федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

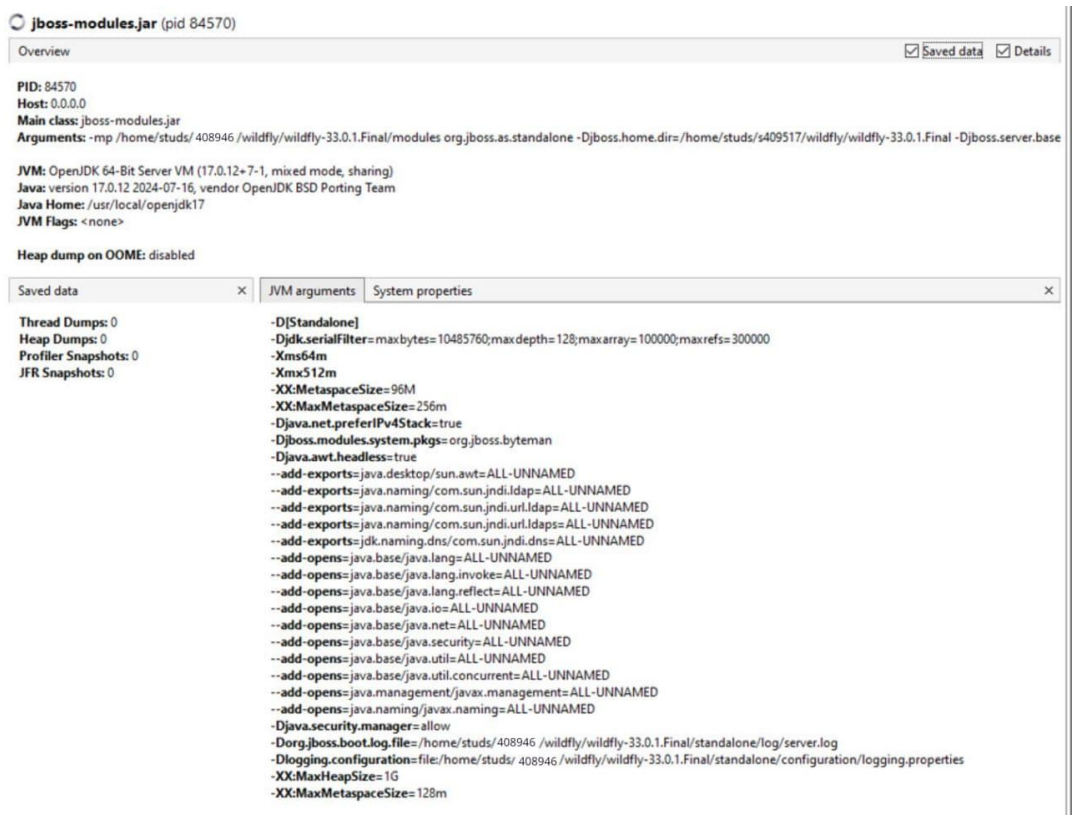
# Лабораторная работа №4

Вариант 815

Студент:
Кулагин Вячеслав, P3209
Преподаватель:
Карташев Владимир Сергеевич

Санкт-Петербург
2025

**Оглавление**



## Задание

1. Для своей программы из лабораторной работы #3 по дисциплине "Веб-программирование" реализовать:
MBean, считающий общее число установленных пользователем точек, а также число точек, попадающих в область. В случае, если пользователь совершил 4 "промаха" подряд, разработанный MBean должен отправлять оповещение об этом событии.
MBean, определяющий процентное отношение "промахов" к общему числу кликов пользователя по координатной плоскости.
2. С помощью утилиты JConsole провести мониторинг программы:
Снять показания MBean-классов, разработанных в ходе выполнения задания 1.
Определить имена всех потоков, выполняющихся при запуске программы.
3. С помощью утилиты VisualVM провести мониторинг и профилирование программы:
Снять график изменения показаний MBean-классов, разработанных в ходе выполнения задания 1, с течением времени.
Определить имя класса, объекты которого занимают наибольший объём памяти JVM; определить пользовательский класс, в экземплярах которого находятся эти объекты.
4. С помощью утилиты VisualVM и профилировщика IDE NetBeans, Eclipse или Idea локализовать и устранить проблемы с производительностью в программе. По результатам локализации и устранения проблемы необходимо составить отчёт, в котором должна содержаться следующая информация:
Описание выявленной проблемы.
Описание путей устранения выявленной проблемы.
Подробное (со скриншотами) описание алгоритма действий, который позволил выявить и локализовать проблему.
Студент должен обеспечить возможность воспроизведения процесса поиска и локализации проблемы по требованию преподавателя.

# Часть 1

Web.lab.jmx.MBeanRegistrar

```
package web.lab.jmx;

import jakarta.annotation.PostConstruct;
import jakarta.enterprise.context.ApplicationScoped;
import jakarta.inject.Inject;
import javax.management.*;
import java.lang.management.ManagementFactory;
import java.util.logging.Level;
import java.util.logging.Logger;

@ApplicationScoped
public class MBeanRegistrar {
    private static final Logger logger = Logger.getLogger(MBeanRegis-
trar.class.getName());

    @Inject private PointsStats pointsStats;
    @Inject private MissRateStats missRateStats;

    @PostConstruct
    public void register() {
        try {
            MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();
            mbs.registerMBean(pointsStats, new Object-
Name("web31:type=PointStats"));
            mbs.registerMBean(missRateStats, new ObjectName("web31:type=Miss-
Ratio"));
```

```
        } catch (Exception e) {
            logger.log(Level.SEVERE, "Failed to register MBeans", e);
        }
    }

    private void registerMBean(MBeanServer server, Object mbean, String object-
Name) {
        try {
            ObjectName name = new ObjectName(objectName);
            if (!server.isRegistered(name)) {
                server.registerMBean(mbean, name);
                logger.info(() -> "Successfully registered MBean: " + object-
Name);
            } else {
                logger.warning(() -> "MBean already registered: " + objectName);
            }
        } catch (InstanceAlreadyExistsException e) {
            logger.warning(() -> "MBean already exists: " + objectName);
        } catch (MalformedObjectNameException e) {
            logger.severe(() -> "Invalid object name format: " + objectName);
        } catch (Exception e) {
            logger.log(Level.SEVERE, "Error registering MBean: " + objectName,
e);
        }
    }
}
```

Web.lab.jmx.MissRateStats

```
package web.lab.jmx;

import jakarta.enterprise.context.ApplicationScoped;
import java.util.concurrent.atomic.AtomicInteger;

@ApplicationScoped
public class MissRateStats implements MissRateStatsMBean {
    private final AtomicInteger totalClicks = new AtomicInteger();
    private final AtomicInteger missCount = new AtomicInteger();

    public void recordClick(boolean hit) {
        totalClicks.incrementAndGet();
        if (!hit) {
            missCount.incrementAndGet();
        }
    }

    @Override
    public int getTotalClicks() {
        return totalClicks.get();
    }

    @Override
    public int getMissCount() {
        return missCount.get();
    }

    @Override
    public double getMissPercentage() {
        int attempts = totalClicks.get();
        return attempts == 0 ? 0.0 : (100.0 * missCount.get()) / attempts;
    }
}
```

Web.lab.jmx.MissRateStatsMBean

```
package web.lab.jmx;

public interface MissRateStatsMBean {
    int getTotalClicks();
    int getMissCount();
    double getMissPercentage();
}
```

Web.lab.jmx.PointStats

```
package web.lab.jmx;

import jakarta.enterprise.context.ApplicationScoped;
import javax.management.*;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.concurrent.atomic.AtomicLong;

@ApplicationScoped
public class PointsStats extends NotificationBroadcasterSupport implements
PointsStatsMBean {
    private final AtomicInteger totalClicks = new AtomicInteger();
    private final AtomicInteger hitsCount = new AtomicInteger();
    private final AtomicInteger consecutiveMisses = new AtomicInteger();
    private final AtomicLong sequenceNumber = new AtomicLong(1);

    public void recordClick(boolean hit) {
        totalClicks.incrementAndGet();

        if (hit) {
            hitsCount.incrementAndGet();
            consecutiveMisses.set(0);
        } else {
            int currentMisses = consecutiveMisses.incrementAndGet();
            if (currentMisses >= 4) {
                try {
                    Notification notification = new Notification(
                            "web.lab.jmx.PointsStats.misses",
                            this,
                            sequenceNumber.getAndIncrement(),
                            System.currentTimeMillis(),
                            "4 промаха подряд"
                    );
                    sendNotification(notification);
                } finally {
                    consecutiveMisses.set(0);
                }
            }
        }
    }

    @Override
    public MBeanNotificationInfo[] getNotificationInfo() {
        return new MBeanNotificationInfo[] {
                new MBeanNotificationInfo(
                        new String[]{"web.lab.jmx.PointsStats.misses"},
                        Notification.class.getName(),
                        "4 consecutive misses notification"
                )
        };
```

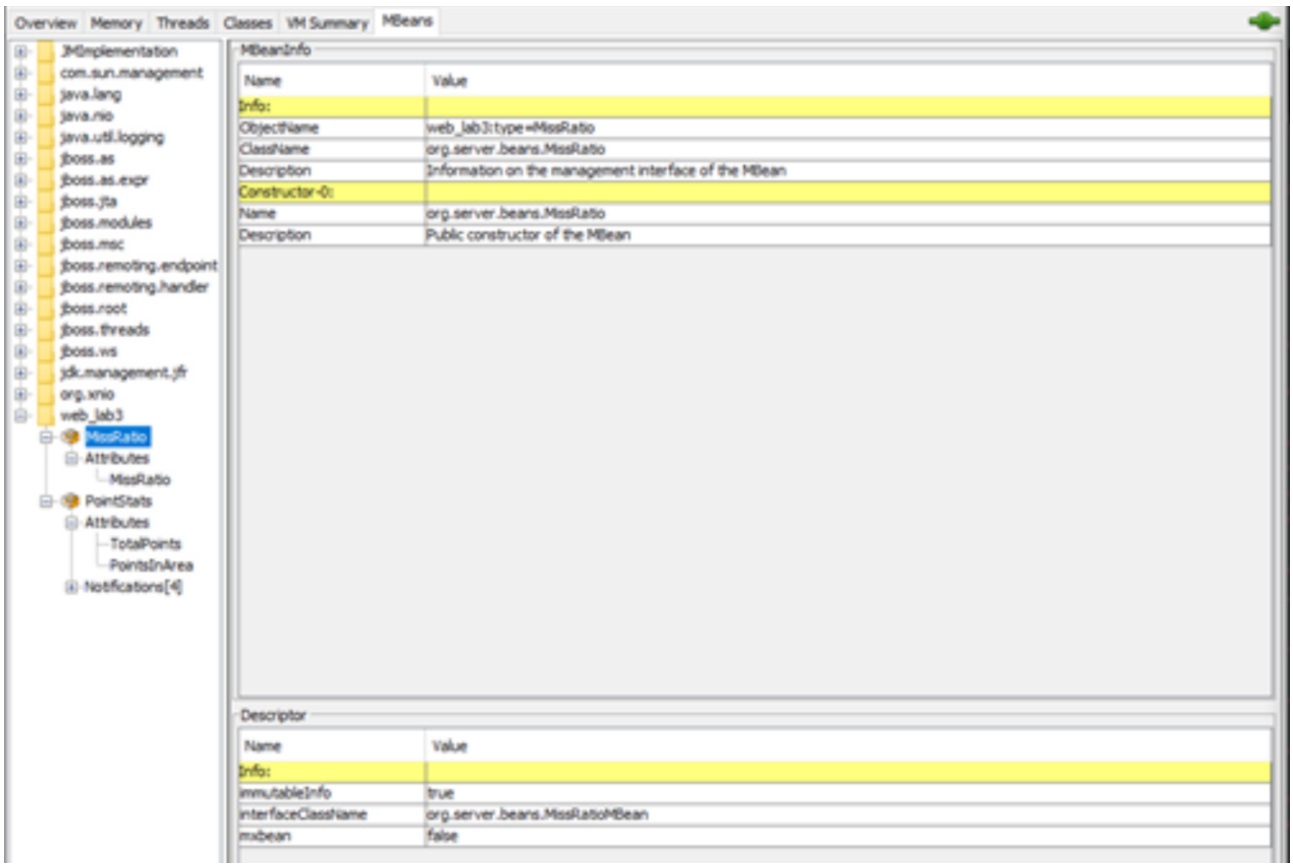
```
    }

    @Override
    public int getTotalClicks() {
        return totalClicks.get();
    }

    @Override
    public int getHitsCount() {
        return hitsCount.get();
    }
}
```

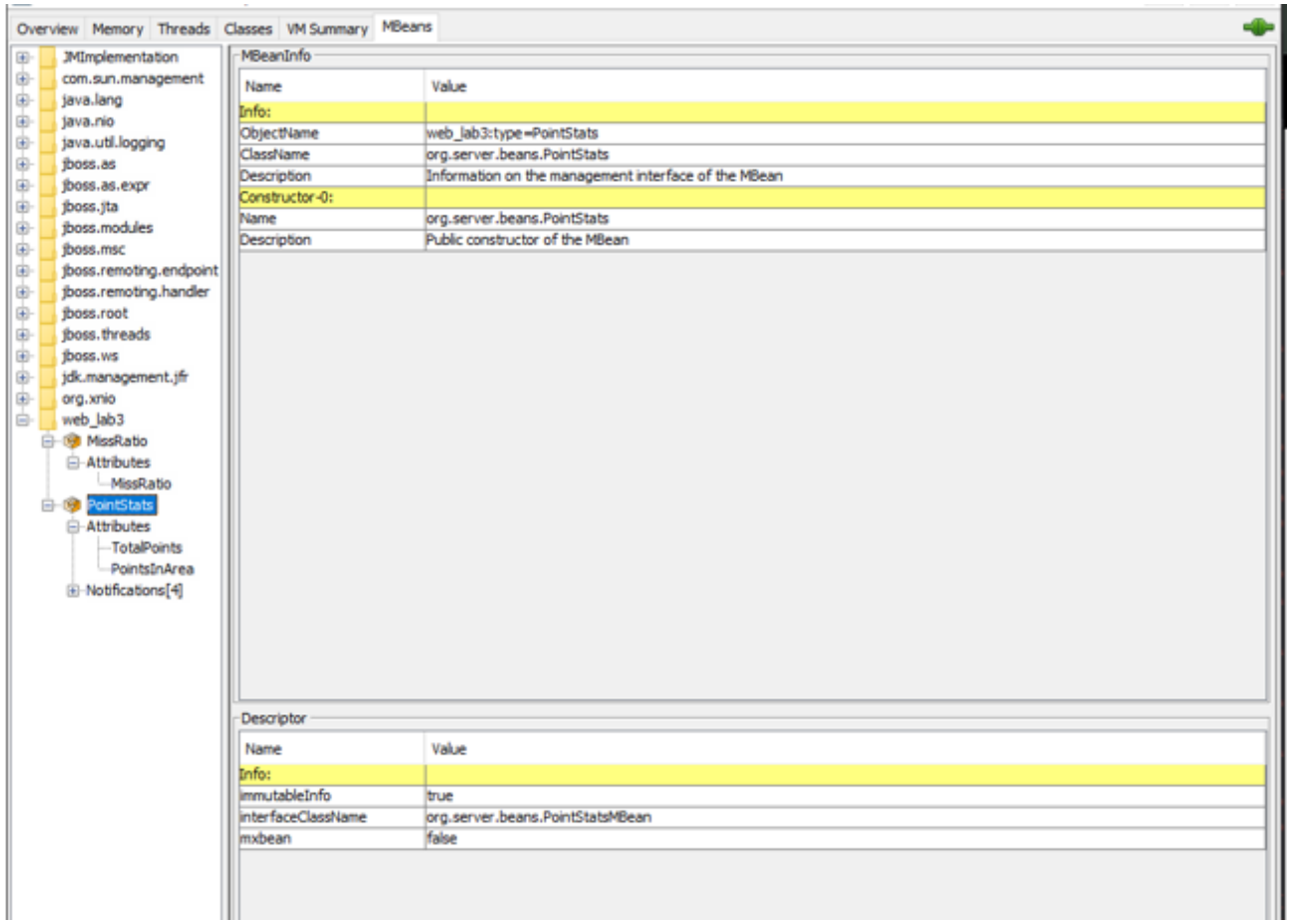
Web.lab.jmx.PointsStatsMBean

```
package web.lab.jmx;

public interface PointsStatsMBean {
    int getTotalClicks();
    int getHitsCount();
}
```

```
Сохранение значений в ResultBean:


package web.lab.web31;

import com.google.gson.Gson;
import jakarta.annotation.PostConstruct;
import jakarta.enterprise.context.ApplicationScoped;
import jakarta.inject.Inject;
import jakarta.inject.Named;
import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;
import jakarta.persistence.TypedQuery;
import jakarta.transaction.Transactional;
import lombok.Getter;
import lombok.Setter;
import web.lab.entity.ResultEntity;
import web.lab.entity.Point;
import web.lab.jmx.MBeanRegistrar;
import web.lab.jmx.MissRateStats;
import web.lab.jmx.PointsStats;

import java.io.Serializable;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;

@Named
@ApplicationScoped
@Getter
@Setter
public class ResultsBean implements Serializable {

    @Inject
    private MBeanRegistrar mbeanRegistrar;

    @PersistenceContext(unitName = "my-persistence-unit")
    private EntityManager entityManager;

    private List<Point> results = new ArrayList<>();
```

```
    private boolean isInitialized = false;

    @PostConstruct
    private void init() {
        if (!isInitialized) {
            loadResults();
            isInitialized = true;
        }
    }

    private void loadResults() {
        TypedQuery<ResultEntity> query = entityManager.createQuery(
                "SELECT r FROM ResultEntity r", ResultEntity.class);

        List<ResultEntity> resultEntities = query.getResultList();
        results = new ArrayList<>();
        for (ResultEntity entity : resultEntities) {
            results.add(new Point(entity.getX(), entity.getY(), entity.getR(),
entity.isHit()));
        }
    }

    @Inject
    private PointsStats stats;

    @Inject
    private MissRateStats missRateStats;

    @Transactional
    public void addResult(Point point) {
        ResultEntity resultEntity = ResultEntity.builder()
                .x(point.getX())
                .y(point.getY())
                .r(point.getR())
                .hit(point.isHit())
                .timestamp(LocalDateTime.now())
                .build();

        boolean hit = point.isHit();
        stats.recordClick(hit);
        missRateStats.recordClick(hit);

        entityManager.persist(resultEntity);
        results.add(new Point(resultEntity.getX(), resultEntity.getY(), re-
sultEntity.getR(), resultEntity.isHit()));
    }

    public String getJsonResults() {
        return new Gson().toJson(results);
    }
}
```


Overview
Memory
Threads
Classes
VM Summary
MBeans
JMImplementation
com.sun.management
java.lang
java.nio
java.util.logging
jboss.as
jboss.as.expr
jboss.jta
jboss.modules
jboss.msc
jboss.remoting.endpoint
jboss.remoting.handler
jboss.root
jboss.threads
jboss.ws
jdk.management.jfr
org.xnio
web_lab3
MissRatio
Attributes
MissRatio
PointStats
Attributes
TotalPoints
PointsInArea
Notifications[4]
MBeanInfo
Name
Value
Info:
ObjectName
web_lab3:type=MissRatio
ClassName
org.server.beans.MissRatio
Description
Information on the management interface of the MBean
Constructor-0:
Name
org.server.beans.MissRatio
Description
Public constructor of the MBean
Descriptor
Name
Value
Info:
immutableInfo
true
interfaceClassName
org.server.beans.MissRatioMBean
mxbean
false


Overview
Memory
Threads
Classes
VM Summary
MBeans
JMImplementation
com.sun.management
java.lang
java.nio
java.util.logging
jboss.as
jboss.as.expr
jboss.jta
jboss.modules
jboss.msc
jboss.remoting.endpoint
jboss.remoting.handler
jboss.root
jboss.threads
jboss.ws
jdk.management.jfr
org.xnio
web_lab3
MissRatio
Attributes
MissRatio
PointStats
Attributes
TotalPoints
PointsInArea
Notifications[4]
MBeanInfo
Name
Value
Info:
ObjectName
web_lab3:type=PointStats
ClassName
org.server.beans.PointStats
Description
Information on the management interface of the MBean
Constructor-0:
Name
org.server.beans.PointStats
Description
Public constructor of the MBean
Descriptor
Name
Value
Info:
immutableInfo
true
interfaceClassName
org.server.beans.PointStatsMBean
mxbean
false

Attribute values

| Name | Value |
|---|---|
| HitPoints | 22 |
| MissPercentage | 45.0 |
| TotalPoints | 40 |

Attribute values

| Name | Value |
|---|---|
| HitPoints | 22 |
| TotalPoints | 40 |

| | | | |
|---|---|---|---|
| Connection name: | pid: 48591 jboss-modules.jar -mp /home/felijy/opi-lab4/wildfly-preview-26.1.3.Final/modules org.jboss.as.standalone -Djboss.home.dir=/home/felijy/opi-lab4/wildfly-preview-26.1.3.Final -Djboss.server.base.dir=/home/felijy/opi-lab4/wildfly-preview-26.1.3.Final/standalone -b 0.0.0.0 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.port=28002 -Dcom.sun.management.jmxremote.rmi.port=28002 -Djava.rmi.server.hostname=0.0.0.0 -Djboss.bind.address.management=0.0.0.0 -Djboss.bind.address=0.0.0.0 | | |
| Virtual Machine: | OpenJDK 64-Bit Server VM version 21.0.7+6-Ubuntu-0ubuntu122.04 | Uptime: | 10 minutes |
| Vendor: | Ubuntu | Process CPU time: | 42.540 seconds |
| Name: | 48591@DESKTOP-4IOU9J5 | JIT compiler: | HotSpot 64-Bit Tiered Compilers |
| | | Total compile time: | 26.412 seconds |

| | | | |
|---|---|---|---|
| Live threads: | 85 | Current classes loaded: | 23,106 |
| Peak: | 155 | Total classes loaded: | 23,117 |
| Daemon threads: | 29 | Total classes unloaded: | 11 |
| Total threads started: | 166 | | |

| | | | |
|---|---|---|---|
| Current heap size: | 186,197 kbytes | Committed memory: | 229,376 kbytes |
| Maximum heap size: | 524,288 kbytes | Pending finalization: | 0 objects |
| Garbage collector: | Name = 'G1 Young Generation', Collections = 52, Total time spent = 0.184 seconds | | |
| Garbage collector: | Name = 'G1 Concurrent GC', Collections = 16, Total time spent = 0.035 seconds | | |
| Garbage collector: | Name = 'G1 Old Generation', Collections = 0, Total time spent = 0.000 seconds | | |

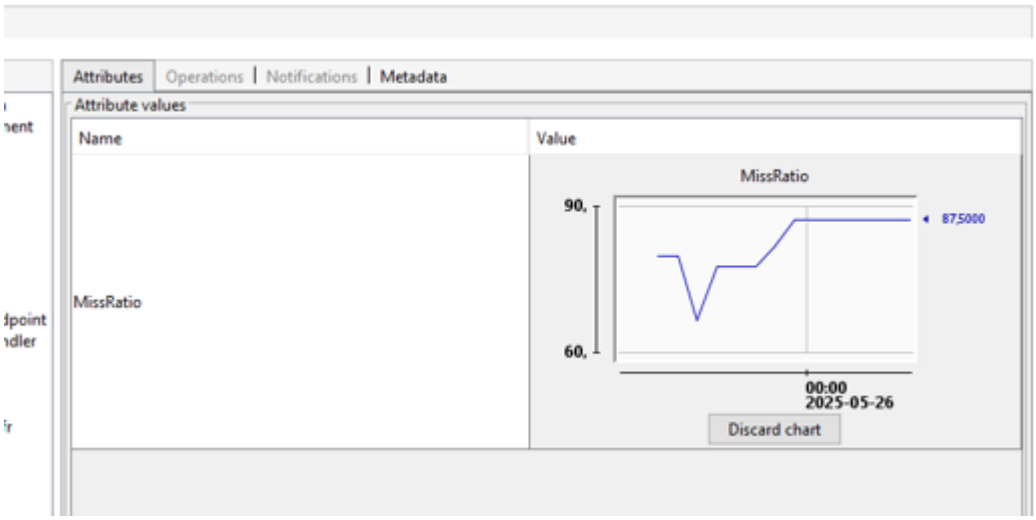
| | | | |
|---|---|---|---|
| Operating System: | Linux 6.6.87.1-microsoft-standard-WSL2 | Total physical memory: | 7,704,784 kbytes |
| Architecture: | amd64 | Free physical memory: | 5,188,368 kbytes |
| Number of processors: | 16 | Total swap space: | 2,097,152 kbytes |
| Committed virtual memory: | 1,715,580 kbytes | Free swap space: | 2,097,152 kbytes |

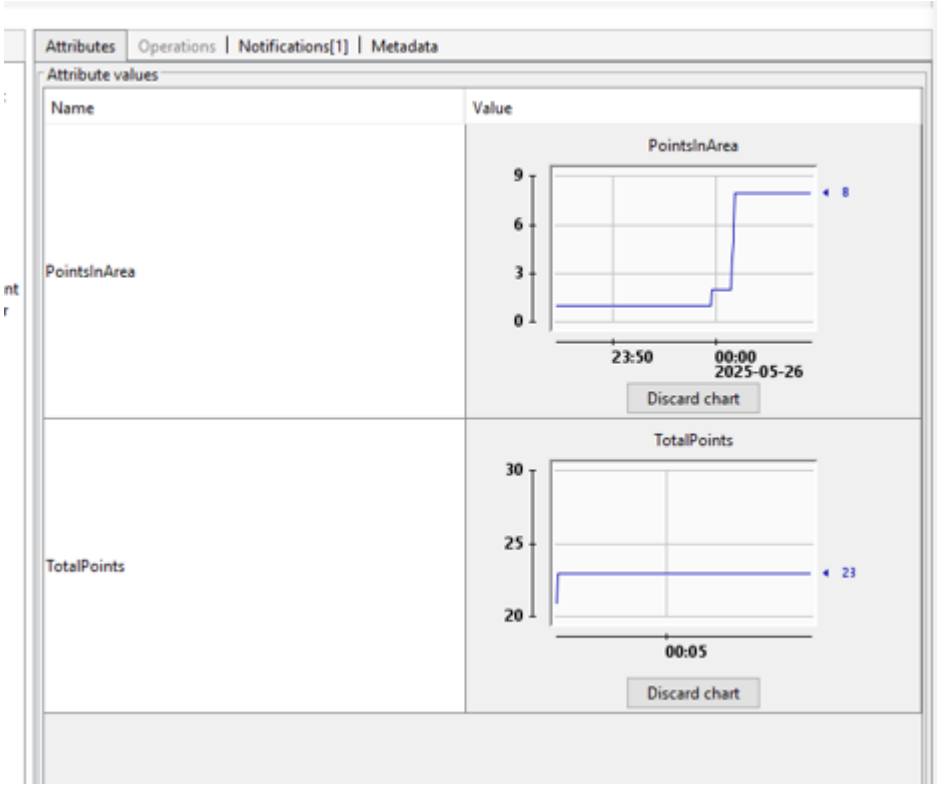
| | |
|---|---|
| VM arguments: | -D[Standalone] -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true --add-exports=java.desktop/sun.awt=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.url.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.url.ldaps=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.security=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.management/javax.management=ALL-UNNAMED --add-opens=java.naming/javax.naming=ALL-UNNAMED -Djava.security.manager=allow -Dorg.jboss.boot.log.file=/home/felijy/opi-lab4/wildfly-preview-26.1.3.Final/standalone/log/server.log -Dlogging.configuration=file:/home/felijy/opi-lab4/wildfly-preview-26.1.3.Final/standalone/configuration/logging.properties |
| Class path: | /home/felijy/opi-lab4/wildfly-preview-26.1.3.Final/jboss-modules.jar |
| Library path: | /usr/java/packages/lib:/usr/lib/x86_64-linux-gnu/jni:/lib/x86_64-linux-gnu:/usr/lib/x86_64-linux-gnu:/usr/lib/jni:/lib:/usr/lib |
| Boot class path: | Unavailable |

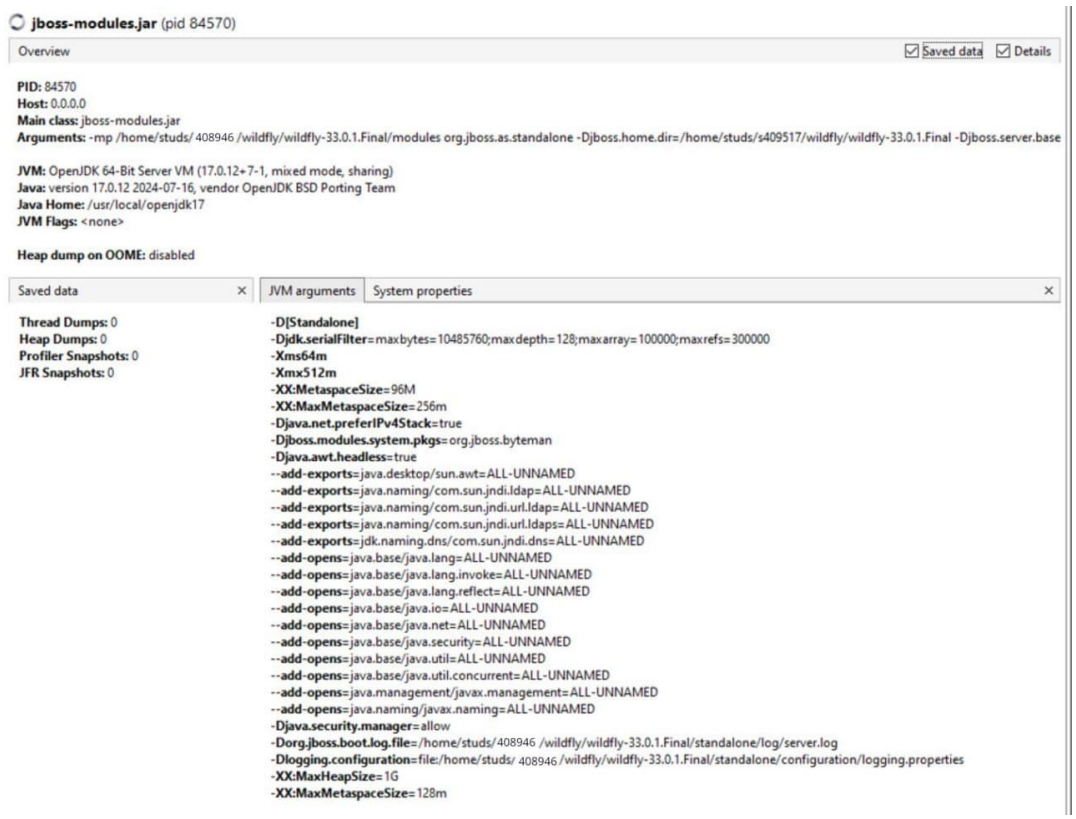
Notification buffer

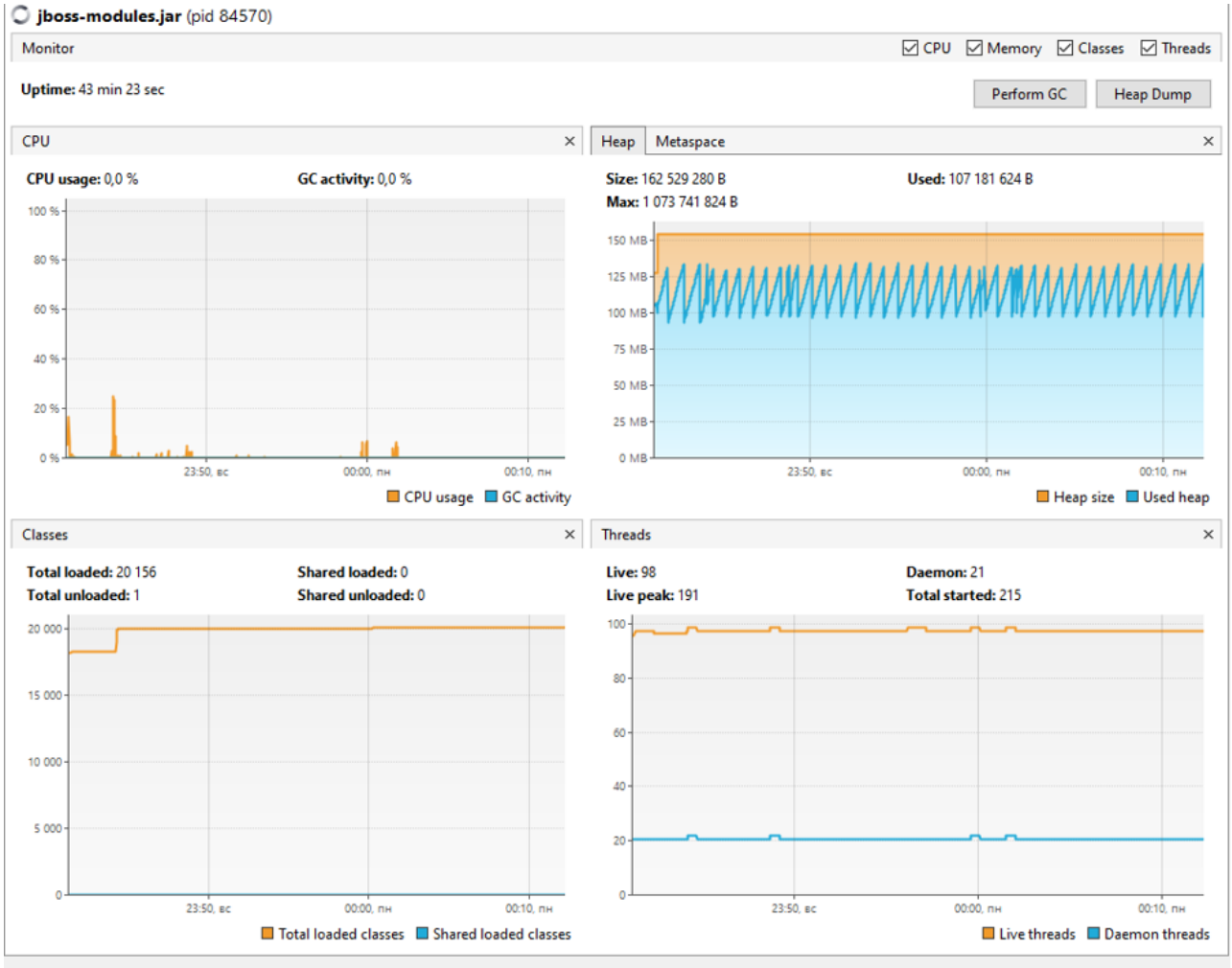
| TimeStamp | Type | UserData | SeqNum | Message | Event | Source |
|---|---|---|---|---|---|---|
| 21:02:05:078 | Point.OutOfBounds | | 11 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:02:03:546 | Point.OutOfBounds | | 10 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:02:02:188 | Point.OutOfBounds | | 9 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:02:00:642 | Point.OutOfBounds | | 8 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:59:602 | Point.OutOfBounds | | 7 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:58:428 | Point.OutOfBounds | | 6 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:57:319 | Point.OutOfBounds | | 5 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:56:337 | Point.OutOfBounds | | 4 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:55:419 | Point.OutOfBounds | | 3 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:53:960 | Point.OutOfBounds | | 2 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |
| 21:01:51:320 | Point.OutOfBounds | | 1 | Point is out of bounds | javax.management... | web_lab3:type=PointStats |

Графики

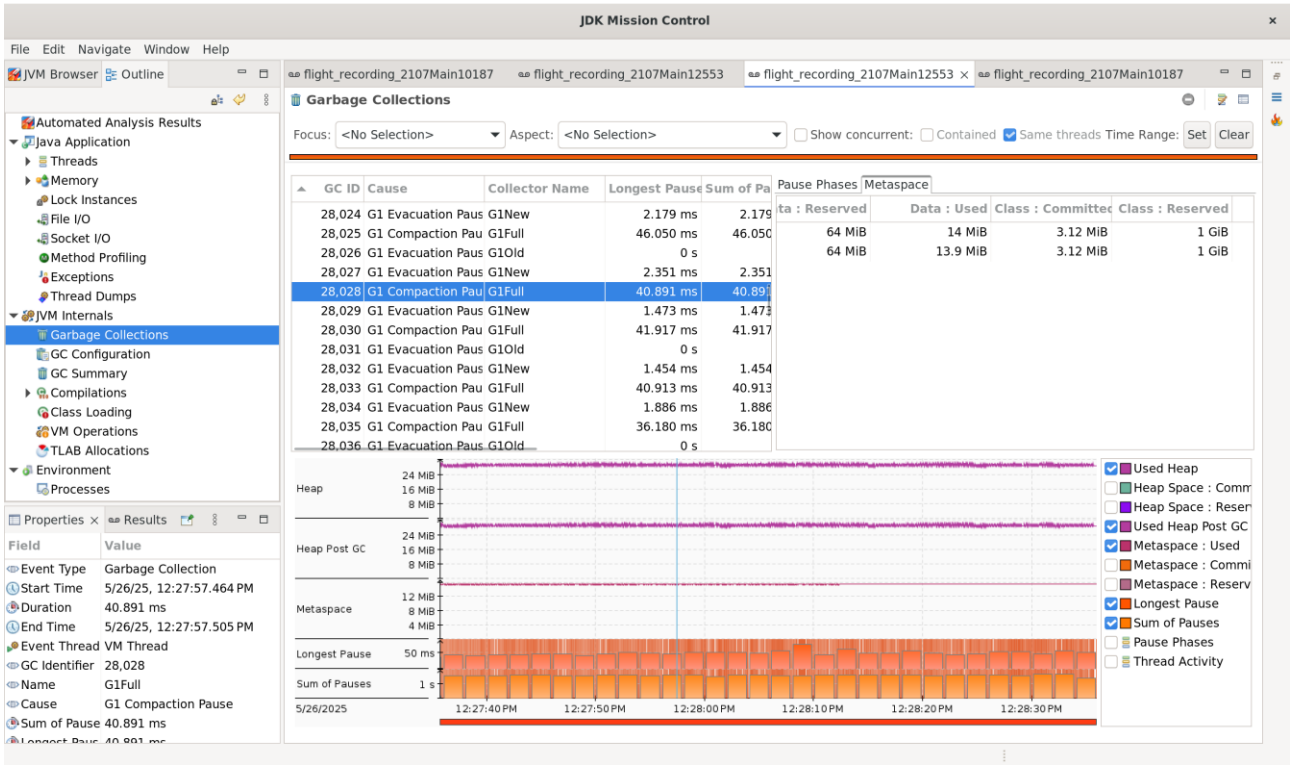

Attributes
Operations | Notifications | Metadata
Attribute values
Name
Value
MissRatio
MissRatio
90,
60,
87,5000
00:00
2025-05-26
Discard chart

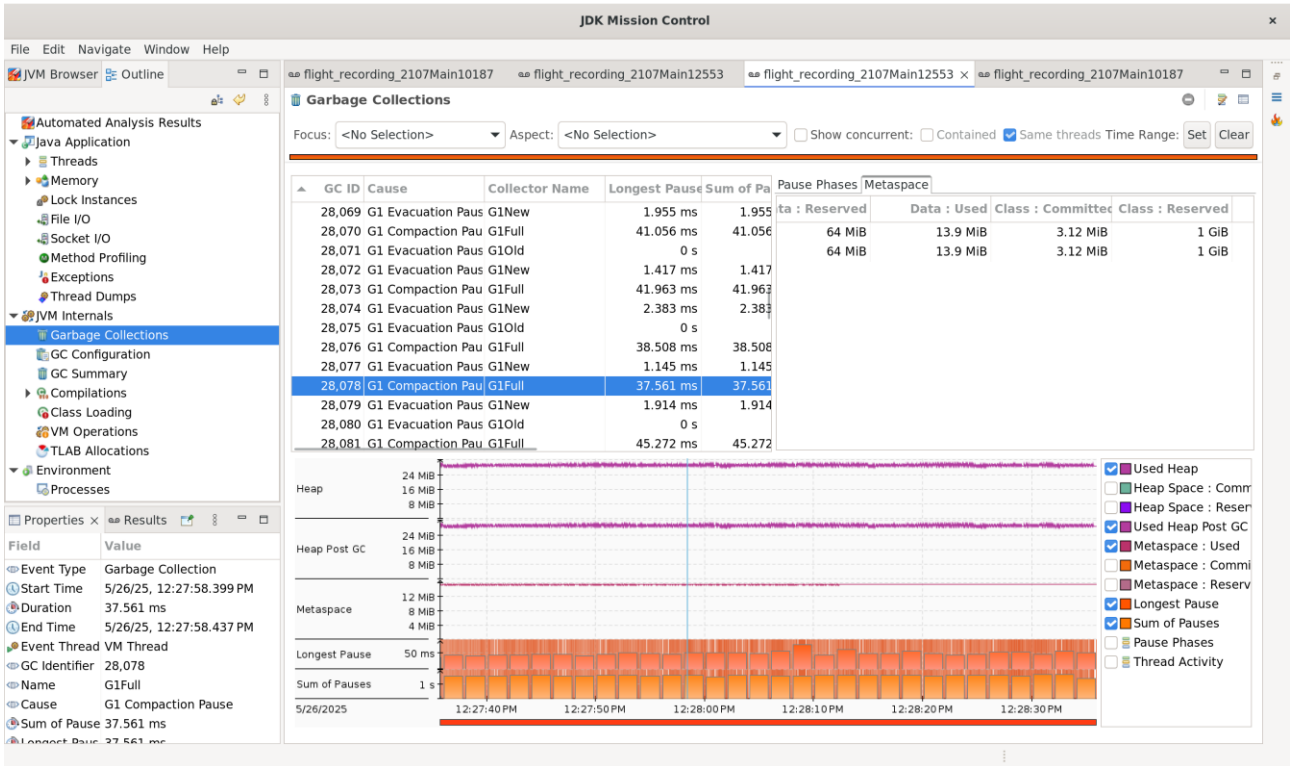

Attributes
Operations | Notifications[1] | Metadata
Attribute values
Name
Value
PointsInArea
PointsInArea
9
6
3
0
8
23:50
00:00
2025-05-26
Discard chart
TotalPoints
TotalPoints
30
25
20
23
00:05
Discard chart


jboss-modules.jar (pid 84570)
Overview
Saved data
Details
PID: 84570
Host: 0.0.0.0
Main class: jboss-modules.jar
Arguments: -mp /home/studs/ 408946 /wildfly/wildfly-33.0.1.Final/modules org.jboss.as.standalone -Djboss.home.dir=/home/studs/s409517/wildfly/wildfly-33.0.1.Final -Djboss.server.base
JVM: OpenJDK 64-Bit Server VM (17.0.12+7-1, mixed mode, sharing)
Java: version 17.0.12 2024-07-16, vendor OpenJDK BSD Porting Team
Java Home: /usr/local/openjdk17
JVM Flags: <none>
Heap dump on OOME: disabled
Saved data
JVM arguments
System properties
Thread Dumps: 0
Heap Dumps: 0
Profiler Snapshots: 0
JFR Snapshots: 0
-D[Standalone]
-Djdk.serialFilter=maxbytes=10485760;maxdepth=128;maxarray=100000;maxrefs=300000
-Xms64m
-Xmx512m
-XX:MetaspaceSize=96M
-XX:MaxMetaspaceSize=256m
-Djava.net.preferIPv4Stack=true
-Djboss.modules.system.pkgs=org.jboss.byteman
-Djava.awt.headless=true
--add-exports=java.desktop/sun.awt=ALL-UNNAMED
--add-exports=java.naming/com.sun.jndi.ldap=ALL-UNNAMED
--add-exports=java.naming/com.sun.jndi.url.ldap=ALL-UNNAMED
--add-exports=java.naming/com.sun.jndi.url.ldaps=ALL-UNNAMED
--add-exports=jdk.naming.dns/com.sun.jndi.dns=ALL-UNNAMED
--add-opens=java.base/java.lang=ALL-UNNAMED
--add-opens=java.base/java.lang.invoke=ALL-UNNAMED
--add-opens=java.base/java.lang.reflect=ALL-UNNAMED
--add-opens=java.base/java.io=ALL-UNNAMED
--add-opens=java.base/java.net=ALL-UNNAMED
--add-opens=java.base/java.security=ALL-UNNAMED
--add-opens=java.base/java.util=ALL-UNNAMED
--add-opens=java.base/java.util.concurrent=ALL-UNNAMED
--add-opens=java.management/javax.management=ALL-UNNAMED
--add-opens=java.naming/javax.naming=ALL-UNNAMED
-Djava.security.manager=allow
-Dorg.jboss.boot.log.file=/home/studs/408946 /wildfly/wildfly-33.0.1.Final/standalone/log/server.log
-Dlogging.configuration=file:/home/studs/ 408946 /wildfly/wildfly-33.0.1.Final/standalone/configuration/logging.properties
-XX:MaxHeapSize=1G
-XX:MaxMetaspaceSize=128m

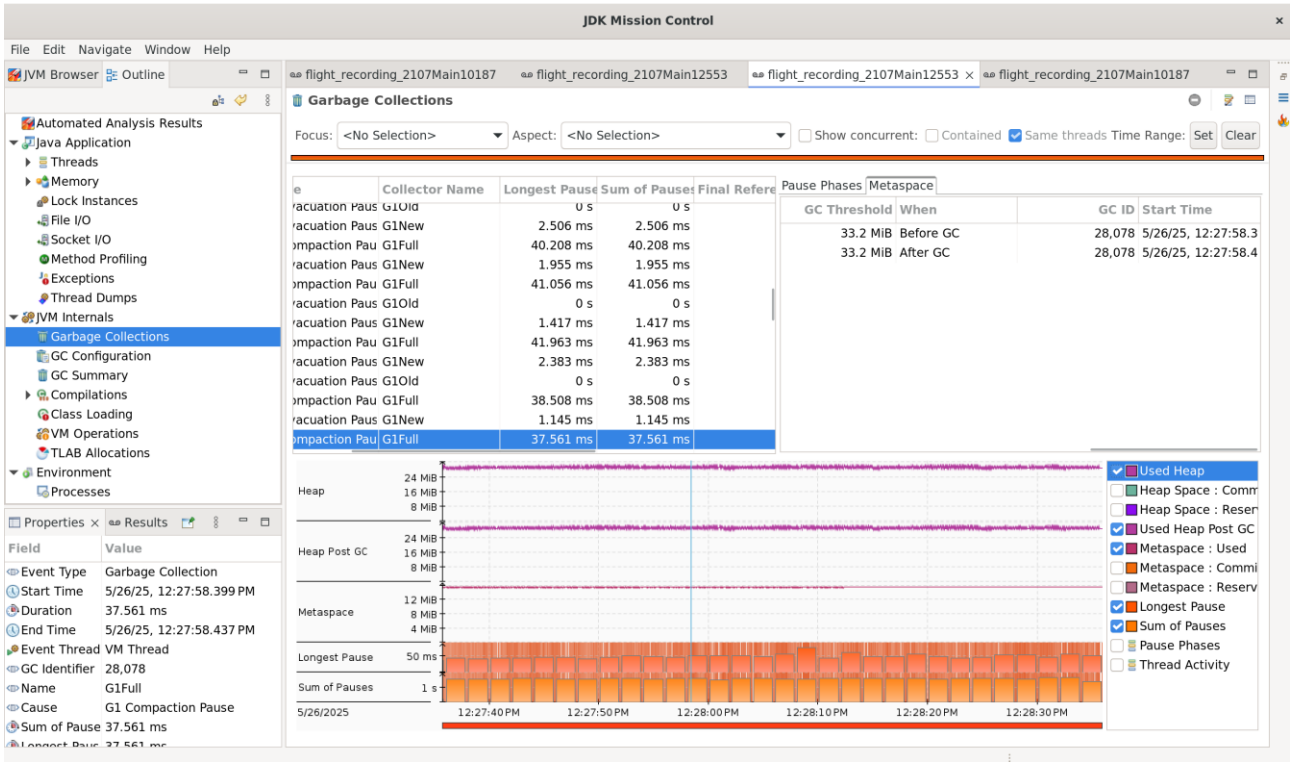

jboss-modules.jar (pid 84570)
Monitor
CPU
Memory
Classes
Threads
Uptime: 43 min 23 sec
Perform GC
Heap Dump
CPU
CPU usage: 0,0 %
GC activity: 0,0 %
100 %
80 %
60 %
40 %
20 %
0 %
23:50, вс
00:00, пн
00:10, пн
CPU usage
GC activity
Heap
Metaspace
Size: 162 529 280 B
Used: 107 181 624 B
Max: 1 073 741 824 B
150 MB
125 MB
100 MB
75 MB
50 MB
25 MB
0 MB
Heap size
Used heap
Classes
Total loaded: 20 156
Shared loaded: 0
Total unloaded: 1
Shared unloaded: 0
20 000
15 000
10 000
5 000
0
Total loaded classes
Shared loaded classes
Threads
Live: 98
Daemon: 21
Live peak: 191
Total started: 215
100
80
60
40
20
0
Live threads
Daemon threads

# Исследование на утечки

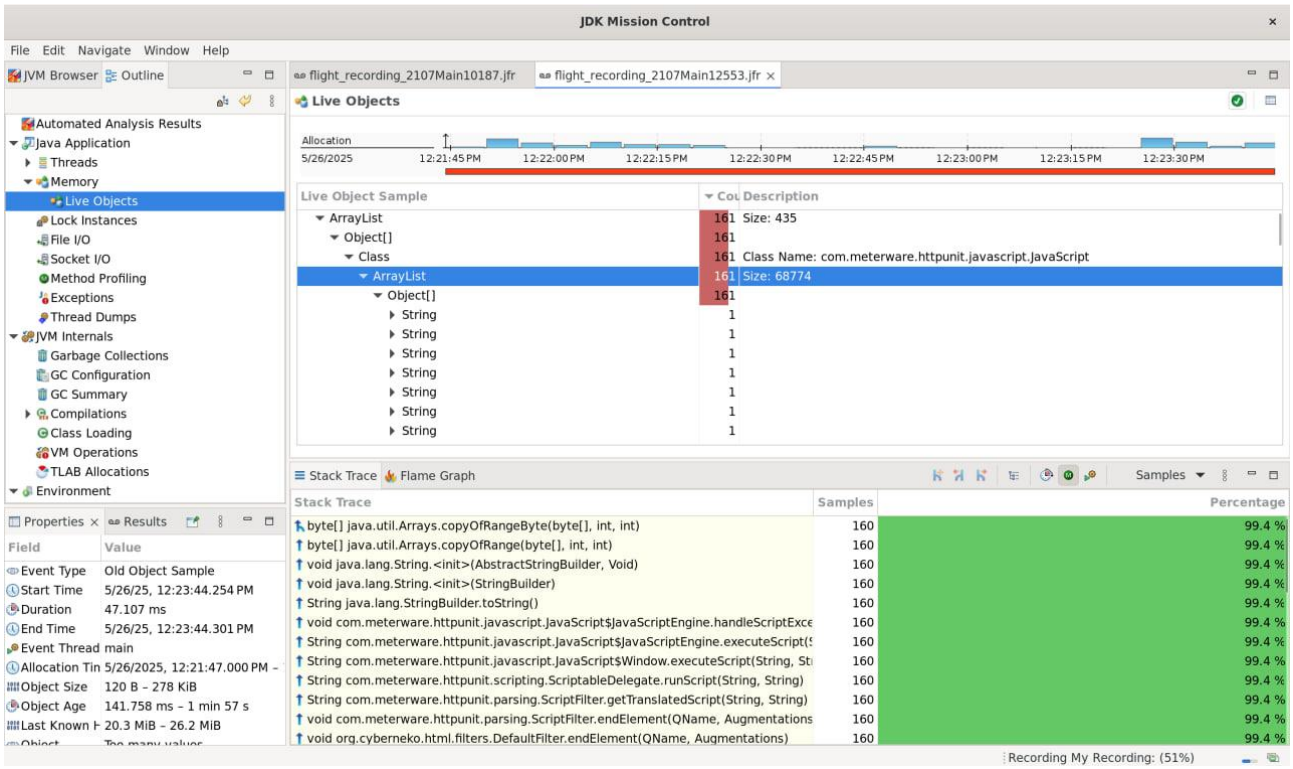


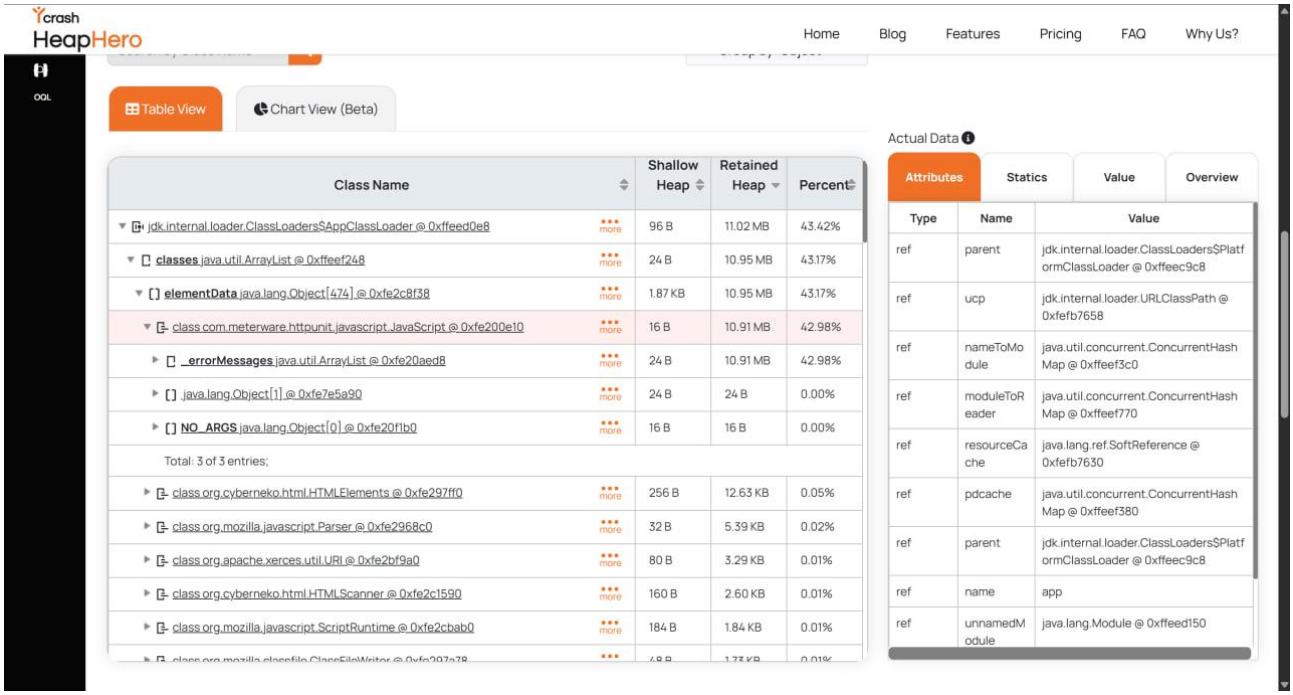

GC работает очень часто, при этом практически не очищая место. То есть он срабатывает, но никак не очищает место в памяти, а иногда даже наоборот оно увеличивается, либо лишь незначительно уменьшается. Это свидетельствует об утечке памяти.




Далее находим что стало ее причиной




Можно также исследовать heap:




Утечка в классе com.meterware.httpunit.JavaScript

## Вывод

В ходе выполнения лабораторной работы я освоил практику создания MBeans в веб-приложениях, изучил утилиты JConsole, VisualVM и JFR для мониторинга и профилирования программ, а также приобрел навыки выявления и устранения утечек памяти на основе полученных данных.