



# Hibernate, OOM и ооочень длинные запросы

Синдеев Сергей

Java Developer

[Sergei\\_Sindeev@rntgroup.com](mailto:Sergei_Sindeev@rntgroup.com)

# Иногда они падают...

## Basic Statistics

Uptime

2.0 min

Start time

2024-01-15 16:41:51

Heap Used



Non-He:

CPU Usage



Исходная точка:

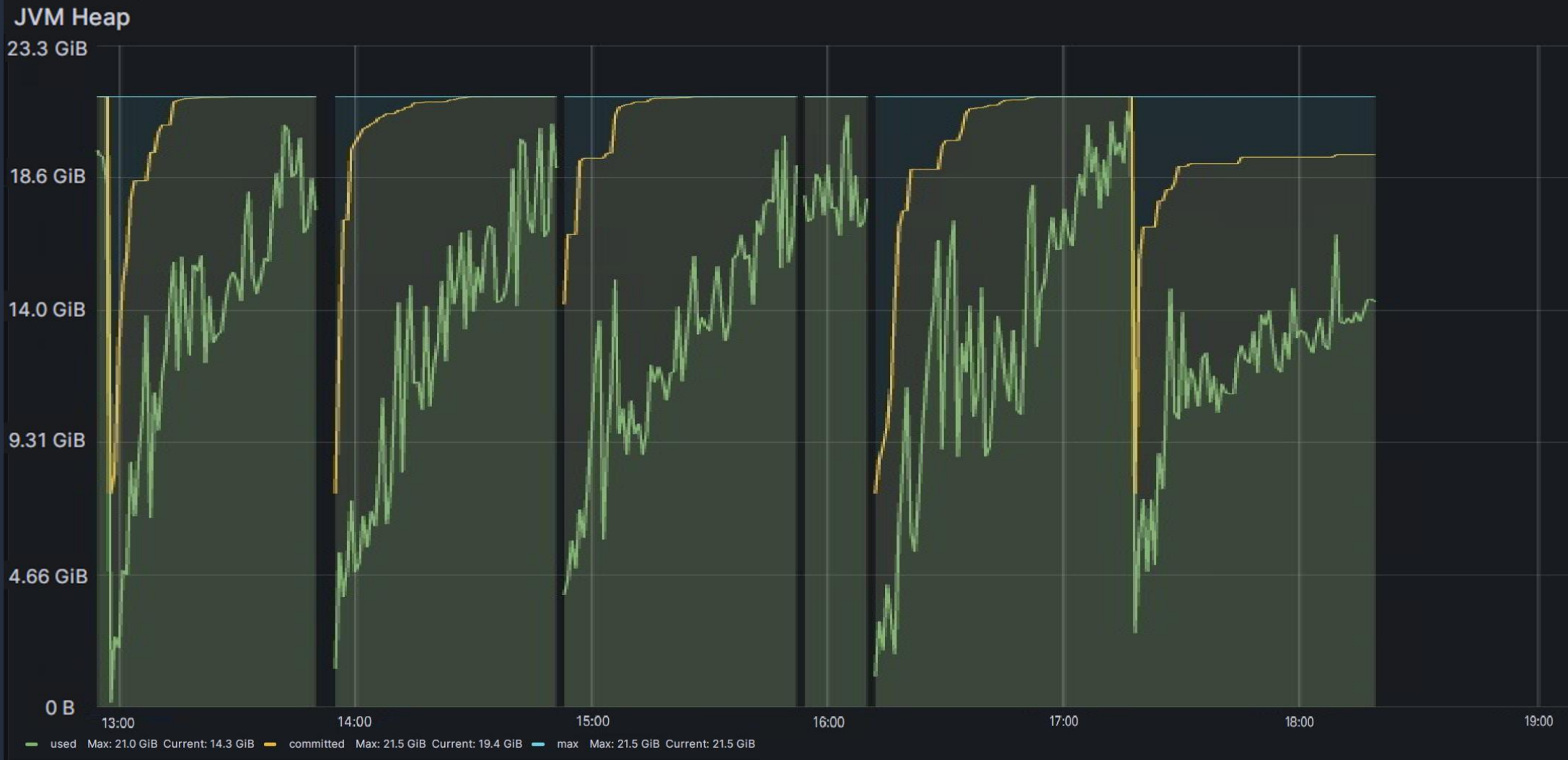
- микросервис с 22Gb RAM
- падения с переполнением памяти (OOM) в среднем каждый час
- паттерн воспроизводимости

Memory Usage

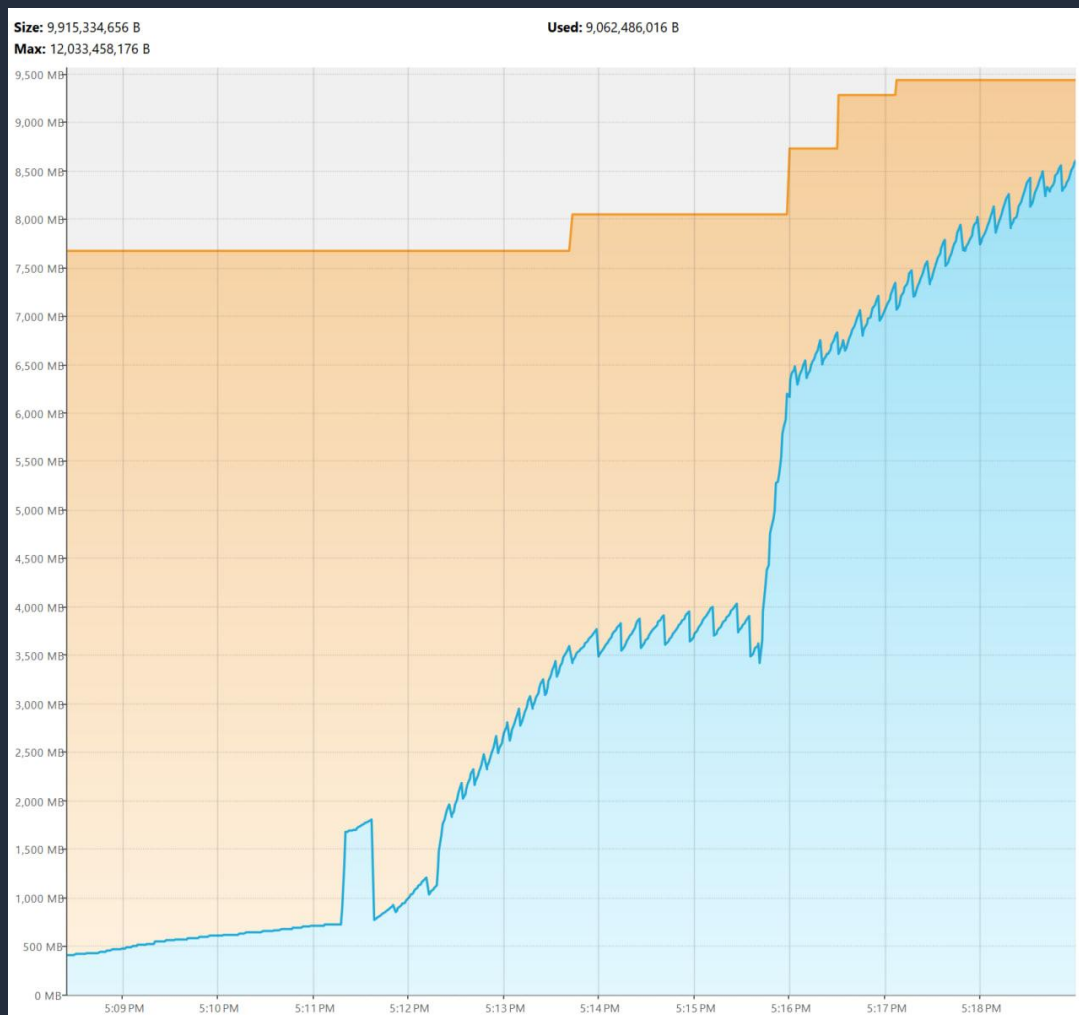


# Возможные причины

## #1 "Испорченный" scheduler



## #2 JMX



## #3 Логи



elastic

# Что такое дамп

[heapdump] heapdump.hprof

Heap Dum...

Objects | Preset: All Objects | Aggregation: | Details: Preview Fields References GC Root Hierarchy

Name	Count	Size	Retained
java.util.concurrent. <b>ConcurrentHashMap</b>	6,044 (0.2%)	386,816 B (0.2%)	47,016,640 B (27.5%)
java.util.concurrent. <b>ConcurrentHashMap\$Node[]</b>	3,013 (0.1%)	2,512,256 B (1.5%)	46,682,720 B (27.3%)
org.hibernate.internal. <b>SessionFactoryImpl</b>	1 (0%)	112 B (0%)	46,326,216 B (27.1%)
org.hibernate.engine.query.spi. <b>QueryPlanCache</b>	1 (0%)	32 B (0%)	46,314,944 B (27.1%)
org.hibernate.internal.util.collections. <b>BoundedConcurrentHashMap</b>	2 (0%)	96 B (0%)	46,314,912 B (27.1%)
org.hibernate.internal.util.collections. <b>BoundedConcurrentHashMap\$Segment[]</b>	2 (0%)	288 B (0%)	46,314,816 B (27.1%)
org.hibernate.internal.util.collections. <b>BoundedConcurrentHashMap\$Segment</b>	64 (0%)	3,072 B (0%)	46,306,784 B (27.1%)
org.hibernate.internal.util.collections. <b>BoundedConcurrentHashMap\$LIRSHashEntry</b>	2,132 (0.1%)	119,392 B (0.1%)	46,269,520 B (27.1%)
org.hibernate.engine.query.spi. <b>HQLQueryPlan</b>	2,038 (0.1%)	97,824 B (0.1%)	45,969,464 B (26.9%)
java.util.concurrent. <b>ConcurrentHashMap\$Node</b>	275,150 (7%)	8,804,800 B (5.2%)	43,594,432 B (25.5%)
org.hibernate.hql.spi. <b>QueryTranslator[]</b>	2,038 (0.1%)	48,912 B (0%)	39,947,800 B (23.4%)
org.hibernate.hql.internal.ast. <b>QueryTranslatorImpl</b>	2,038 (0.1%)	130,432 B (0.1%)	39,898,888 B (23.3%)
java.lang. <b>String</b>	362,383 (9.2%)	8,697,192 B (5.1%)	36,572,016 B (21.4%)
org.hibernate.hql.internal.ast.tree. <b>QueryNode</b>	2,038 (0.1%)	163,040 B (0.1%)	33,546,760 B (19.6%)
<b>byte[]</b>	376,971 (9.5%)	34,411,128 B (20.1%)	31,659,504 B (18.5%)
java.lang. <b>Object[]</b>	54,665 (1.4%)	7,977,488 B (4.7%)	20,935,040 B (12.3%)
org.hibernate.hql.internal.ast. <b>HqlSqlWalker</b>	2,038 (0.1%)	309,776 B (0.2%)	20,927,872 B (12.2%)
java.util. <b>HashMap\$Node[]</b>	59,143 (1.5%)	4,650,088 B (2.7%)	19,180,392 B (11.2%)
java.util. <b>HashMap</b>	39,408 (1%)	1,891,584 B (1.1%)	19,090,136 B (11.2%)
org.hibernate.hql.internal.ast. <b>HqlParser</b>	2,038 (0.1%)	114,128 B (0.1%)	19,083,096 B (11.2%)

# И как его получить?..

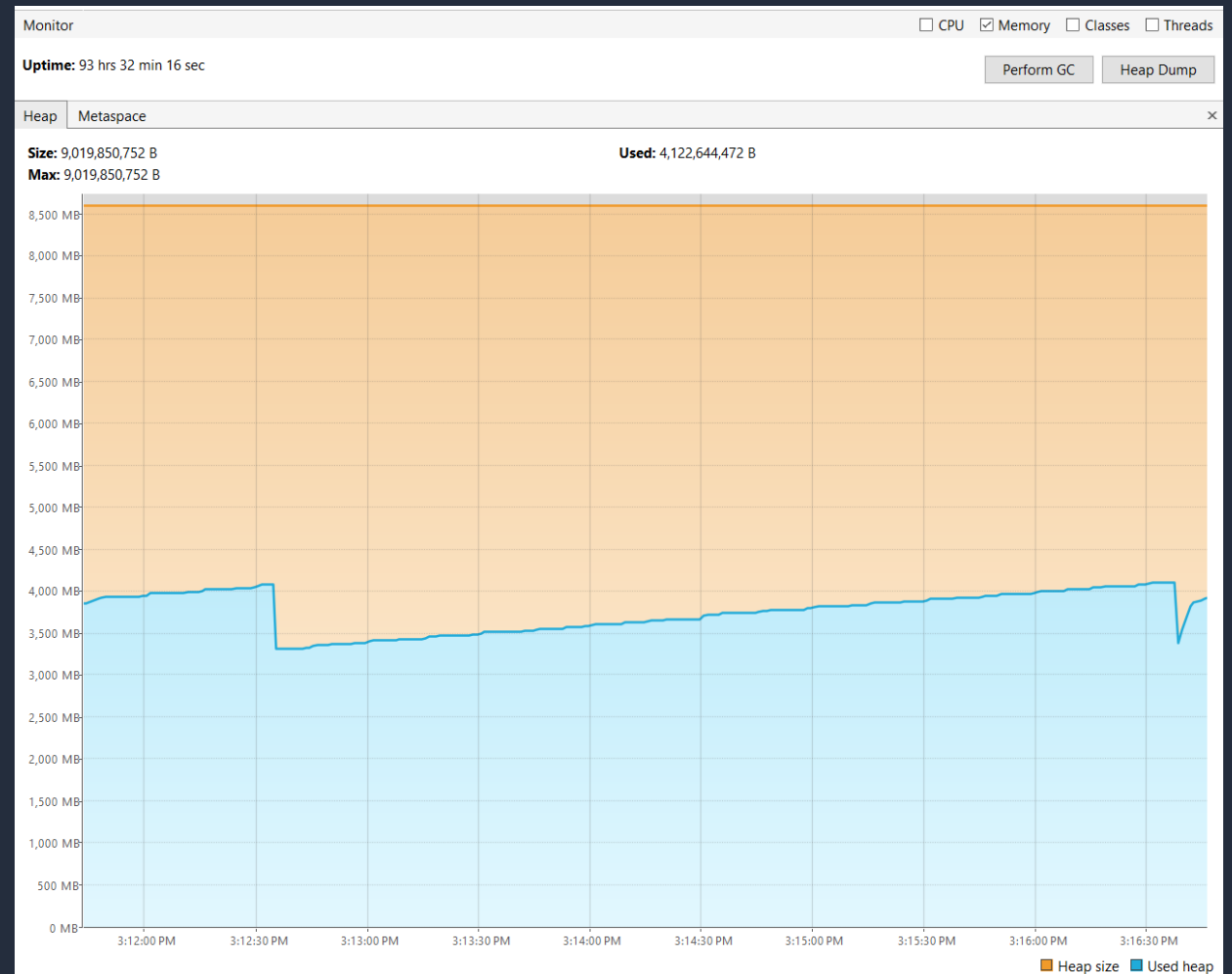
## #1 VM keys



JVM argume... System properties

```
-XX:+ExitOnOutOfMemoryError
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/opt/dumps
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5005
-Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.port=9010
-Dcom.sun.management.jmxremote.rmi.port=9010
-Djava.rmi.server.hostname=127.0.0.1
-XX:MaxRAMPercentage=70
```

## #2 JMX



## #3 Actuator

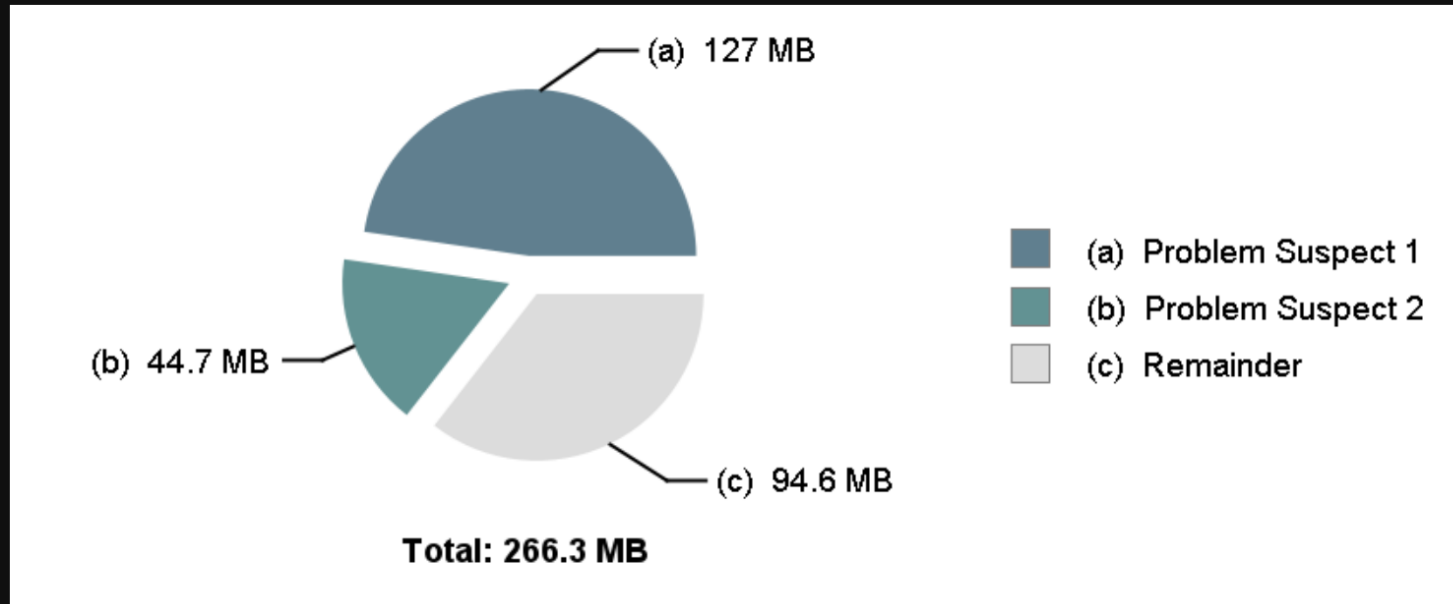


```
<dependency>
  ... <groupId>org.springframework.boot</groupId>
  ... <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

```
management:
  endpoints:
    web:
      exposure:
        include: heapdump
```

```
▼ _links:
  ▼ self:
    ▼ href:      "https://localhost:8080/actuator"
      templated: false
  ▼ health:
    ▼ href:      "https://localhost:8080/actuator/health"
      templated: false
  ▼ heapdump:
    ▼ href:      "https://localhost:8080/actuator/heapdump"
      templated: false
  ▼ prometheus:
    ▼ href:      "https://localhost:8080/actuator/prometheus"
      templated: false
  ▼ metrics:
    ▼ href:      "https://localhost:8080/actuator/metrics"
      templated: false
```

# Разбираем дамп



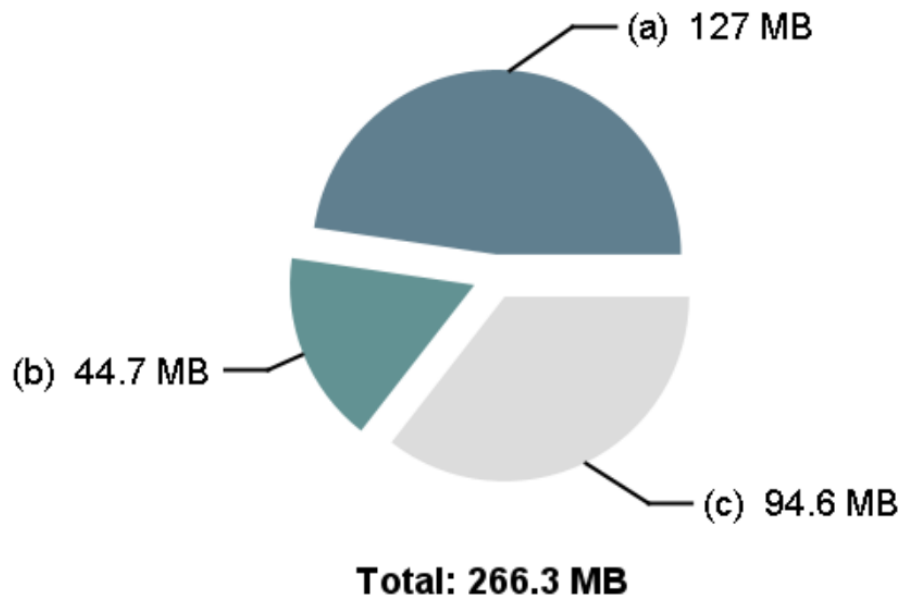
## ❌ Problem Suspect 1

One instance of `org.hibernate.stat.internal.StatisticsImpl` loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c00000` occupies **133,162,432 (47.68%)** bytes. The memory is accumulated in one instance of `java.util.concurrent.ConcurrentHashMap$Node[]`, loaded by `<system class loader>`, which occupies **133,146,968 (47.68%)** bytes.

## ❌ Problem Suspect 2

One instance of `org.hibernate.internal.SessionFactoryImpl` loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c00000` occupies **46,905,624 (16.80%)** bytes. The memory is accumulated in one instance of `org.hibernate.internal.util.collections.BoundedConcurrentHashMap$Segment[]`, loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c00000`, which occupies **46,864,960 (16.78%)** bytes.





Eclipse Memory Analyzer Tool  
показывает 2 подозрительных объекта:

Problem Suspect 1

One instance of **"org.hibernate.stat.internal.StatisticsImpl"** occupies 133,162,432 (47.68%) bytes. The memory is accumulated in one instance of **"org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c0000"** occupies 133,146,968 (47.68%) bytes.

- org.hibernate.stat.internal.StatisticsImpl

Problem Suspect 2

One instance of **"org.hibernate.internal.SessionFactoryImpl"** occupies 46,905,624 (16.80%) bytes. The memory is accumulated in one instance of **"org.hibernate.internal.util.collections.BoundedConcurrentHashMap\$Segment []"**, loaded by **"org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c0000"**, which occupies 46,864,960 (16.78%) bytes.

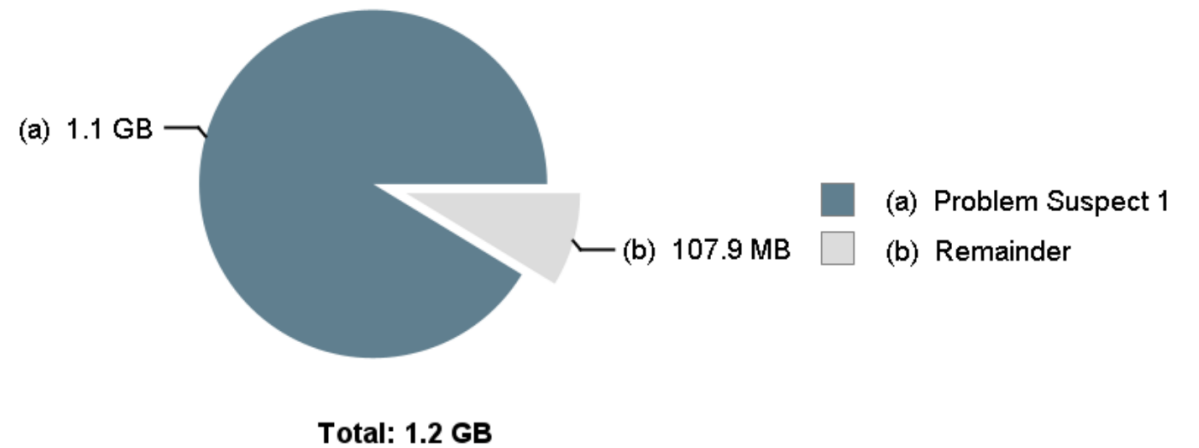
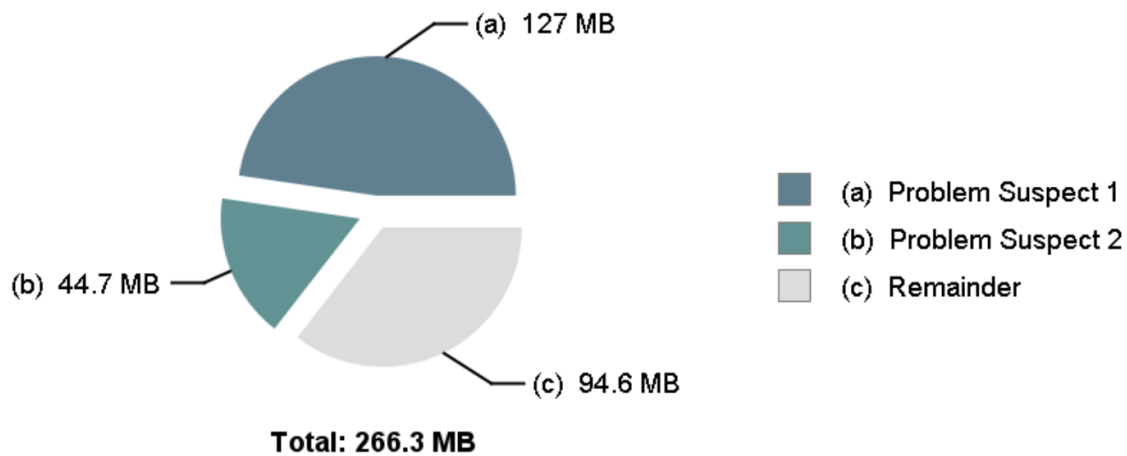
- org.hibernate.internal.SessionFactoryImpl

# Немножко гуглим...

Class Name	Shallow Heap	Retained Heap
<Regex>	<Numeric>	<Numeric>
org.hibernate.internal.SessionFactoryImpl @ 0x5344f2070	112	46,905,624
statistics org.hibernate.stat.internal.StatisticsImpl @ 0x535cf1ba8	208	133,162,432
queryStatsMap java.util.concurrent.ConcurrentHashMap @ 0x535cf2188	64	133,147,032
table java.util.concurrent.ConcurrentHashMap\$Node[524288] @ 0x547800000	2,097,168	133,146,968
[167153] java.util.concurrent.ConcurrentHashMap\$Node @ 0x53b2eda40	32	3,312
next java.util.concurrent.ConcurrentHashMap\$Node @ 0x53d1ac278	32	2,624
val org.hibernate.stat.internal.QueryStatisticsImpl @ 0x53b2edb70	56	384
<class> class org.hibernate.stat.internal.QueryStatisticsImpl @ 0x535ae6a30	8	968
query java.lang.String @ 0x53b2eda60 select generatedAlias0 from Account as generatedAlias0	24	272
cacheHitCount java.util.concurrent.atomic.LongAdder @ 0x53b2edba8	32	32
cacheMissCount java.util.concurrent.atomic.LongAdder @ 0x53b2edbc8	32	32
cachePutCount java.util.concurrent.atomic.LongAdder @ 0x53b2edbe8	32	32
executionCount java.util.concurrent.atomic.LongAdder @ 0x53b2edc08	32	32
executionRowCount java.util.concurrent.atomic.LongAdder @ 0x53b2edc28	32	32
executionMaxTime java.util.concurrent.atomic.AtomicLong @ 0x53b2edc48 29	24	24
executionMinTime java.util.concurrent.atomic.AtomicLong @ 0x53b2edc60 5	24	24
totalExecutionTime java.util.concurrent.atomic.AtomicLong @ 0x53b2edc78 62	24	24
readLock java.util.concurrent.locks.ReentrantReadWriteLock\$ReadLock @ 0x53b2edc90	16	16
writeLock java.util.concurrent.locks.ReentrantReadWriteLock\$WriteLock @ 0x53b2edce0	16	16
Total: 12 entries		
key java.lang.String @ 0x53b2eda60 select generatedAlias0 from Account as generatedAlias0 wher	24	272
<class> class java.util.concurrent.ConcurrentHashMap\$Node @ 0x7bfa2d8f0 System Class	0	0
Total: 4 entries		
[206342] java.util.concurrent.ConcurrentHashMap\$Node @ 0x53dec81d8	32	3,312

Переключаем настройку

```
spring:
  jpa:
    properties:
      hibernate:
        generate_statistics: false
```



#### Problem Suspect 1

One instance of `org.hibernate.stat.internal.StatisticsImpl` loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c00000` occupies **133,162,432 (47.68%)** bytes. The memory is accumulated in one instance of `java.util.concurrent.ConcurrentHashMap$Node[]`, loaded by `<system class loader>`, which occupies **133,146,968 (47.68%)** bytes.

#### Problem Suspect 2

One instance of `org.hibernate.internal.SessionFactoryImpl` loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c00000` occupies **46,905,624 (16.80%)** bytes. The memory is accumulated in one instance of `org.hibernate.internal.util.collections.BoundedConcurrentHashMap$Segment[]`, loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x532c00000`, which occupies **46,864,960 (16.78%)** bytes.

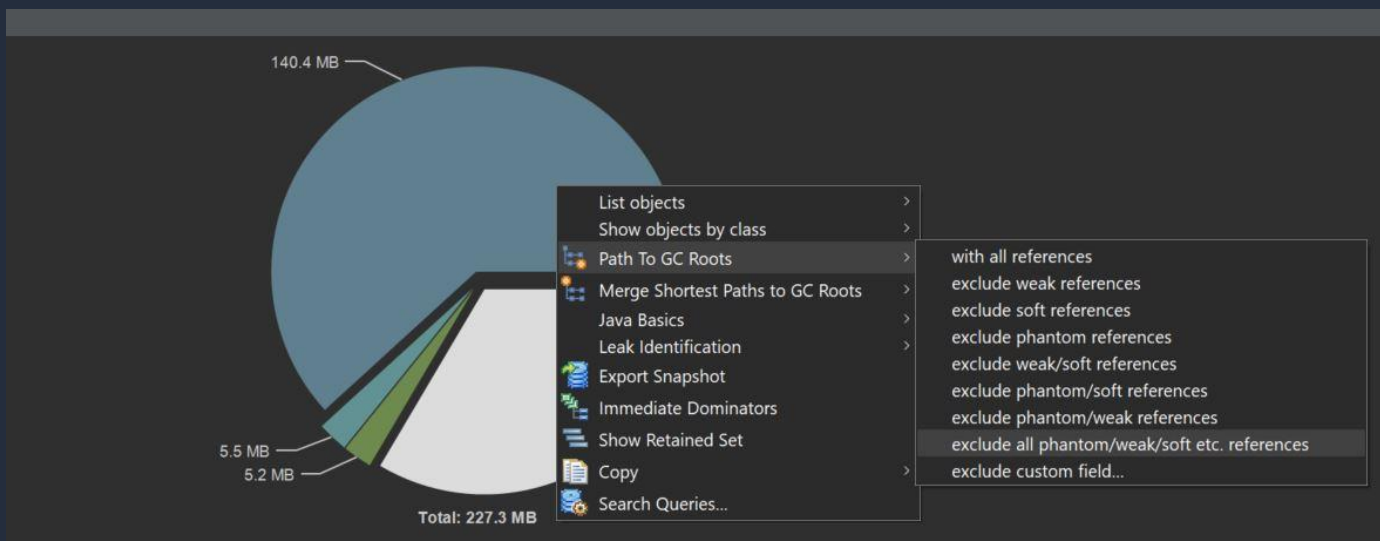
#### Problem Suspect 1

One instance of `org.hibernate.internal.SessionFactoryImpl` loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x2a1000080` occupies **1,188,324,712 (91.31%)** bytes. The memory is accumulated in one instance of `org.hibernate.hql.internal.ast.QueryTranslatorImpl`, loaded by `org.springframework.boot.loader.LaunchedURLClassLoader @ 0x2a1000080`, which occupies **649,526,968 (49.91%)** bytes.

Thread `java.lang.Thread @ 0x2a17e6ee8 qtp1106189986-1565` has a local variable or reference to `org.hibernate.persister.collection.BasicCollectionPersister @ 0x2a3e89888` which is on the shortest path to `org.hibernate.hql.internal.ast.QueryTranslatorImpl @ 0x2b8327ea0`. The thread `java.lang.Thread @ 0x2a17e6ee8 qtp1106189986-1565` keeps local variables with total size **46,304 (0.00%)** bytes.

# Теперь разберёмся со вторым виновником

step #1



step #3

Class Name	Shallow Heap	Retained Heap
<Regex>	<Numeric>	<Numeric>
org.hibernate.internal.SessionFactoryImpl @ 0x2a2bc158f	128	1,188,324,712
<class> class org.hibernate.internal.SessionFactoryImpl	16	3,976
uuid java.lang.String @ 0x2a2bc1608 805fa8e5-3ec9-	24	80
observer org.hibernate.internal.SessionFactoryObserv	16	200
sessionFactoryOptions org.hibernate.boot.internal.Se	208	632
metamodel org.hibernate.metamodel.internal.Metam	72	3,309,456
jdbcServices org.hibernate.engine.jdbc.internal.JdbcS	32	48
serviceRegistry org.hibernate.service.internal.Sessionl	56	1,536
cacheAccess org.hibernate.cache.internal.EnabledCac	64	1,456
fastSessionServices org.hibernate.internal.FastSessio	152	504
settings org.hibernate.cfg.Settings @ 0x2a41e3b20	24	24
properties java.util.HashMap @ 0x2a41e3b38	48	3,424
sqlFunctionRegistry org.hibernate.dialect.function.SC	16	3,424
criteriaBuilder org.hibernate.query.criteria.internal.Cri	16	16
jpaPersistenceUnitUtil org.hibernate.jpa.internal.Pers	24	216
queryPlanCache org.hibernate.engine.query.spi.Quer	32	1,188,313,264
<class> class org.hibernate.engine.query.spi.Query	16	1,088
factory org.hibernate.internal.SessionFactoryImpl @	128	1,188,324,712
nativeQueryInterpreter org.hibernate.engine.quer	16	16
queryPlanCreator org.hibernate.boot.internal.Sess	16	16
queryPlanCache org.hibernate.internal.util.collecti	48	1,188,294,072
<class> class org.hibernate.internal.util.collecti	40	168
segments org.hibernate.internal.util.collections.	144	1,188,294,024
class org.hibernate.internal.util.collections.Bo	0	0
[31] org.hibernate.internal.util.collections.Boi	48	176,480
[30] org.hibernate.internal.util.collections.Boi	48	149,496
[29] org.hibernate.internal.util.collections.Boi	48	376
[28] org.hibernate.internal.util.collections.Boi	48	69,216
[27] org.hibernate.internal.util.collections.Boi	48	82,464
[26] org.hibernate.internal.util.collections.Boi	48	66,928
[25] org.hibernate.internal.util.collections.Boi	48	173,352
[24] org.hibernate.internal.util.collections.Boi	48	213,944
[23] org.hibernate.internal.util.collections.Boi	48	136,546,056

step #2

Status: Found 30 paths so far.

Class Name	Shallow Heap	Retained Heap
<Regex>	<Numeric>	<Numeric>
org.hibernate.internal.SessionFactoryImpl @ 0x5e84a94e0	128	1,188,324,712

- List objects
- Show objects by class
- Path To GC Roots
- Merge Shortest Paths to GC Roots
- Java Basics
- Leak Identification
- Export Snapshot

- with outgoing references
- with incoming references

## step #4

```
SELECT l.query.toString() FROM INSTANCEOF org.hibernate.engine.query.spi.QueryPlanCache$HQLQueryPlanKey l
```

l.query.toString()

```
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.personalNumber, generatedAlias0.username, generatedAlias0.displayName from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, ...
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6, :param7, :param8, :param9, :param10, :param1...
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6, :param7, :param8, :param9, :param10, :param1...
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6, :param7, :param8, :param9, :param10) )
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6, :param7, :param8, :param9) )
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6, :param7, :param8) )
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6, :param7) )
select generatedAlias0.id, generatedAlias0.name from PortalGroup as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( generatedAlias0.id in (:param1, :param2, :param3, :param4, :param5, :param6) )
```

## step #5

```
org.hibernate.internal.util.collections.BoundedConcurrentHashMap$LIRSHashEntry @ 0x2aedf7b00 56
> value org.hibernate.engine.query.spi.HQLQueryPlan @ 0x2aedec3c0 40
java.lang.String @ 0x2aedec3a8 select generatedAlias0.displayName, generatedAlias0.id from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and ( ( gener. 24
  value byte[19126406] @ 0x306400000 select generatedAlias0.displayName, generatedAlias0.id from Account as generatedAlias0 where ( generatedAlias0.actual=:param0 ) and 19,126,424
  key org.hibernate.engine.query.spi.QueryPlanCache$HQLQueryPlanKey @ 0x2aedec3e8 32
```

~19Mb

## На пути к исправлению

```
public List<Item> findAllById(Collection<Long> veryLongCollection) {
    if (veryLongCollection.size() > IDS_INPUT_THRESHOLD) {
        log.warn("Too much ids requested - {}", veryLongCollection.size());
    }
    Iterable<List<Long>> partitionedIds = Iterables.partition(veryLongCollection, BATCH_SIZE);
    List<Item> items = new ArrayList<>(veryLongCollection.size());
    for (var iterator = partitionedIds.iterator(); iterator.hasNext(); ) {
        List<Long> idsBatch = iterator.next();
        List<Item> itemsChunk = processIdsAndGetItems(idsBatch);
        items.addAll(itemsChunk);
    }
    return items;
}
```

Теперь запрос:

- Логируется
- Разбивается на запросы меньшего размера
- Кэш работает эффективнее

*hibernate.query.in\_clause\_parameter\_padding*: **OFF**

SELECT \* FROM items WHERE id IN (param0, param1, param2, param3, param4, param5, param6, param7)

SELECT \* FROM items WHERE id IN (param0, param1, param2, param3, param4, param5, param6)

SELECT \* FROM items WHERE id IN (param0, param1, param2, param3, param4, param5)

SELECT \* FROM items WHERE id IN (param0, param1, param2, param3)

*hibernate.query.in\_clause\_parameter\_padding*: **ON**

SELECT \* FROM items WHERE id IN (param0, param1, param2, param3, param4, param5, param6, param7)

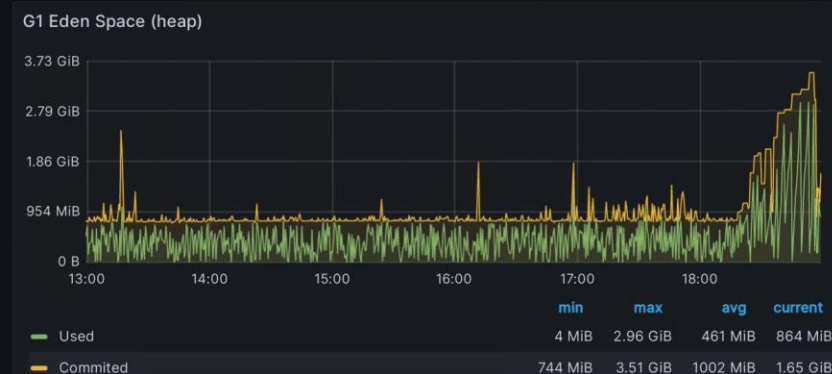
SELECT \* FROM items WHERE id IN (param0, param1, param2, param3)

# Happy(?) end(?)



Конечная точка:  
- потребление памяти снизилось  
- ООМ прекратились  
- "пила" памяти

## JVM Statistics - Memory



# Синдеев Сергей

Java Developer

[Sergei\\_Sindeev@rntgroup.com](mailto:Sergei_Sindeev@rntgroup.com)