

Отчёт по нагрузочному тестированию сервиса “book”

Оглавление

Проведенные тесты.....	3
Максимальная производительность	3
Подтверждение максимальной производительности.....	4
Тест надёжности.....	5
Промежуточные выводы.....	6
Дополнительные тесты	7
Локализация утечки памяти.....	8
Тест без утечки памяти	10
Поиск блокировок в сервисе.....	11

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.

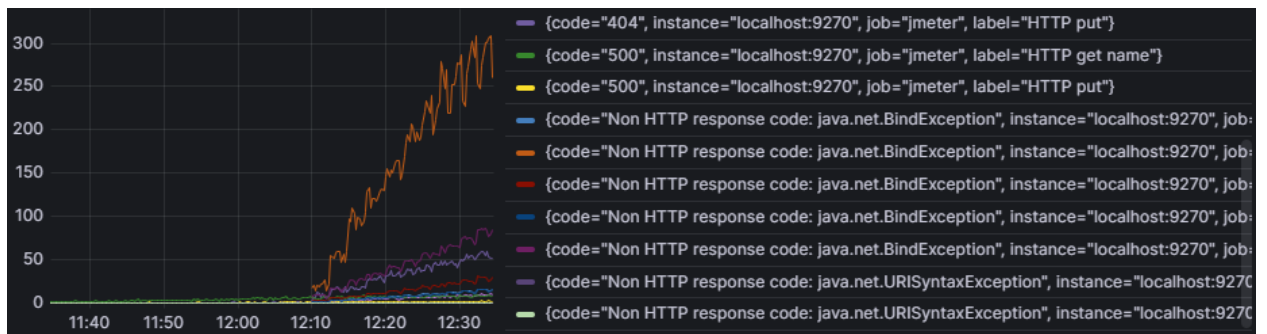


График 1 Не успешных запросов с кодом ошибки

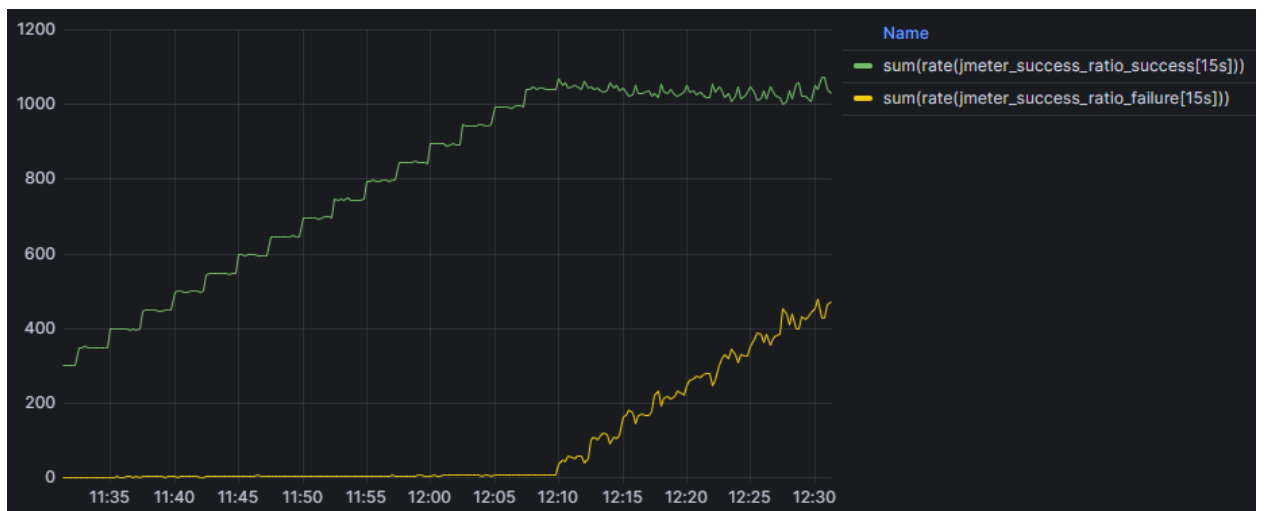
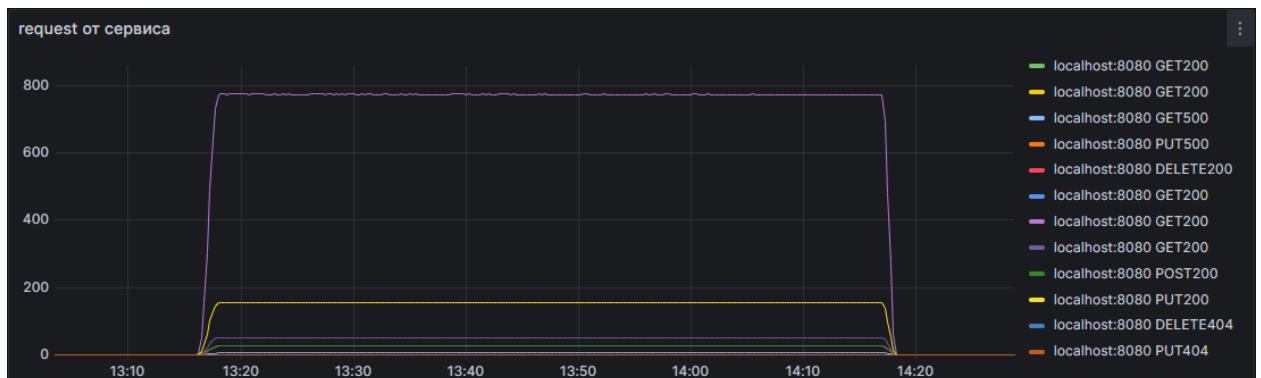


График 2 Количество успешных и не успешных запросов

Подтверждение максимальной производительности

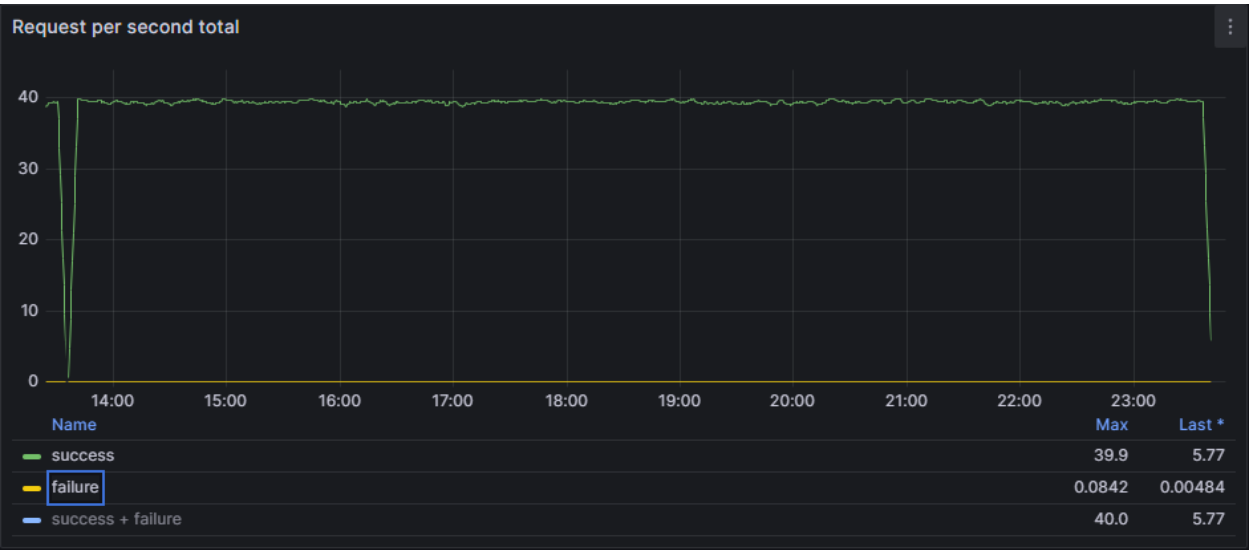
Плавное повышение нагрузки до максимальной(N=26) и фиксация на 1 час.



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 @ 0xf007300	32	5 586 032	31,92 %
oldBooksPrices java.util.concurrent.ConcurrentHashMap @ 0xf007b98	64	5 549 104	31,71 %
table java.util.concurrent.ConcurrentHashMap\$Node[65536] @ 0xffc6abb8	262 160	5 549 040	31,71 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffe6d38	32	752	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffe2750	32	672	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff83c6e	32	672	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff83ca0	32	672	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff8d368	32	672	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffc0f98	32	672	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff91668	32	640	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffa3378	32	640	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffc0198	32	640	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff84c6c	32	624	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffa7340	32	616	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff84518	32	616	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffa0398	32	600	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffc4888	32	600	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffe4880	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xffe6340	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff73ab30	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff78838	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff78838	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff79940	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff79948	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff7c4d8	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff7c4d0	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff7db0a	32	560	0,00 %
java.util.concurrent.ConcurrentHashMap\$Node @ 0xff18198	32	560	0,00 %
books java.util.Collections\$SynchronizedSet @ 0xf007320	24	36 840	0,21 %
rnd java.util.Random @ 0xf007bd8	32	56	0,00 %
Total: 3 entries			
jdk.internal.loader.ClassLoaders\$AppClassLoader @ 0xfed852a8 JNI Global	96	1 517 536	8,67 %
org.springframework.beans.factory.support.DefaultListableBeanFactory @ 0xf293b78	232	1 214 744	6,94 %
jdk.internal.loader.ClassLoaders\$PlatformClassLoader @ 0xfed90a80 JNI Global	96	331 496	1,89 %
class java.lang.invoke.MethodType @ 0xfed6c4d0 System Class	40	237 216	1,36 %
org.springframework.boot.autoconfigure.condition.ConditionEvaluationReport @ 0xf2	32	230 584	1,32 %
class java.time.zone.ZoneRulesProvider @ 0xf216798 System Class	16	204 088	1,17 %
java.util.zip.ZipFileSource @ 0xfed8b00	80	198 272	1,13 %
java.util.zip.ZipFileSource @ 0xfed44018	80	175 360	1,00 %
class sun.util.locale.BaseLocale\$Cache @ 0xf479488 System Class	8	158 888	0,91 %
java.util.zip.ZipFileSource @ 0xfed6c3d0	80	150 312	0,86 %
class sun.util.calendar.ZoneInfoFile @ 0xfedc7a10 System Class	120	149 976	0,86 %
java.util.zip.ZipFileSource @ 0xfed6c3d0	80	138 072	0,78 %
java.util.zip.ZipFileSource @ 0xfedc7638	80	134 456	0,77 %
java.util.zip.ZipFileSource @ 0xfed4e968	80	123 976	0,71 %
java.util.zip.ZipFileSource @ 0xfed6c3d0	80	123 528	0,71 %

ru.neoflex.skills.performance.service.BookService @ 0xf007300	32	5 586 032	31,92 %
oldBooksPrices java.util.concurrent.ConcurrentHashMap @ 0xf007b98	64	5 549 104	31,71 %
table java.util.concurrent.ConcurrentHashMap\$Node[65536] @ 0xffc6abb8	262 160	5 549 040	31,71 %

Анализ heap dump в другом случае

One instance of «**ru.neoflex.skills.performance.service.BookService**» loaded by «**jdk.internal.loader.ClassLoaders\$AppClassLoader @ 0xc017e460**» occupies **813 116 400 (98,34 %)** bytes. The memory is accumulated in one instance of «**java.util.concurrent.ConcurrentHashMap\$Node[]**», loaded by «**<system class loader>**», which occupies **810 995 152 (98,08 %)** bytes.

Thread «**org.springframework.boot.web.embedded.netty.NettyWebServer\$1 @ 0xc1308610 server**» has a local variable or reference to «**org.springframework.boot.web.embedded.netty.NettyWebServer @ 0xc19b0818**» which is on the shortest path to «**java.util.concurrent.ConcurrentHashMap\$Node[16777216] @ 0xf5300000**». The thread **org.springframework.boot.web.embedded.netty.NettyWebServer\$1 @ 0xc1308610 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 »](#)

The screenshot shows the Eclipse Memory Analyzer (MAT) interface. The 'dominator_tree' tab is active, displaying a list of heap objects. The top entries are:

Class Name	Shallow Heap	Retained Heap	Percentage
ru.neoflex.skills.performance.service.BookService @ 0xc17b3f28	32	813 116 400	98,34 %
oldBooksPrices java.util.concurrent.ConcurrentHashMap @ 0xc17e1ed8	64	810 995 288	98,08 %
table java.util.concurrent.ConcurrentHashMap\$Node[16777216] @ 0xf5300000	67 108 880	810 995 152	98,08 %

The 'table' entry is highlighted, showing its details in the 'Attributes' tab. The 'Attributes' tab shows the 'table' entry with its 'Value' field set to 'table java.util.concurrent.ConcurrentHashMap\$Node[16777216] @ 0xf5300000'.

The 'Heap Dump History' tab is also visible, showing a list of heap dumps. The 'Recently Used Files' list includes:

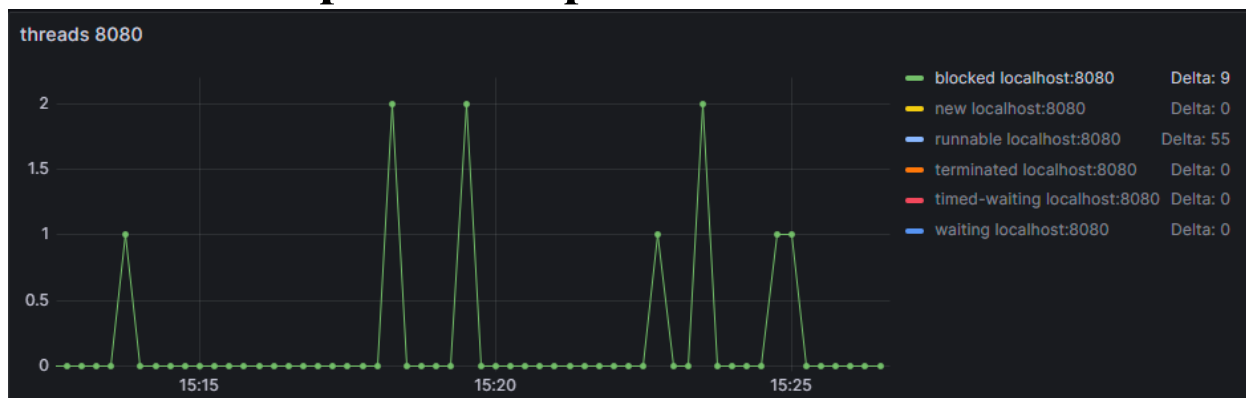
- D:\dump21.hprof
- D:\dump22.hprof
- D:\dump.hprof
- D:\heap20.hprof
- D:\dump15.hprof
- D:\dump14.hprof
- D:\dump13.hprof
- D:\dump12.hprof
- D:\dump11.hprof
- D:\dump.hprof10
- D:\dump8.hprof
- D:\dump7.hprof
- D:\dump6.hprof
- D:\dump5.hprof
- D:\dump2.hprof
- D:\dump1.hprof

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

новый 1

```
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

change.log

новый 1

```
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:170)
105     at org.apache.logging.log4j.core.appender.AbstractOutputStreamAppender.append(AbstractOutputStreamAppender.java:156)
106     at org.apache.logging.log4j.core.config.AppenderControl.tryCallAppender(AppenderControl.java:156)
```

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)