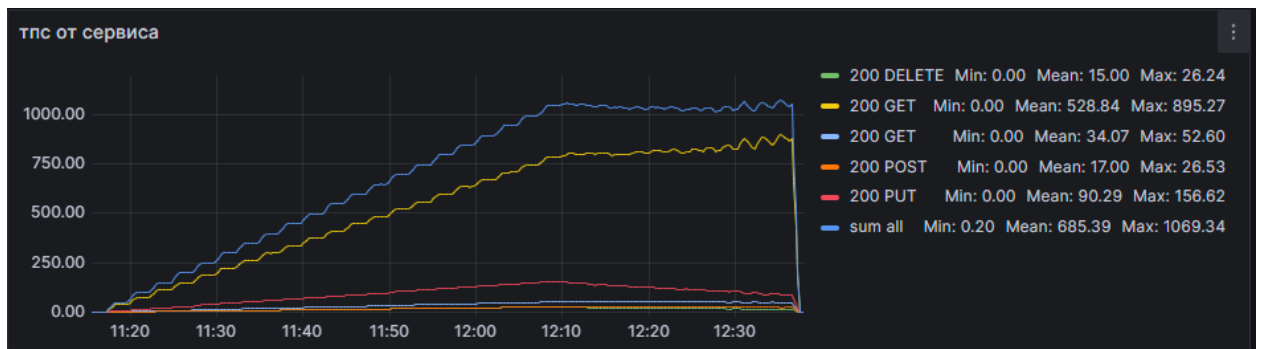
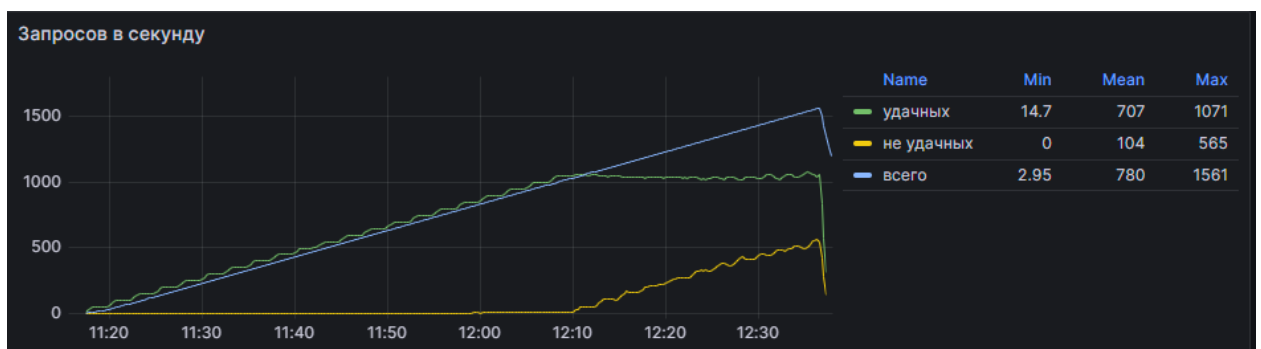
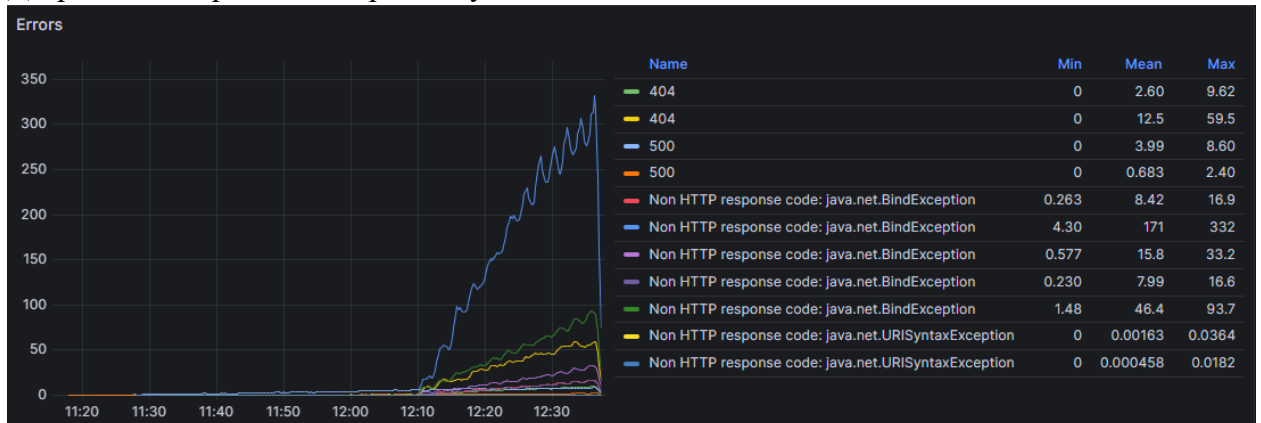
## Проведенные тесты

### Максимальная производительность

При проведении серии тестов с автоматическим управлением количества потоков и интенсивностью операций эквивалентной профилю нагрузки умноженной на коэффициент

N, установлено начало деградации системы при коэффициенте N=26(1040 запросов в секунду)

Деградация выражалась в резком увеличении количества ошибок.



# Подтверждение максимальной производительности

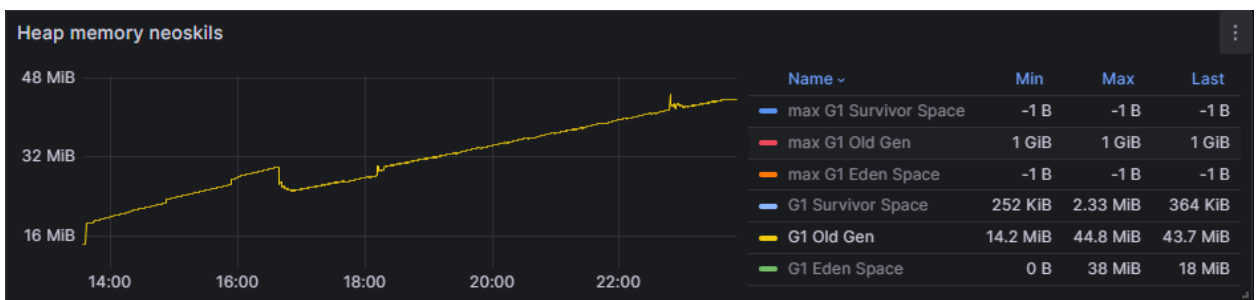
Плавное повышение нагрузки до максимальной(N=26) и фиксация на 1 час.



При выполнении теста с нагрузкой N=26 профиля(1040 запросов в секунду), деградации сервиса не наблюдалось.

# Тест надёжности

Тест проводился при нагрузке 40 запросов в секунду в соответствии с профилем длительное время.



# Промежуточные выводы

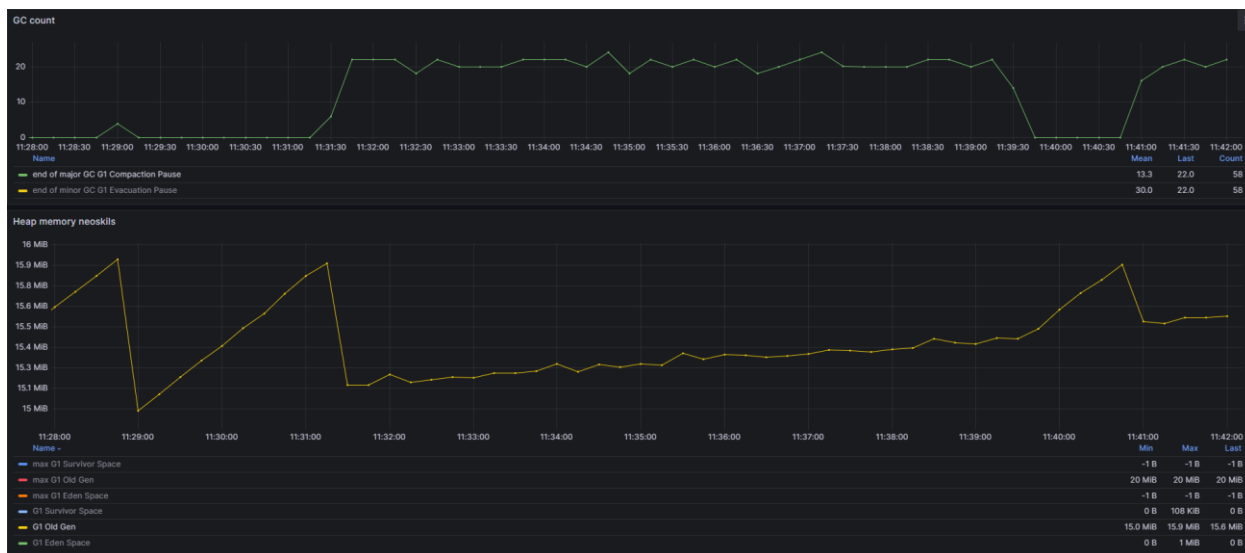
Тестируемый сервис при тестовой нагрузке с автоматическим управлением количеством потоков показал стабильную работу до интенсивности нагрузки в N26(1040 запросов в секунду) включительно

(N = 100% профиля нагрузки)

Тест надёжности показал равномерный рост(без падений) объёма памяти используемого для хранения долгоживущих объектов и массивов

# Дополнительные тесты

Для выявления утечки памяти был повторён тест надёжности с выделенной сервису памятью 20Мб



Иногда, при срабатывании сборки мусора в старом поколении количество памяти, не уменьшается. Что возможно свидетельствует об утечке памяти.

После заполнения памяти, сервис перестал отвечать на запросы.



# Локализация утечки памяти

С помощью Eclipse Memory Analyzer и снимка “heap\_dump” был найден объект создающий утечку.

## Problem Suspect 1

One instance of «**ru.neoflex.skills.performance.service.BookService**» loaded by «**jdk.internal.loader.ClassLoaders\$AppClassLoader @ 0xfed852a8**» occupies **5 586 032 (31,92 %)** bytes. The memory is accumulated in one instance of «**java.util.concurrent.ConcurrentHashMap\$Node[]**», loaded by «**<system class loader>**», which occupies **5 549 040 (31,71 %)** bytes.

Thread «**org.springframework.boot.web.embedded.netty.NettyWebServer\$1 @ 0xff5311e8 server**» has a local variable or reference to «**org.springframework.boot.web.embedded.netty.NettyWebServer @ 0xff4aaa48**» which is on the shortest path to «**java.util.concurrent.ConcurrentHashMap\$Node[65536] @ 0xffc6abb8**». The thread **org.springframework.boot.web.embedded.netty.NettyWebServer\$1 @ 0xff5311e8 server** keeps local variables with total size **256 (0,00 %)** bytes.

The stacktrace of this Thread is available. [See stacktrace.](#) [See stacktrace with involved local variables.](#)

### Keywords

ru.neoflex.skills.performance.service.BookService  
 jdk.internal.loader.ClassLoaders\$AppClassLoader  
 java.util.concurrent.ConcurrentHashMap\$Node[]

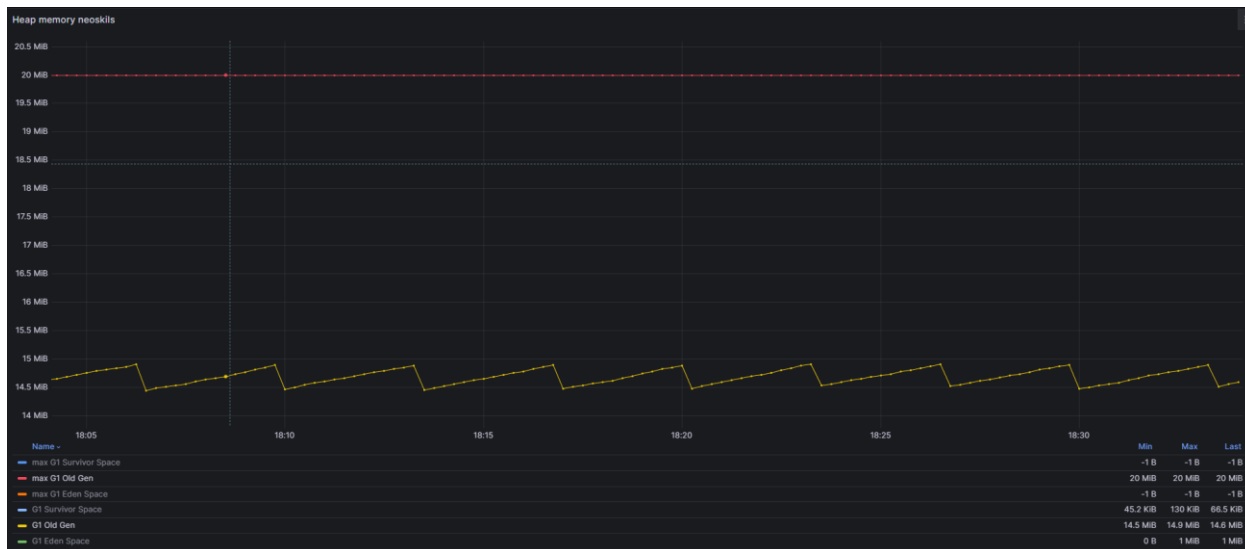
[Details »](#)

Class Name	Shallow Heap	Retained Heap	Percentage
ru.neoflex.skills.performance.service.BookService @ 0xff007300	32	5 586 032	31,92 %
oldBooksPrices java.util.concurrent.ConcurrentHashMap @ 0xff007b98	64	5 549 104	31,71 %
table java.util.concurrent.ConcurrentHashMap\$Node[65536] @ 0xffc6abb8	262 160	5 549 040	31,71 %

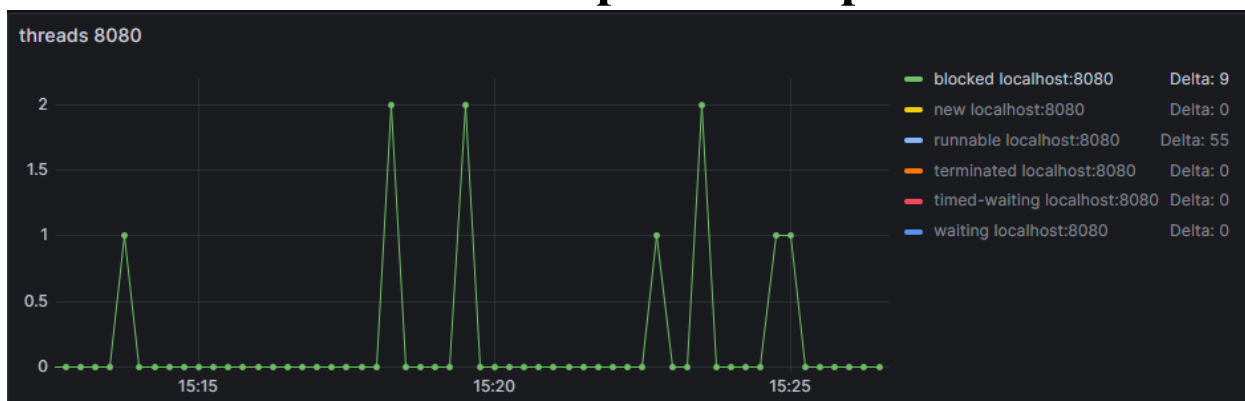


## Тест без утечки памяти

Тест надёжности был повторён после добавлении строки “memory.leak.enabled: false” в конфигурационный файл сервиса. Утечки памяти не наблюдается.



## Поиск блокировок в сервисе



С помощью утилиты jcmd был найден идентификатор тестируемого приложения.

Командой jcmd «идентификатор» Thread.print вывели в консоль необходимые запросы,

По ключевому слову “BLOCKED” были найдены заблокированные потоки.

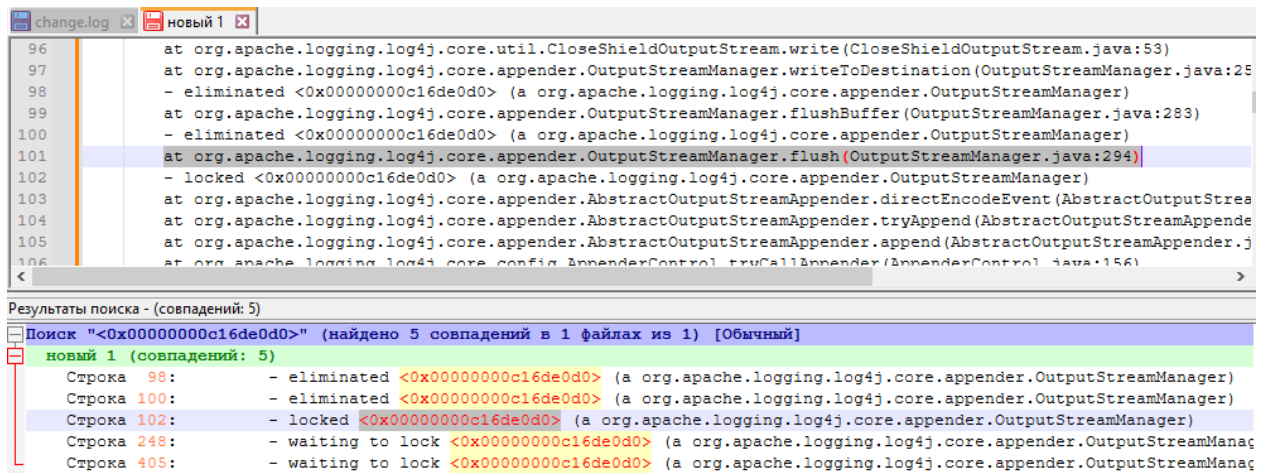
```
change.log x  новый 1 x
243 java.lang.Thread.State: RUNNABLE
244
245 "BookThread-2" #35 [11040] daemon prio=10 os_prio=2 cpu=296859.38ms elapsed=6791.16s tid=0x000002b7edb5b090 nid=1104
246 java.lang.Thread.State: BLOCKED (on object monitor)
247 at org.apache.logging.log4j.core.appender.OutputStreamManager.writeBytes(OutputStreamManager.java:352)
248 - waiting to lock <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)
249 at org.apache.logging.log4j.core.layout.TextEncoderHelper.writeEncodedText(TextEncoderHelper.java:96)
250 at org.apache.logging.log4j.core.layout.TextEncoderHelper.encodeText(TextEncoderHelper.java:65)
251 at org.apache.logging.log4j.core.layout.StringBuilderEncoder.encode(StringBuilderEncoder.java:68)
252 at org.apache.logging.log4j.core.layout.StringBuilderEncoder.encode(StringBuilderEncoder.java:32)

Результаты поиска - (совпадений: 2)
Поиск "blocked" (найдено 2 совпадений в 1 файлах из 1) [Обычный]
новый 1 (совпадений: 2)
Строка 246: java.lang.Thread.State: BLOCKED (on object monitor)
Строка 403: java.lang.Thread.State: BLOCKED (on object monitor)
Поиск "BLOCKED" (найдено 0 совпадений в 0 файлах из 1) [Обычный]
```

Поток был заблокирован на 352 строке кода

Он ждёт освобождения объекта <0x00000000c16de0d0>

Занятого в строке 294



The screenshot shows an IDE window with a Java stack trace and search results. The stack trace is as follows:

```
96 at org.apache.logging.log4j.core.util.CloseShieldOutputStream.write(CloseShieldOutputStream.java:53)
97 at org.apache.logging.log4j.core.appender.OutputStreamManager.writeToDestination(OutputStreamManager.java:25)
98 - eliminated <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)
99 at org.apache.logging.log4j.core.appender.OutputStreamManager.flushBuffer(OutputStreamManager.java:283)
100 - eliminated <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)
101 at org.apache.logging.log4j.core.appender.OutputStreamManager.flush(OutputStreamManager.java:294)
102 - locked <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)
103 at org.apache.logging.log4j.core.appender.AbstractOutputStreamAppender.directEncodeEvent(AbstractOutputStreamAppender.java:179)
104 at org.apache.logging.log4j.core.appender.AbstractOutputStreamAppender.tryAppend(AbstractOutputStreamAppender.java:174)
105 at org.apache.logging.log4j.core.appender.AbstractOutputStreamAppender.append(AbstractOutputStreamAppender.java:164)
106 at org.apache.logging.log4j.core.config.AppenderControl.tryCallAppender(AppenderControl.java:156)
```

Below the stack trace, the search results are displayed:

Результаты поиска - (совпадений: 5)  
Поиск "<0x00000000c16de0d0>" (найдено 5 совпадений в 1 файлах из 1) [Обычный]  
новый 1 (совпадений: 5)  
Строка 98: - eliminated <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)  
Строка 100: - eliminated <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)  
Строка 102: - locked <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)  
Строка 248: - waiting to lock <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)  
Строка 405: - waiting to lock <0x00000000c16de0d0> (a org.apache.logging.log4j.core.appender.OutputStreamManager)