

# No wonders.

Even at scale.

Alexey Ustenko, Pavel Bushmelev  
ATTRAQ, 2018



Alexey  
Ustenko



Pavel  
Bushmelev

Dev. Manager

CloudOps Manager

Creates product

Runs product

...both “having fun” resolving incidents



Who we are

“ATTRAQT provides SaaS solutions that maximize onsite search, online merchandising and personalization for eCommerce retailers.”

[attraqt.com](http://attraqt.com)



You Have Selected

Footwear

Categories

- Men (1262)
- Women (536)
- Kids (889)

Brand

- Nike (732)
- adidas (215)
- adidas Originals (408)
- adidas Skateboarding (18)
- Arena (2)
- ASICS (23)
- Benetton (16)
- Brooks (8)

Size

- 6 (1210)
- 7 (1194)
- 8 (1045)
- 9 (821)
- 10 (875)
- 0 (6)
- 0.5 (1)
- 1 (786)

Exclusive

- JD Exclusive (562)
- Web Exclusive (156)

Colour

- Black (812)
- Blue (322)
- Brown (96)
- Green (162)
- Grey (395)
- Multi (4)
- Orange (33)
- Pink (188)
- Purple (19)
- Red (117)
- White (487)
- Yellow (37)

Price

From £5.00 - To £255.00

Sale

- Only view sale items (794)

### Latest Footwear

MEN'S WOMEN'S KIDS'

Fresh style starts from the feet up. Get your footwear game sorted with JD's on-point collection of creps and kicks. Right now, adidas Originals' NMD and EQT ADV trainers have got street style on lock, while Nike's Radiation Flare football boots change the game every time you hit the pitch. With JD-exclusive footwear from Vans, Jordan and Converse too, the King of Trainers reign supreme when it comes to trainer royalty.

**ALWAYS THE LEADER**

Sort by Recommended 2733 Products Show More



Quick Buy

Nike Air Max 90 Essential OG £100.00

Nike Air £110.00

trainers

Search

Results

- trainers
- trainers junior
- trainers kids
- trainers men
- trainers women

Recently Viewed

- Vans Old Skool £60.00
- adidas Originals Superstar £75.00
- ADIDAS Stan Smith Shoes £75.00
- adidas Originals Stan Smith £75.00



Product Info Review Delivery Returns

A true innovator, these men's Old Skool trainers from Vans introduced the world to the iconic side stripe, the instantly recognisable signature that lets everyone know you're rocking the legendary skate company. A mainstay on the footwear scene, these classic kicks have endured for decades after a decade thanks to their long-lasting on-board performance and effortless comfort. A canvas upper delivers lightweight wear, while a padded ankle collar keeps the foot feeling fresh. With the classic outsole pattern to the underside for supreme grip whether you're on two feet or on four wheels, finished with the iconic Off The Wall tab to the heels.

Care & Material  
Leather & Textile Upper/Synthetic Sole

Colour:  
Black

Product Code: 081773/081773

Recommended for you

- Vans Sk8-Hi
- Vans Classic Slip-On
- Nike Air Force 1 Low

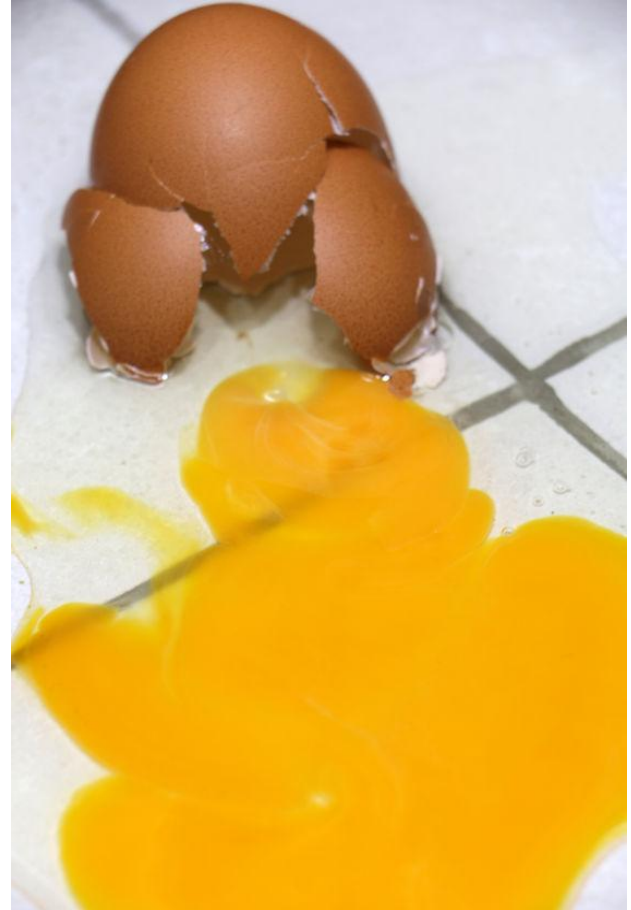
## Our scale

- > 2K servers
- > 99.9% uptime
- > 360M queries last Black Friday
- > 6K peak QPS



Murphy's law

Anything that  
can go wrong  
will go wrong



Life at scale

Something that

*cannot* go wrong

still goes wrong



Life at scale

Something that  
*cannot* go wrong  
still goes wrong



Life at scale

Something that  
*cannot* go wrong  
still goes wrong



## Incident - stages

Rejection

Investigation

Process  
change

Fix

## Incident - stages

Rejection

Investigation

Process  
change

Fix

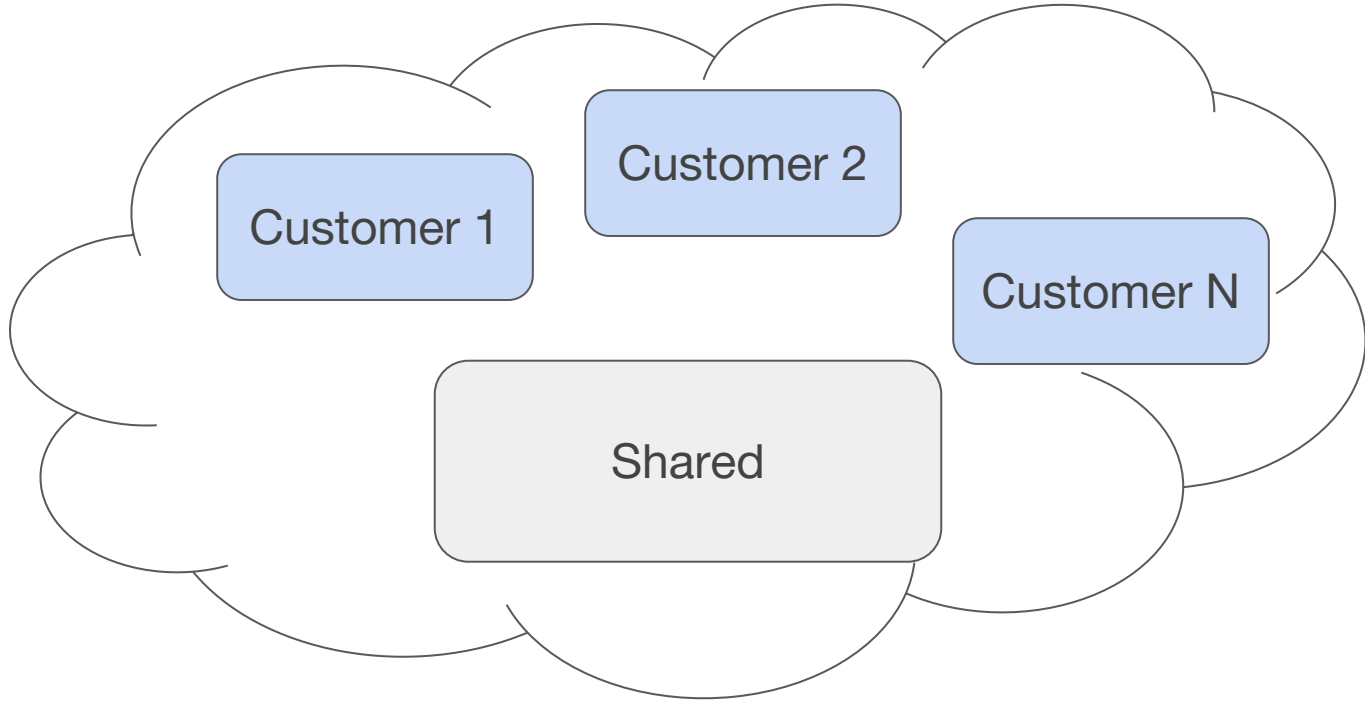


# Surprise menu

# Java 8

**Backend**

# Architecture



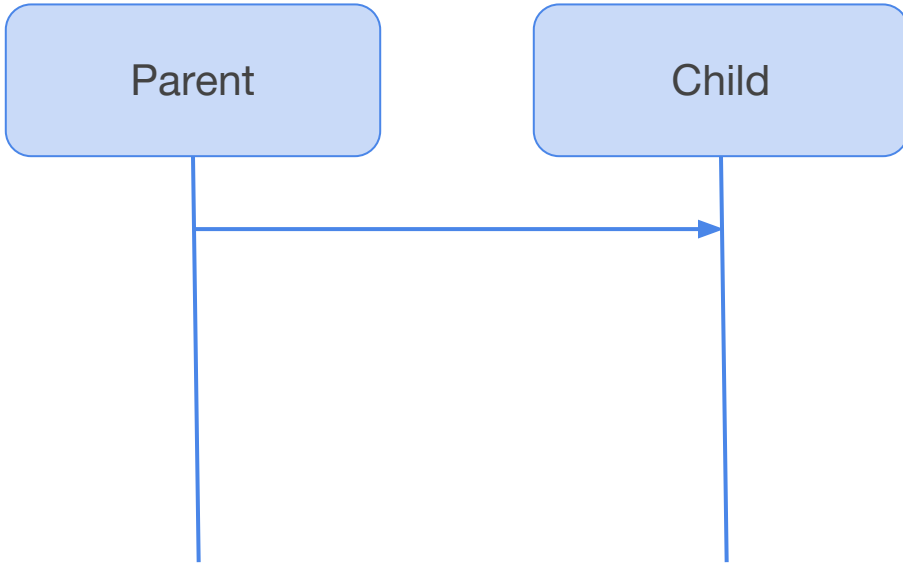




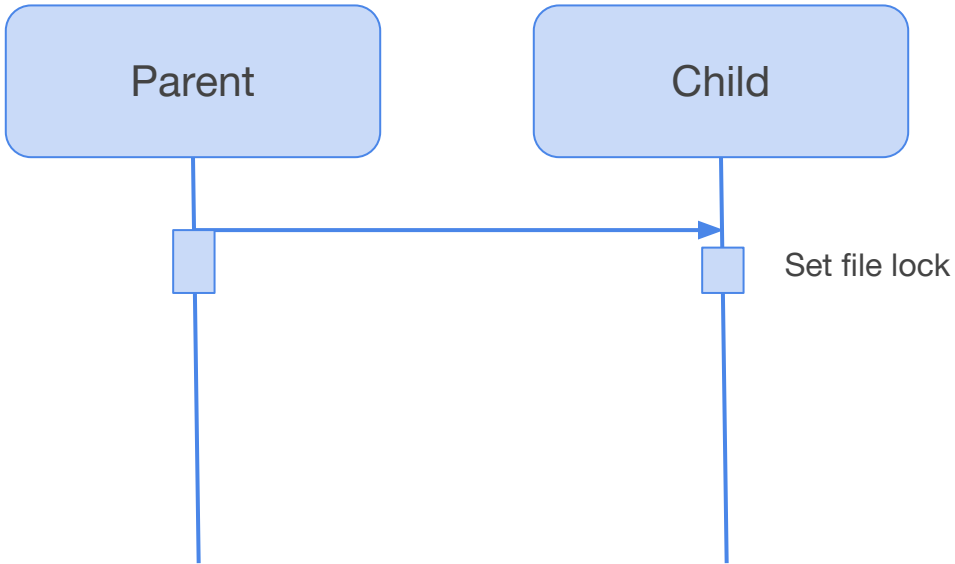
Aperitif

Let's read  
config...

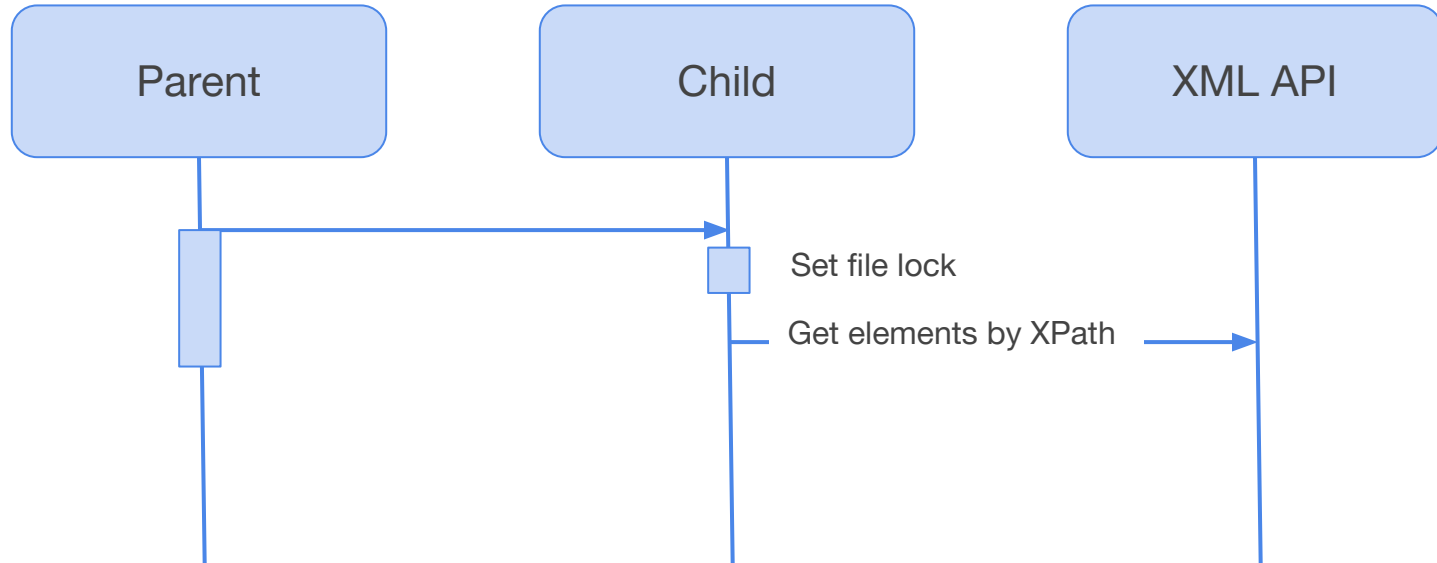
What can possibly go wrong!



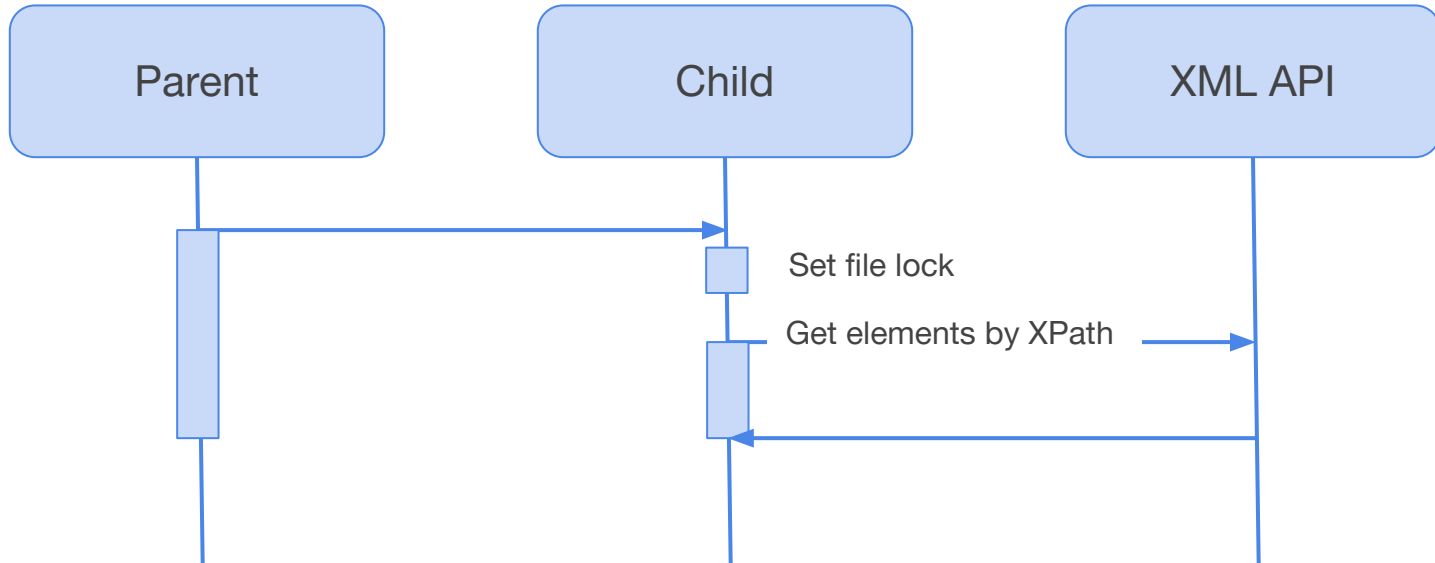
# What can possibly go wrong!



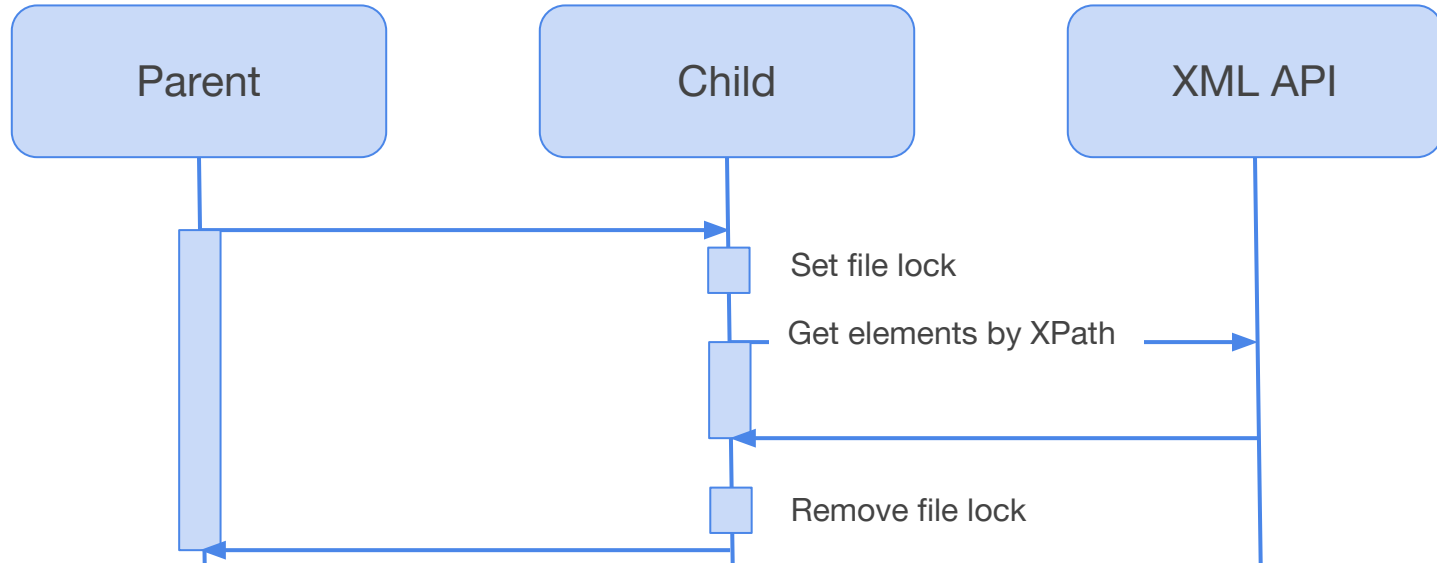
# What can possibly go wrong!



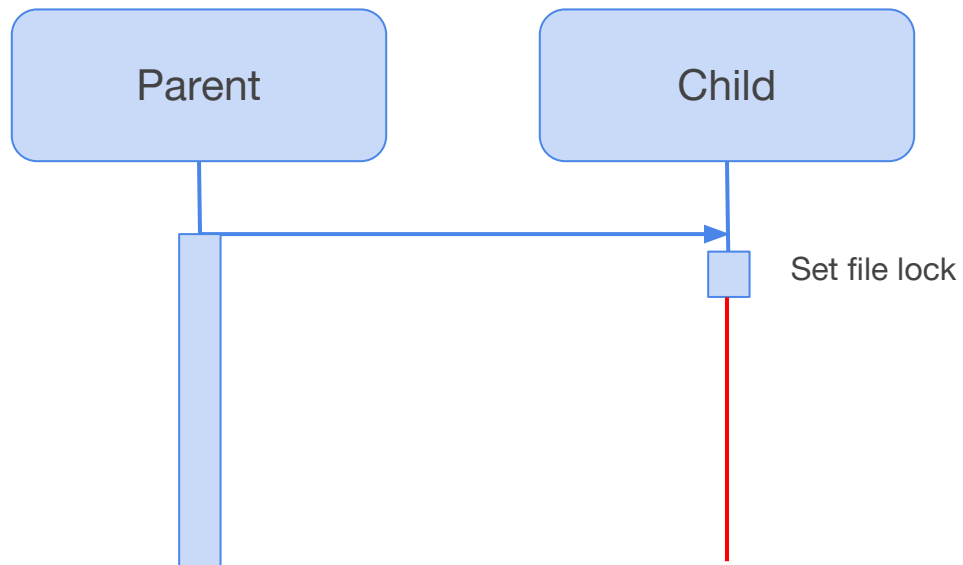
# What can possibly go wrong!



# What can possibly go wrong!



# Evidence





*Can't reproduce!*

Code?

Run child  
concurrently

Corrupt /  
Delete  
XML

Run 5 000  
times





Until it  
happens  
again

```
top # the child process takes 100% CPU  
kill -3 <pid> # a few times
```

```
java.lang.Thread.State: RUNNABLE
```

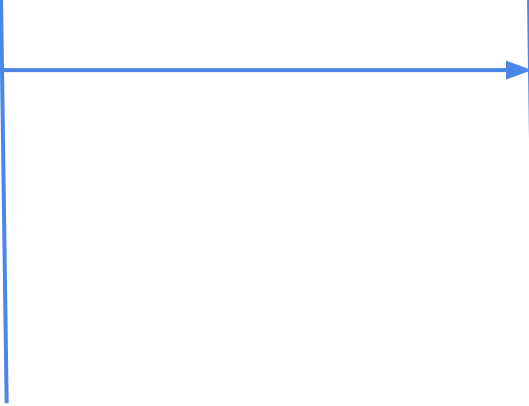
```
at org.apache.xml.dtm.ref.DTMDefaultBase._nextsib(DTMDefaultBase.java:565)
```

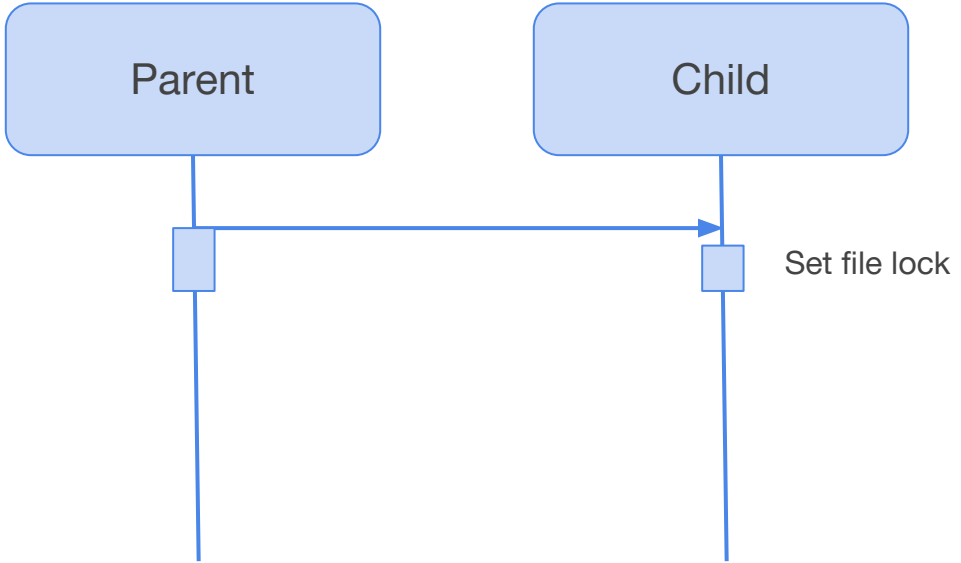
```
at o.a.xml.dtm.ref.DTMDefaultBase.getNextSibling(DTMDefaultBase.java:1142)
```

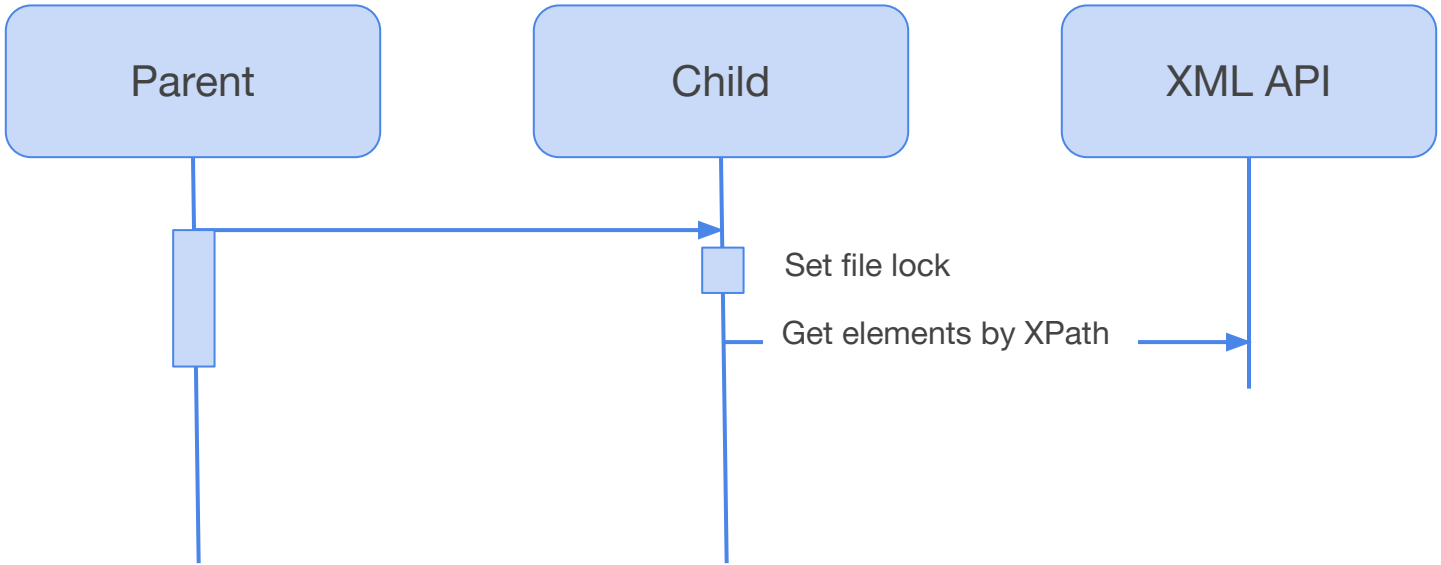
```
at o.a.xml.dtm.ref.DTMDefaultBaseTraversers$...next(DTM...Traversers.java:463)
```

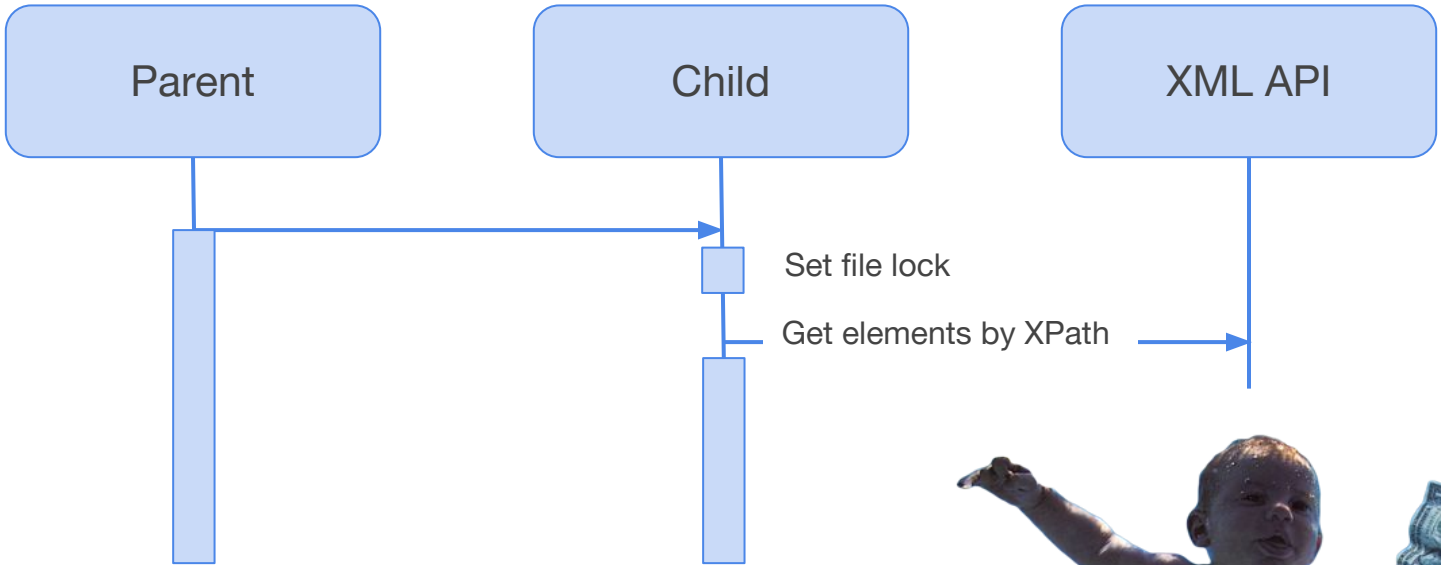
```
at org.apache.xpath.axes.AxesWalker.getNextNode(AxesWalker.java:335)
```

```
at org.apache.xpath.axes.AxesWalker.nextNode(AxesWalker.java:363)
```

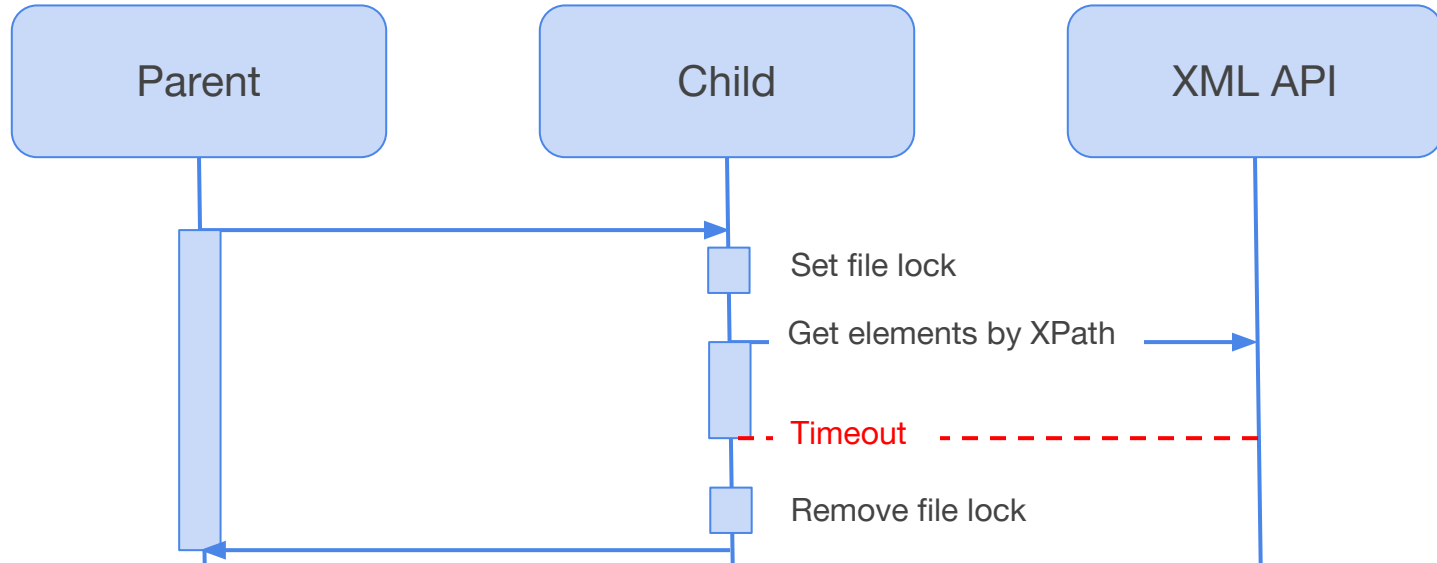




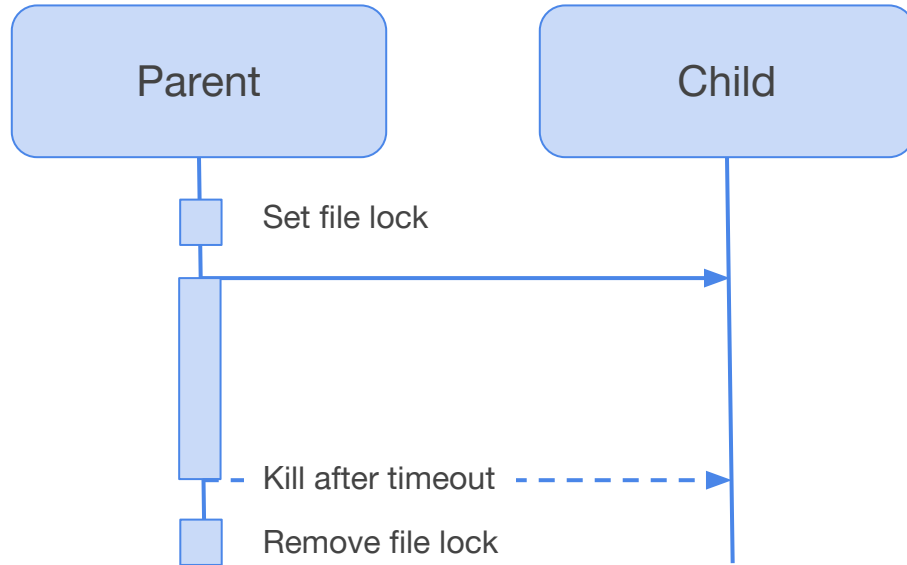




# Quick fix

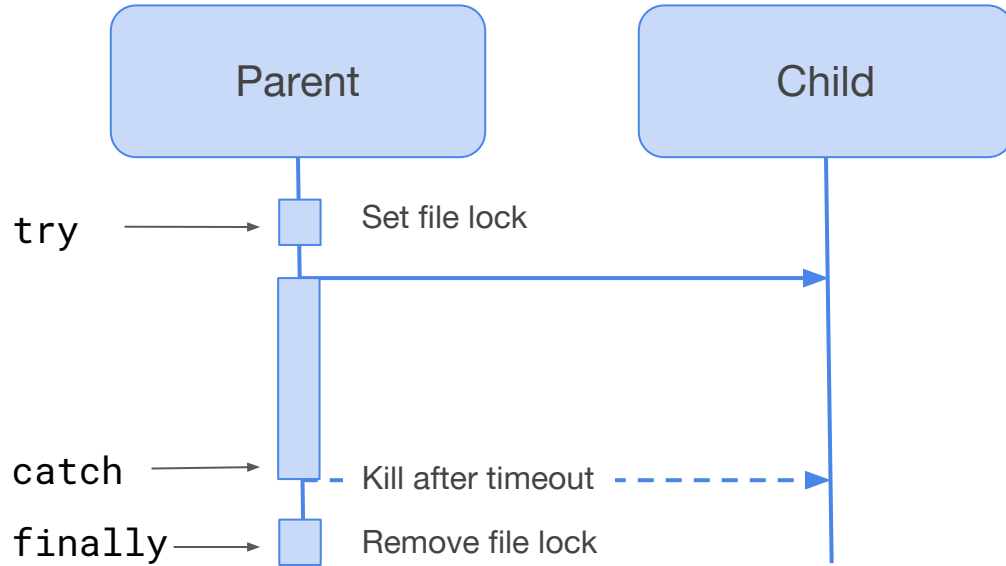


# Better design





# Better design



```
try (FileLock lock = file.getChannel().lock()) {  
    Process child = new ProcessBuilder(...)  
        .start();  
    if (!child.waitFor(..., TimeUnit.SECONDS)) {  
        child.destroy();  
    }  
} catch (IOException | InterruptedException ex) {  
}
```

```
try (FileLock lock = file.getChannel().lock()) {
```

Since Java 8



```
    Process child = new ProcessBuilder(...)
```

```
        .start();
```

```
    if (!child.waitFor(..., TimeUnit.SECONDS)) {
```

```
        child.destroy();
```

```
    }
```

```
} catch (IOException | InterruptedException ex) {
```

```
}
```



Appetizer

Crash after  
flying 40 minutes

**NEW VERSION!**



**WORKS FOR  
ALL CUSTOMERS!**



One customer



One customer



One customer





One customer



```
# Problematic frame:  
# V [libjvm.so+0xa2f731]
```

**HAVE YOU TRIED**

**TURNING IT OFF AND ON AGAIN?**

Try again!



Try again!



Try again!



Try again!



```
# Problematic frame:  
# C [libzip.so+0xf1cf]
```

Quick solution

# Downgrade

**Reproduce**





*Git bisect !*

OS?

Zip library?

JDK  
version?

Bug?  
Where???

```
git checkout <problematic-version>
```

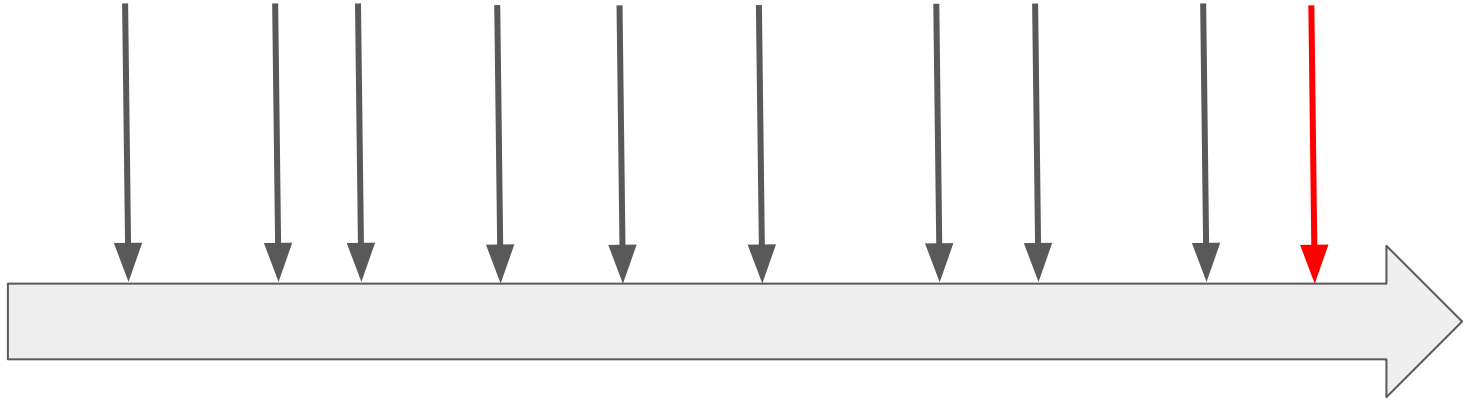
```
git bisect start
```

```
git bisect bad
```

```
git bisect good <stable-version>
```

```
Bisecting: 80 revisions left to test after  
this (roughly 6 steps)
```

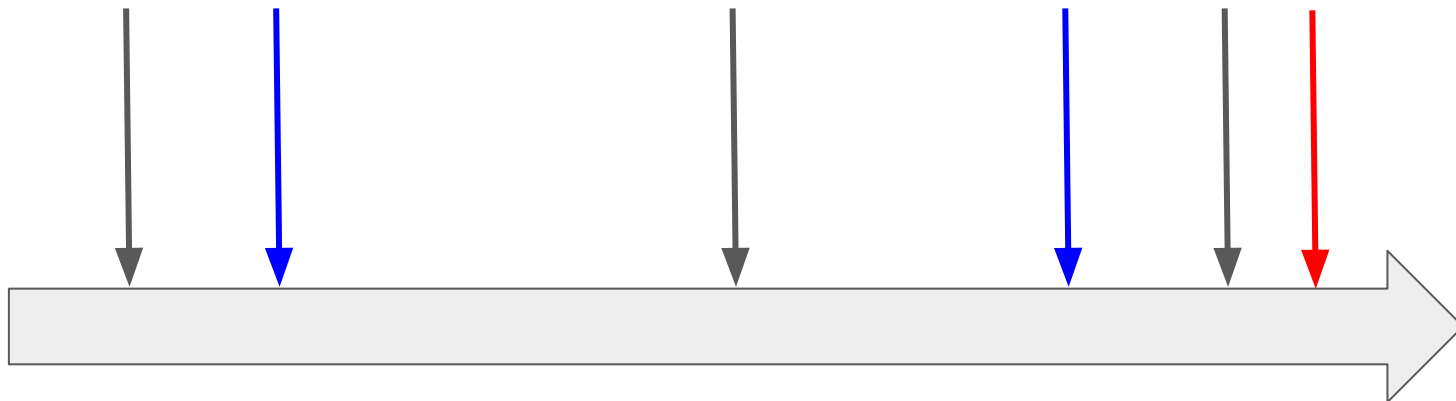
# Old version





→ `response.getWriter().write("<page...")`

→ `response.getWriter().flush()`

## New version

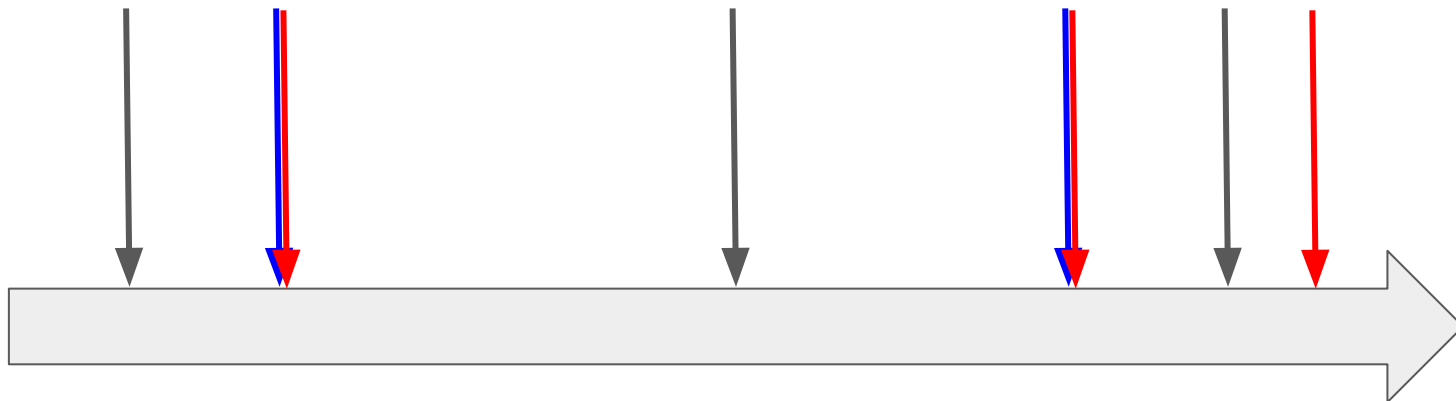



 `response.getWriter().write(...)`


 `objectMapper.writeValue(response.getWriter())`

 `response.getWriter().flush()`

## New version



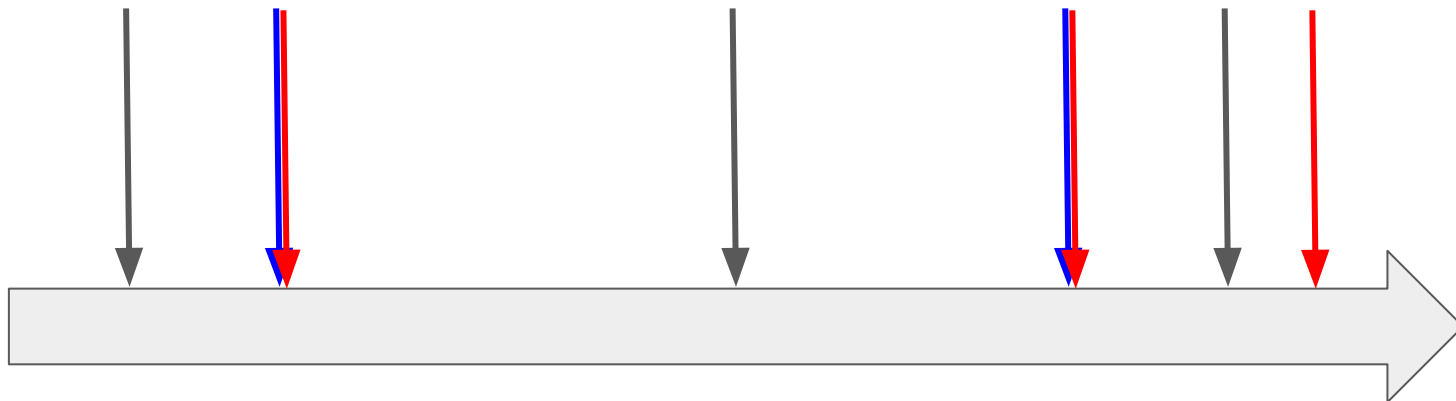
 `response.getWriter().write(...)`


 `objectMapper.writeValue(response.getWriter())`


 `response.getWriter().flush()`

New version

10000



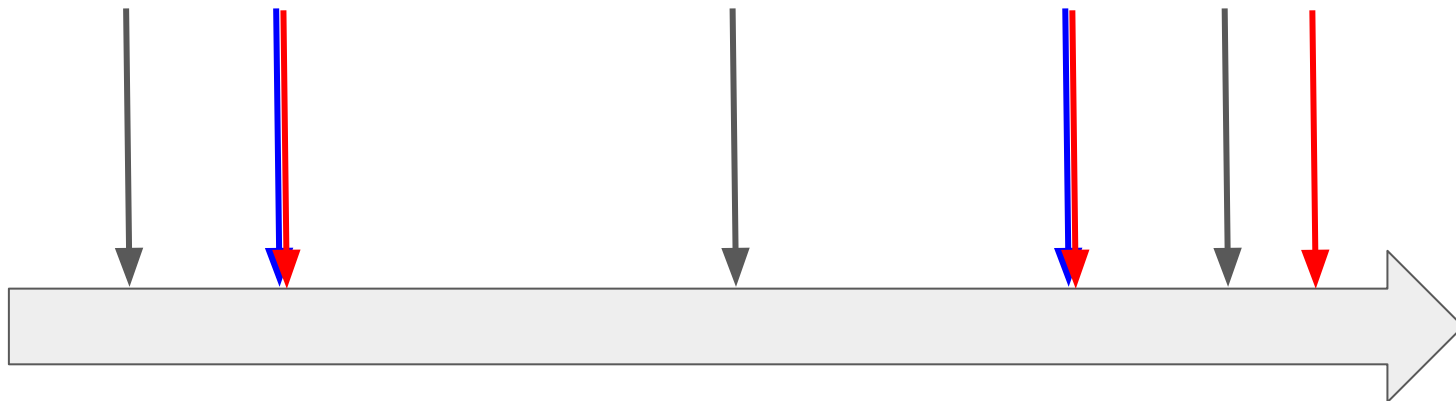
 `response.getWriter().write(...)`


 `objectMapper.writeValue(response.getWriter())`


 `response.getWriter().flush()`

New version

11000



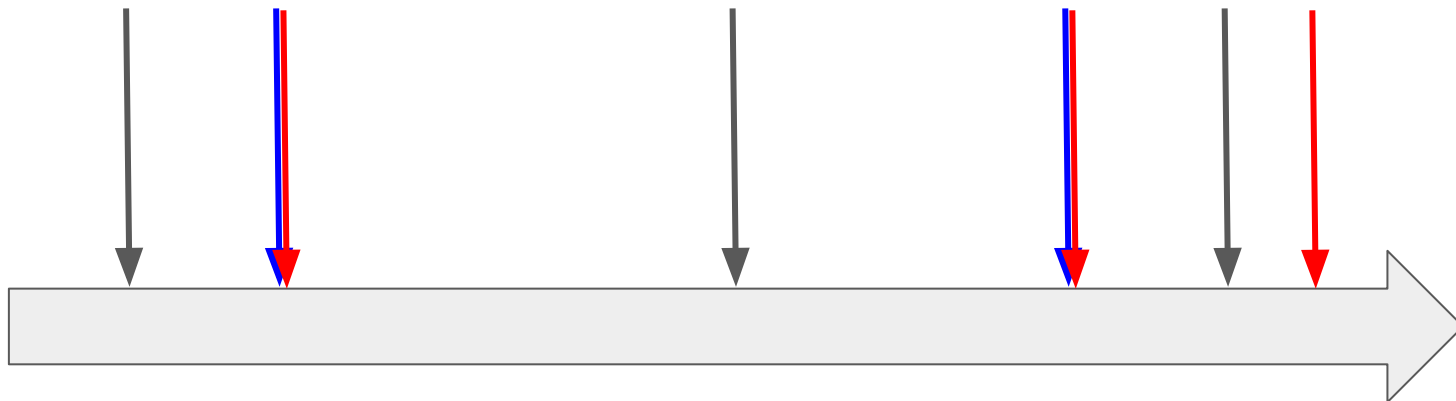
 `response.getWriter().write(...)`


 `objectMapper.writeValue(response.getWriter())`


 `response.getWriter().flush()`

New version

12000



 `response.getWriter().write(...)`

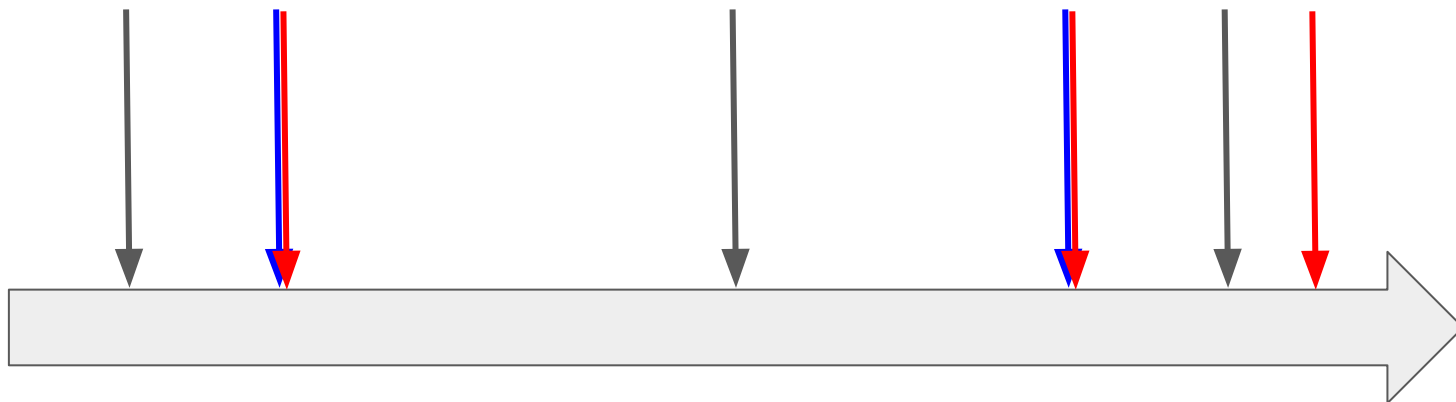
 `objectMapper.writeValue(response.getWriter())`


 `response.getWriter().flush()`




New version

13000

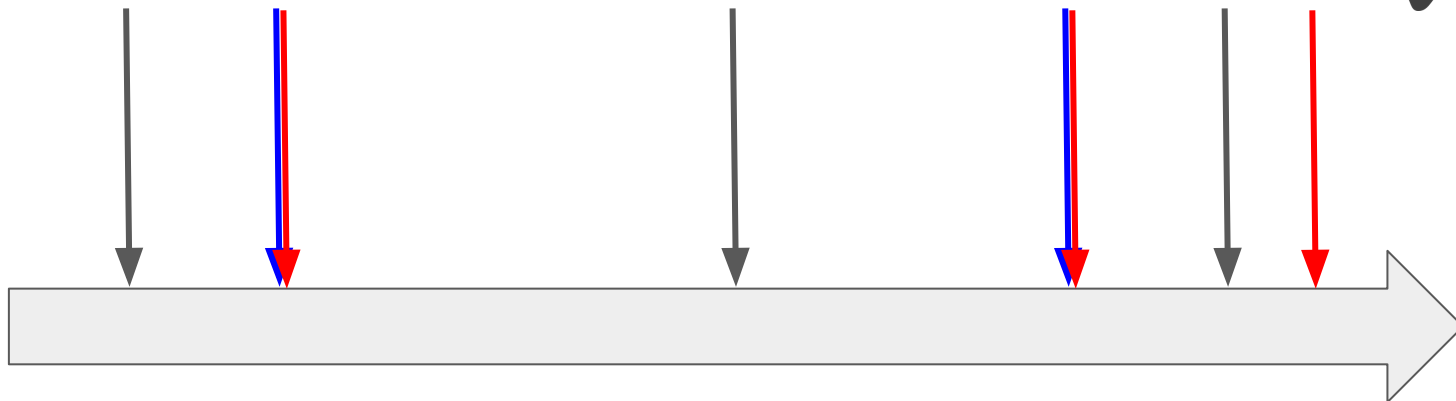


 `response.getWriter().write(...)`

 `objectMapper.writeValue(response.getWriter())`

 `response.getWriter().flush()`

New version



`response.getWriter().write(...)`

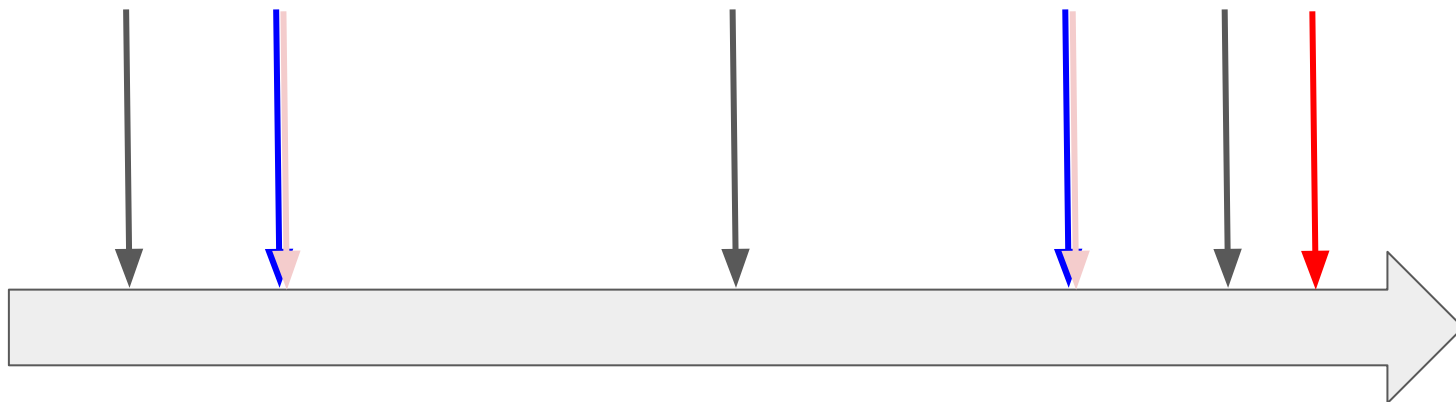



`objectMapper.writeValue(response.getWriter())`




`response.getWriter().flush()`

## Fix - defer flush



 `response.getWriter().write(...)`

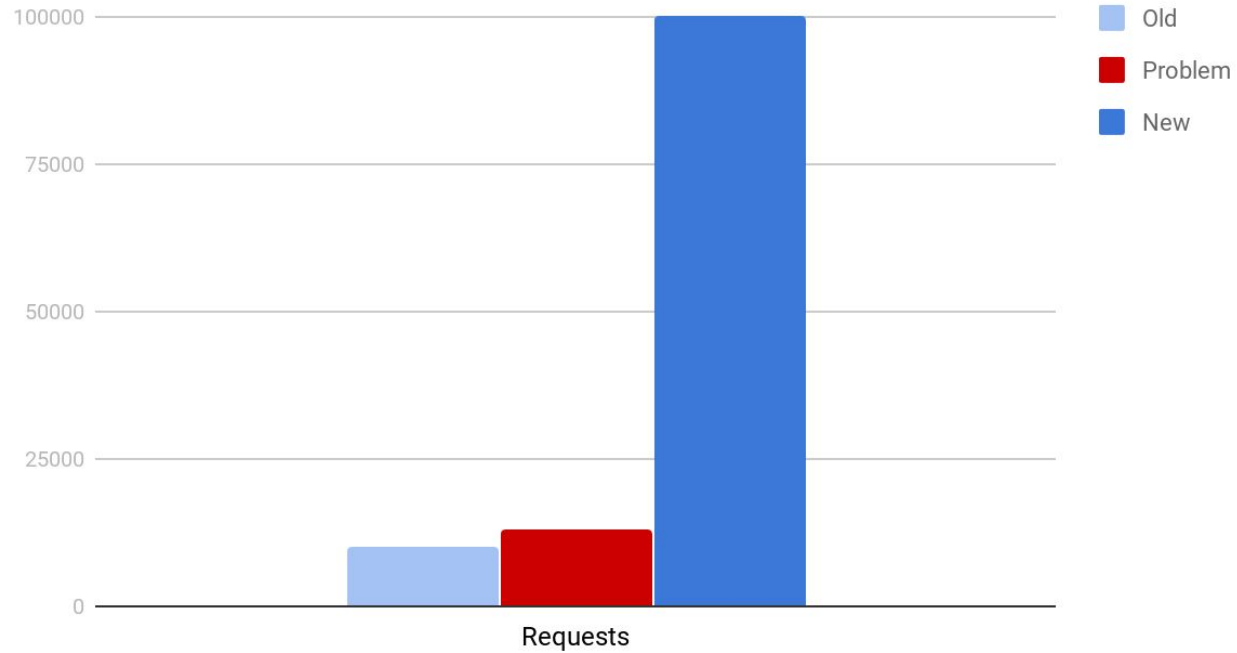
 `objectMapper.writeValue(response.getWriter())`

 `response.getWriter().flush()`

```
final class DeferringWriter extends BufferedWriter {  
  
    public DeferringWriter(Writer out, int size) {  
        super(out, size);  
    }  
  
    @Override public void flush() throws IOException {}  
  
    @Override public void close() throws IOException {}  
  
    void complete() throws IOException {  
        super.flush();  
        super.close();  
    }  
}
```

# Process change

## Acceptance test





Know your ~~enemy~~ library



Easy rollbacks



Tougher tests





Entree

Less is more

One customer





**HAVE YOU TRIED**

**TURNING IT OFF AND ON AGAIN?**

One customer

**Restart**

One customer

# Upscale

One customer

**Upseate**

Quick solution

**Replace  
servers**



*jstacks?*

CPU is low

GC is low

Restart  
**doesn't** help

VM  
replacement  
**does** help

java.lang.Thread.State: RUNNABLE

at java.io.RandomAccessFile.readBytes(Native Method)

at java.io.RandomAccessFile.read(RandomAccessFile.java:377)

java.lang.Thread.State: RUNNABLE

at java.io.RandomAccessFile.readBytes(Native Method)

at java.io.RandomAccessFile.read(RandomAccessFile.java:377)

java.lang.Thread.State: RUNNABLE

at java.io.RandomAccessFile.readBytes(Native Method)

at java.io.RandomAccessFile.read(RandomAccessFile.java:377)

java.lang.Thread.State: RUNNABLE

at java.io.RandomAccessFile.readBytes(Native Method)

at java.io.RandomAccessFile.read(RandomAccessFile.java:377)

```
java.lang.Thread.State: RUNNABLE
```

```
at java.io.RandomAccessFile.readBytes(Native Method)
```

```
at java.io.RandomAccessFile.read(RandomAccessFile.java:377)
```

```
java.lang.Thread.State: RUNNABLE
```

```
at java.io.RandomAccessFile.readBytes(Native Method)
```

```
at java.io.RandomAccessFile.read(RandomAccessFile.java:377)
```

```
java.lang.Thread.State: RUNNABLE
```

```
at java.io.RandomAccessFile.readBytes(Native Method)
```

```
at java.io.RandomAccessFile.read(RandomAccessFile.java:377)
```

```
java.lang.Thread.State: RUNNABLE
```

```
at java.io.RandomAccessFile.readBytes(Native Method)
```

```
at java.io.RandomAccessFile.read(RandomAccessFile.java:377)
```

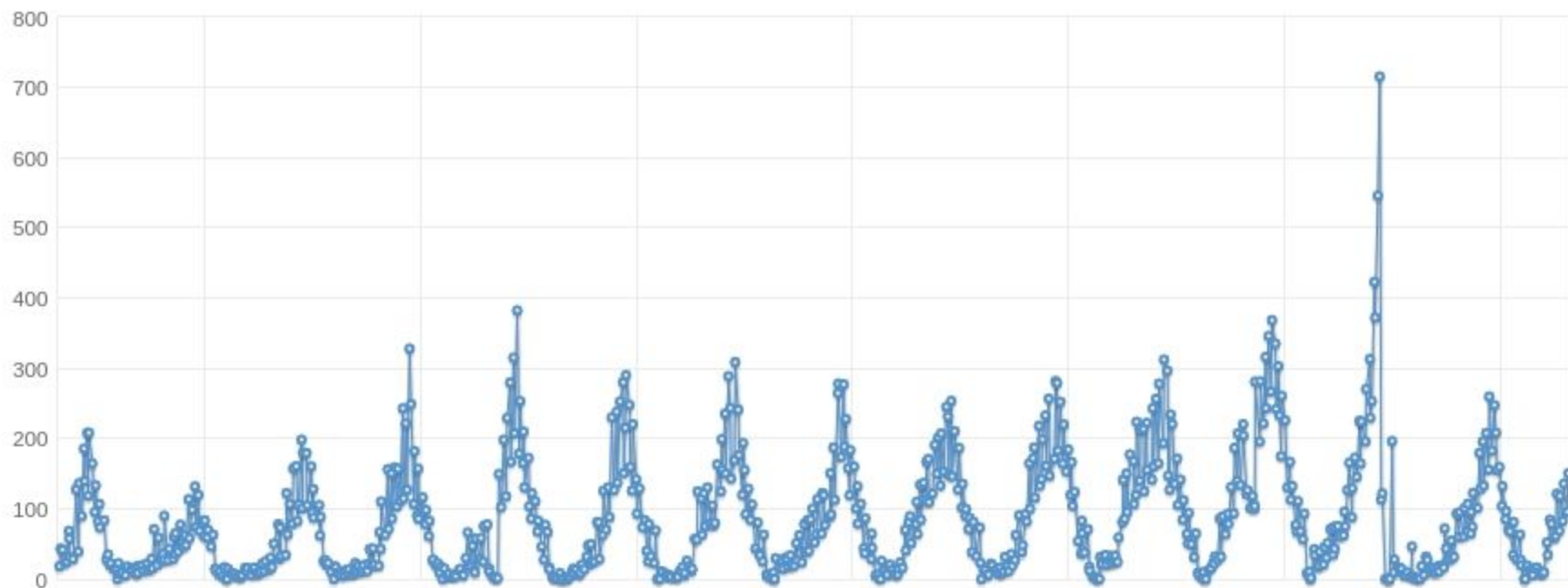


Read Throughput ( Ops/s )

Statistic:

Time Range:

Period:

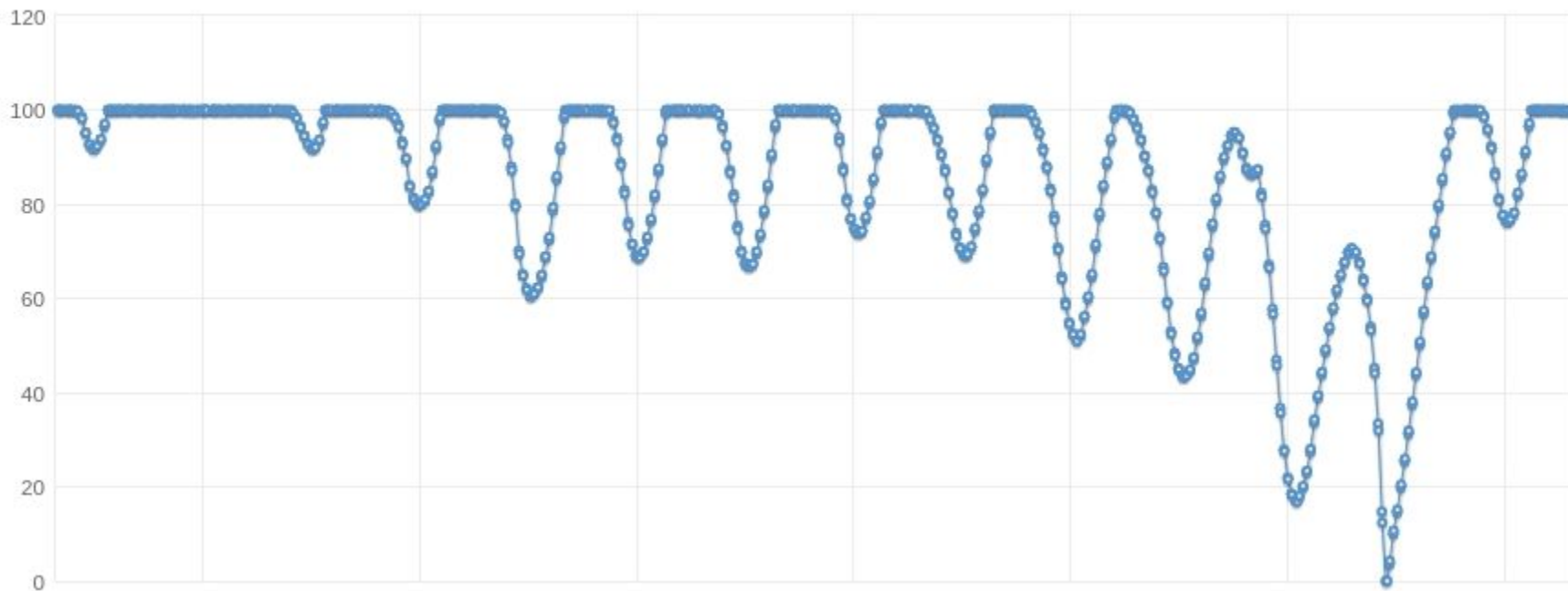


Burst Balance ( Percent )

Statistic: Minimum

Time Range: Last 2 Weeks

Period: 5 Minutes



Small disk



Small disk



More I/O  
per request



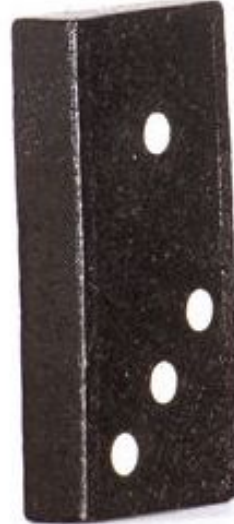
Small disk



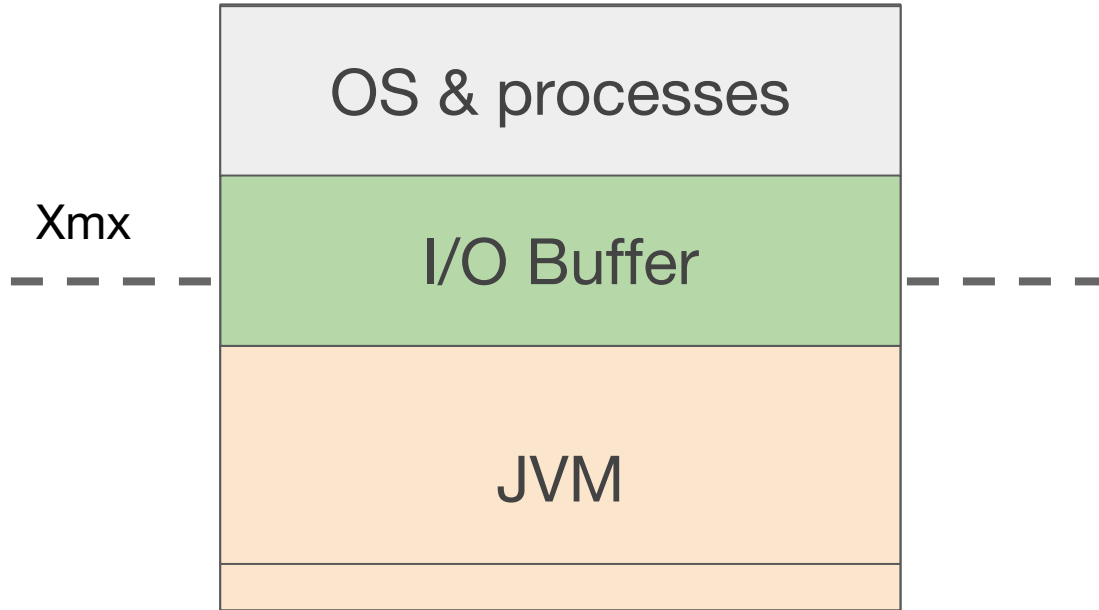
More I/O  
per request



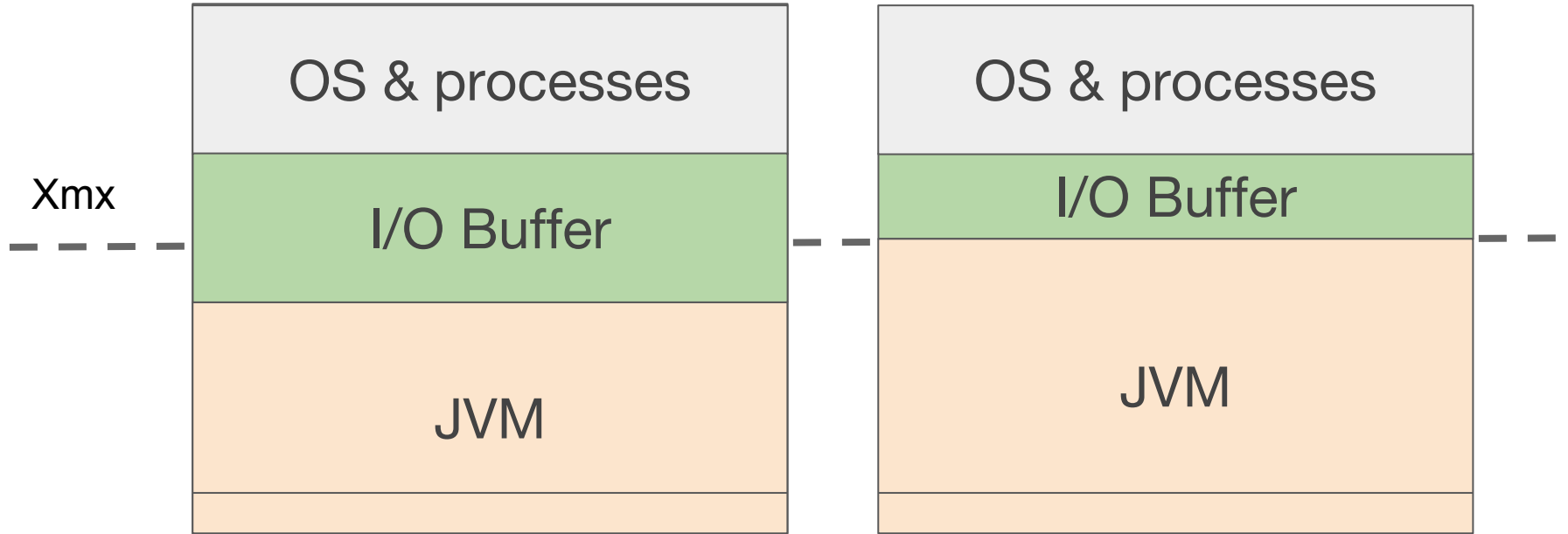
More  
requests



# Server Memory



# Server Memory



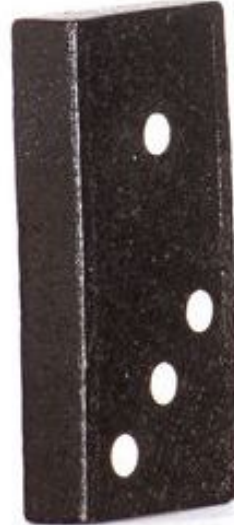
Small disk



More I/O  
per request



More  
requests



Small I/O  
buffer



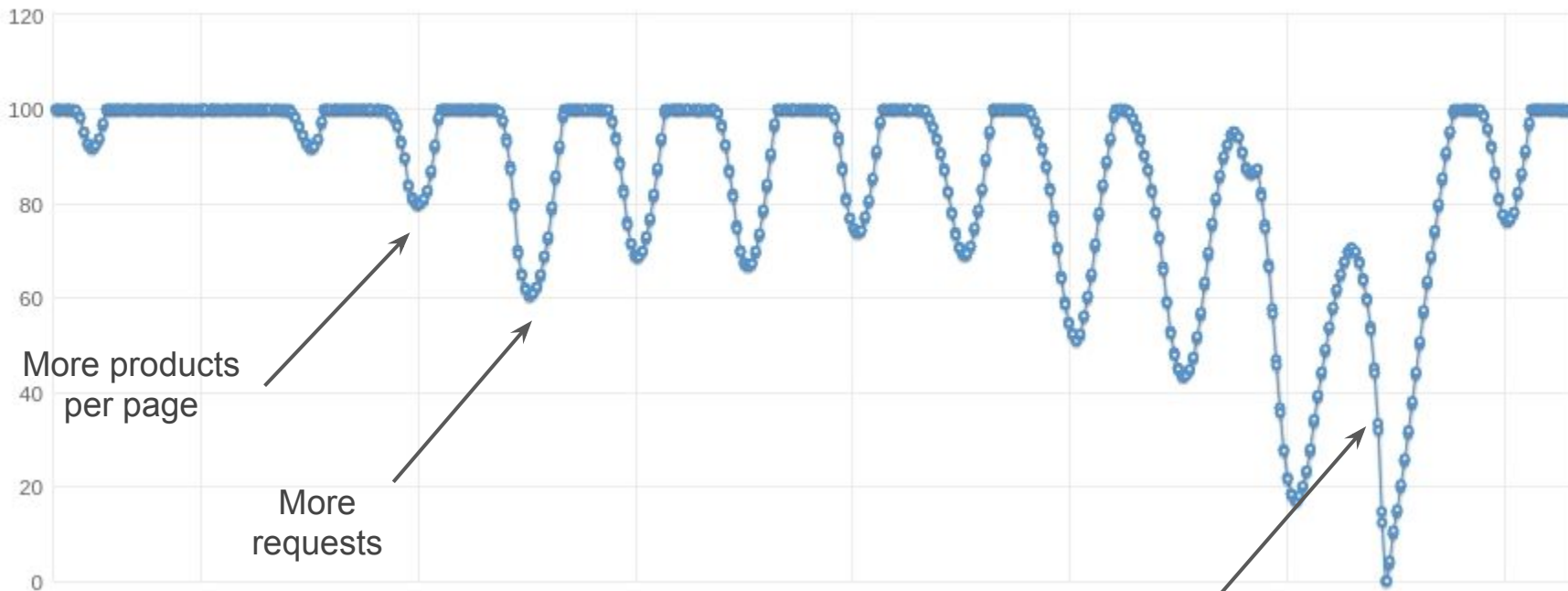


Burst Balance ( Percent )

Statistic: Minimum

Time Range: Last 2 Weeks

Period: 5 Minutes

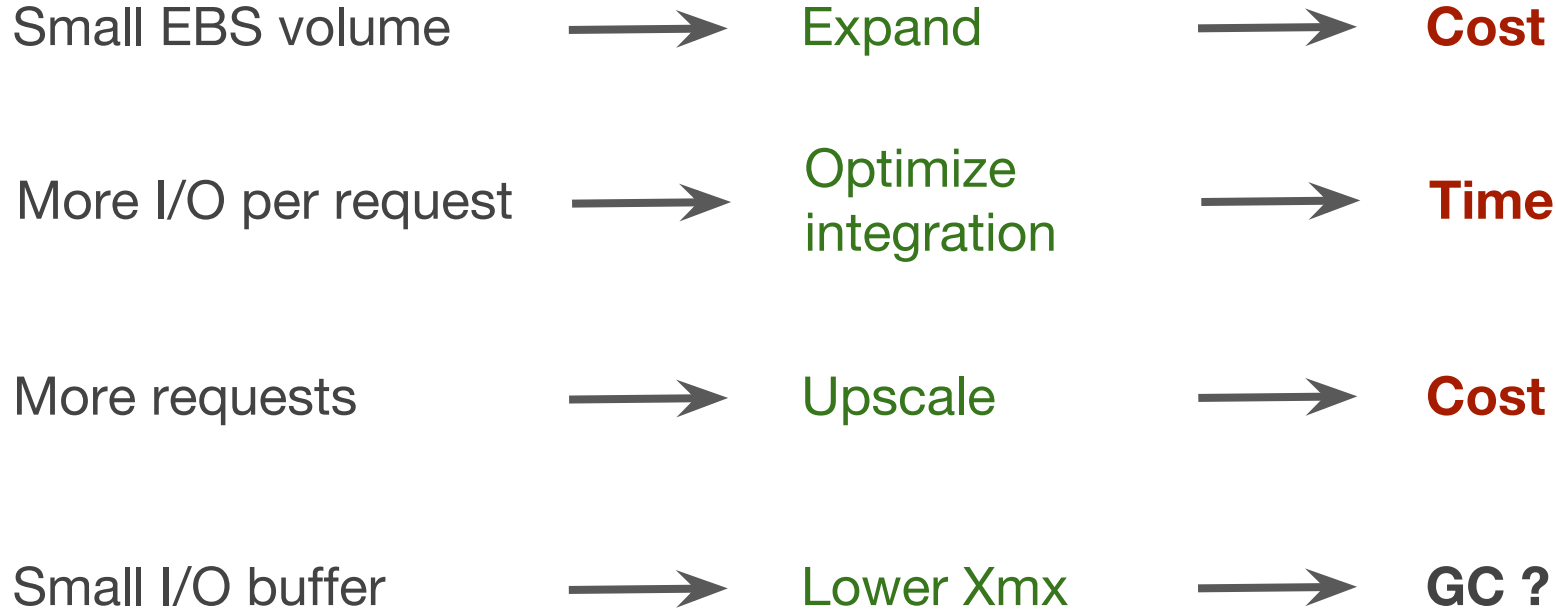


More products per page

More requests

JVM takes more memory

## Possible fixes



✓ Hard to catch by tests

✓ Monitoring + alerting

✓ Leave memory to OS!





Dessert

You need  
more!

Some customers



Hours



# Workaround



# Workaround



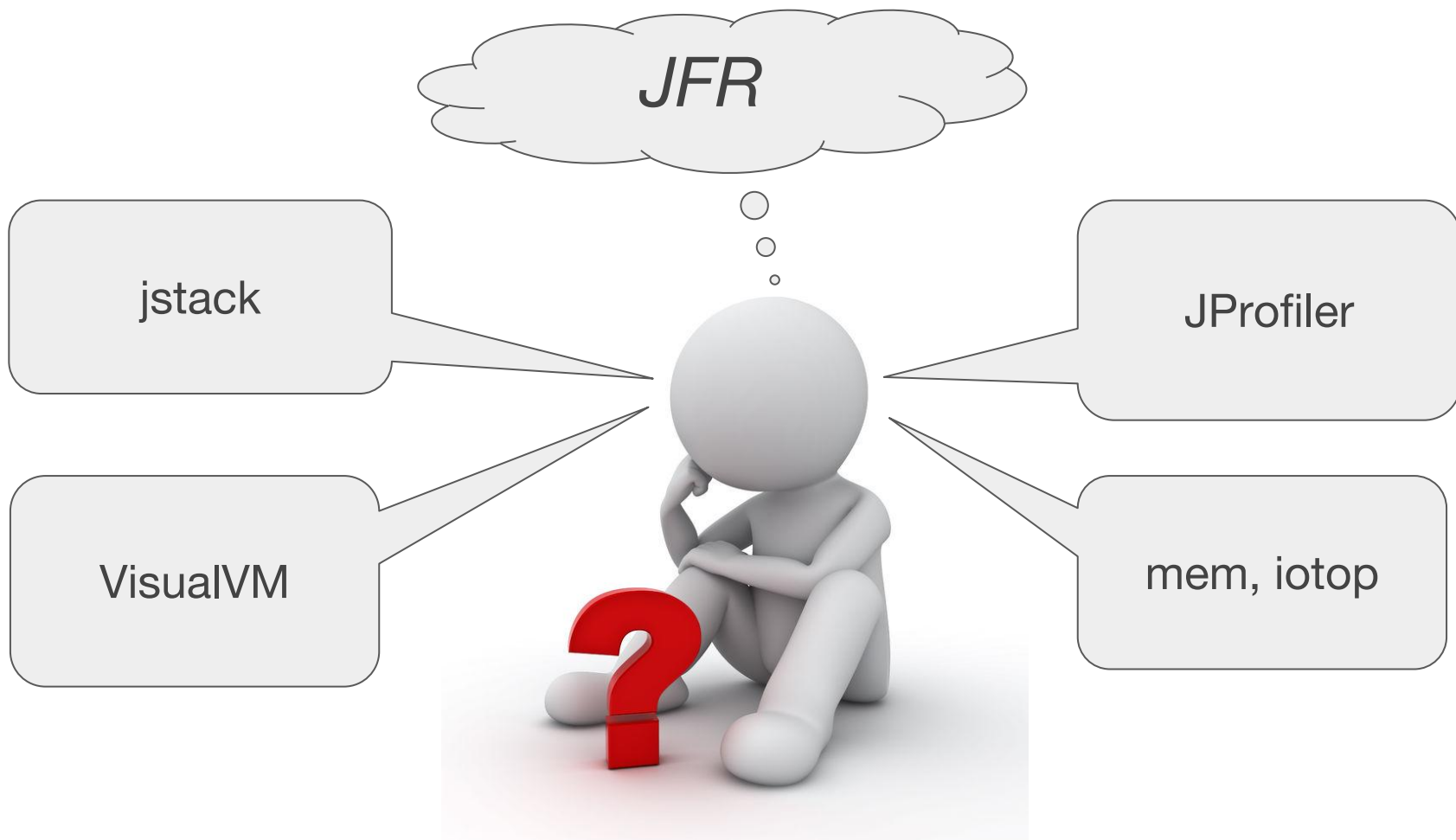
**Reproducible!**





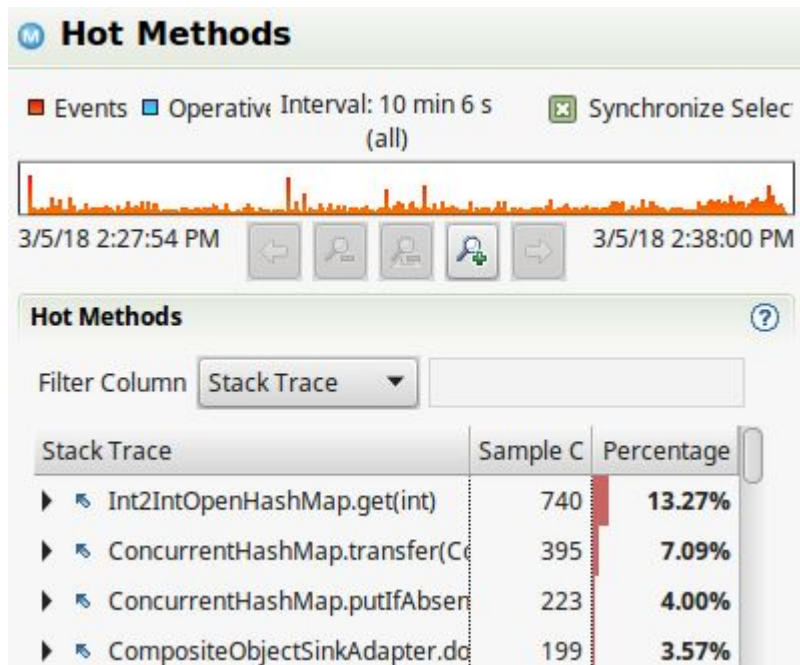
App bottleneck? Specific requests?  
Caches? JDK version?  
JVM parameters?





# Java Flight Recorder

Healthy



Degraded

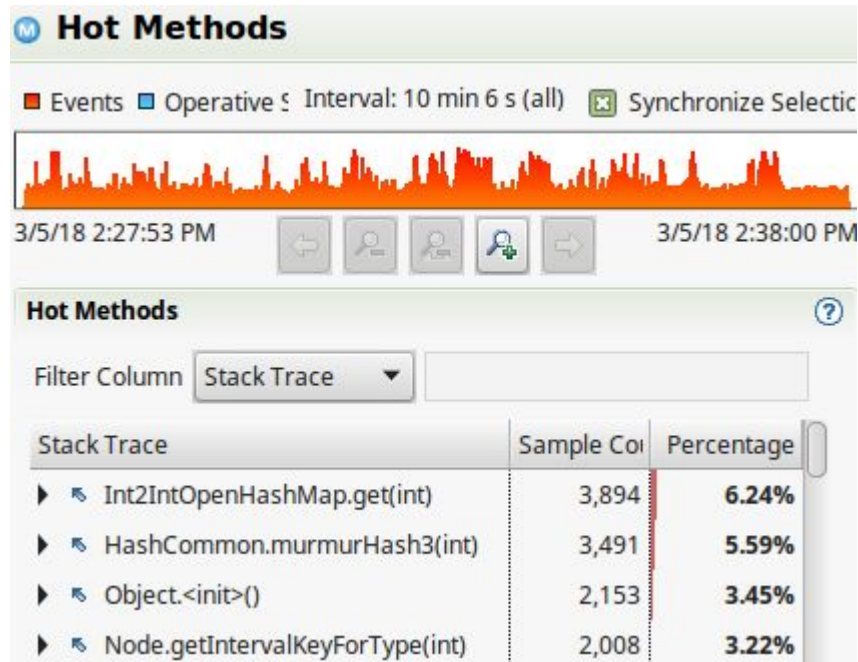
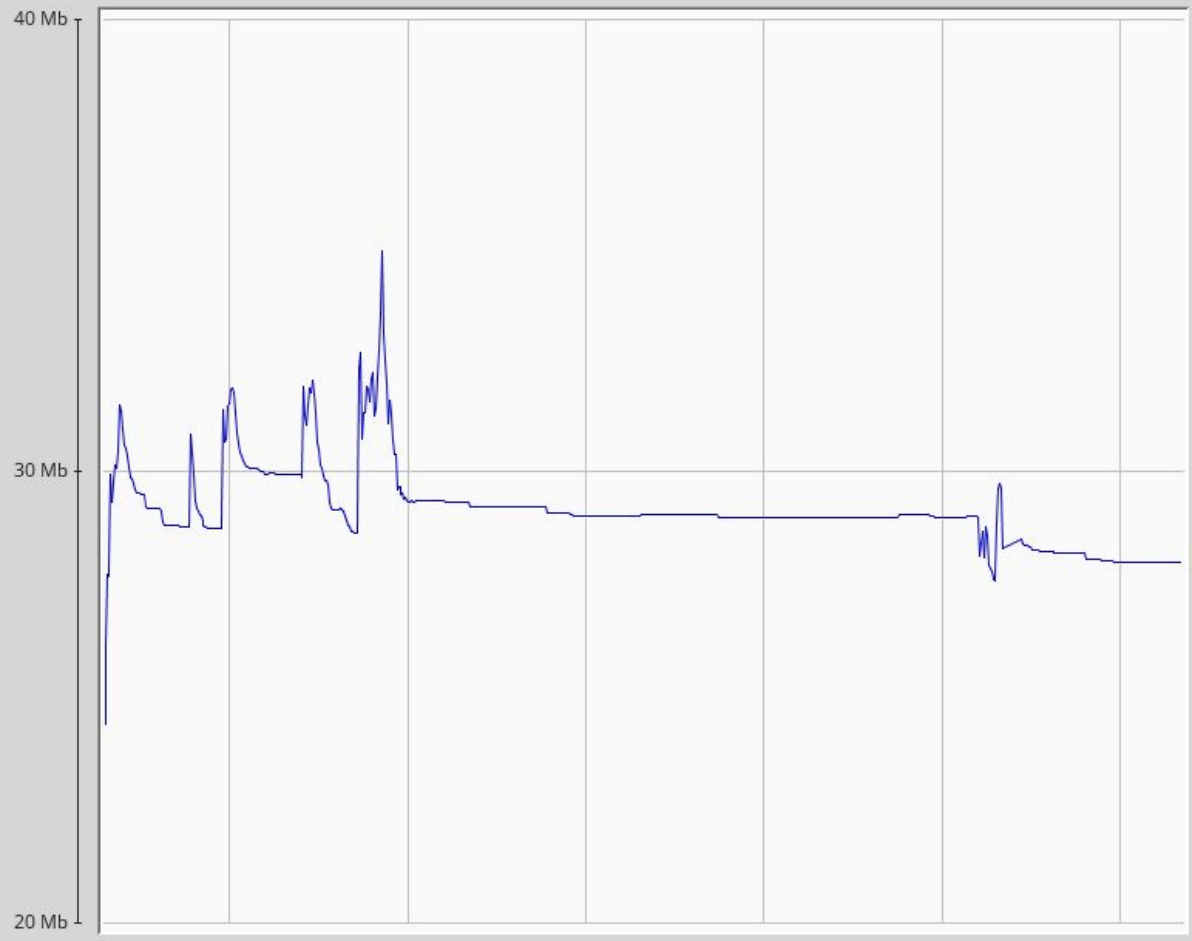


Chart: Memory Pool "Code Cache" ▼

Time Range: All ▼

Perform GC



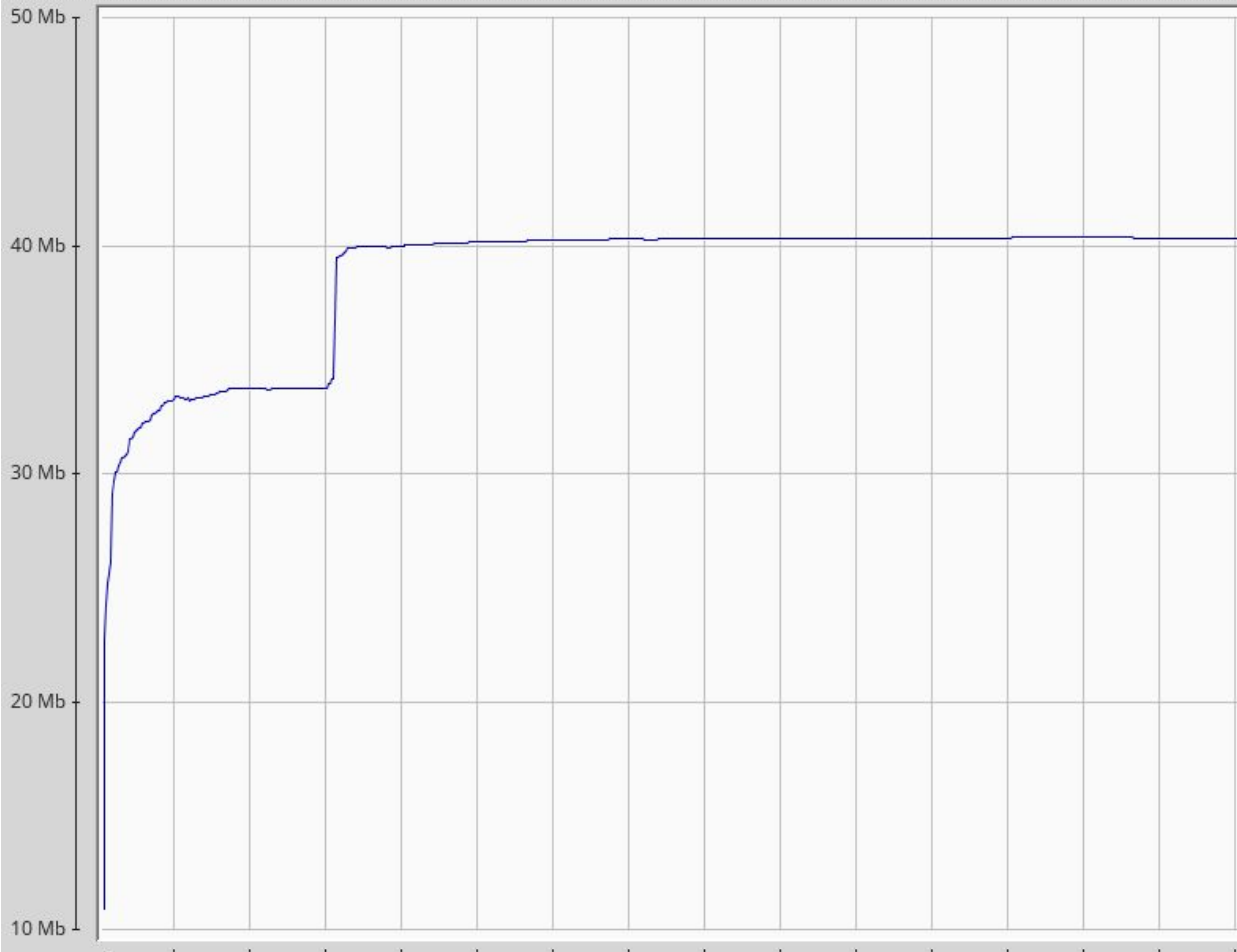


Fix

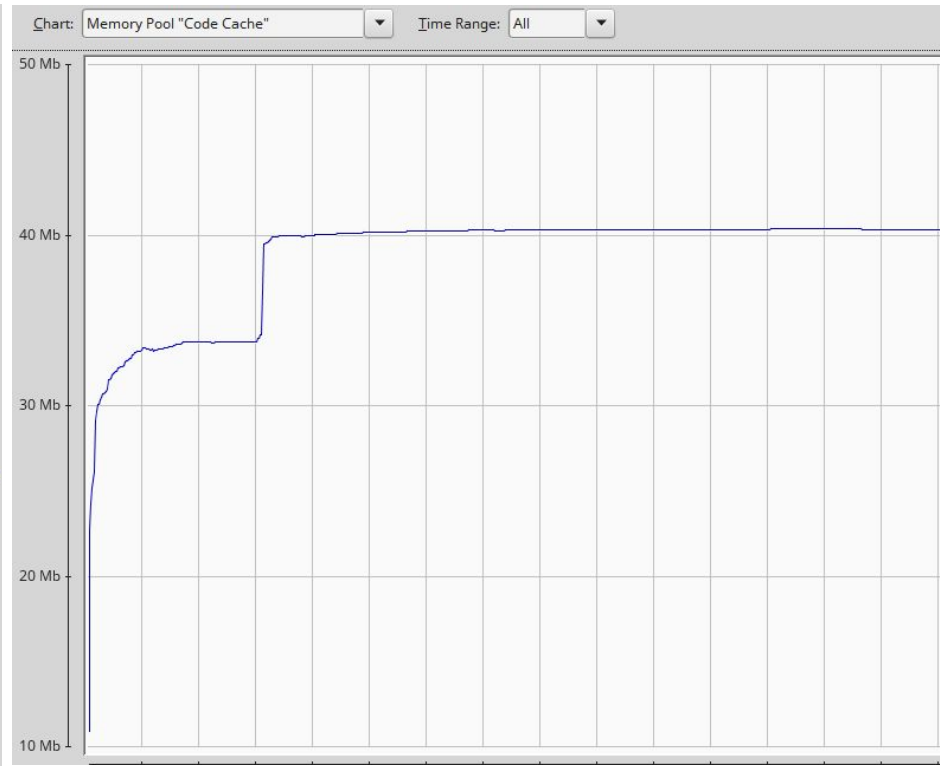
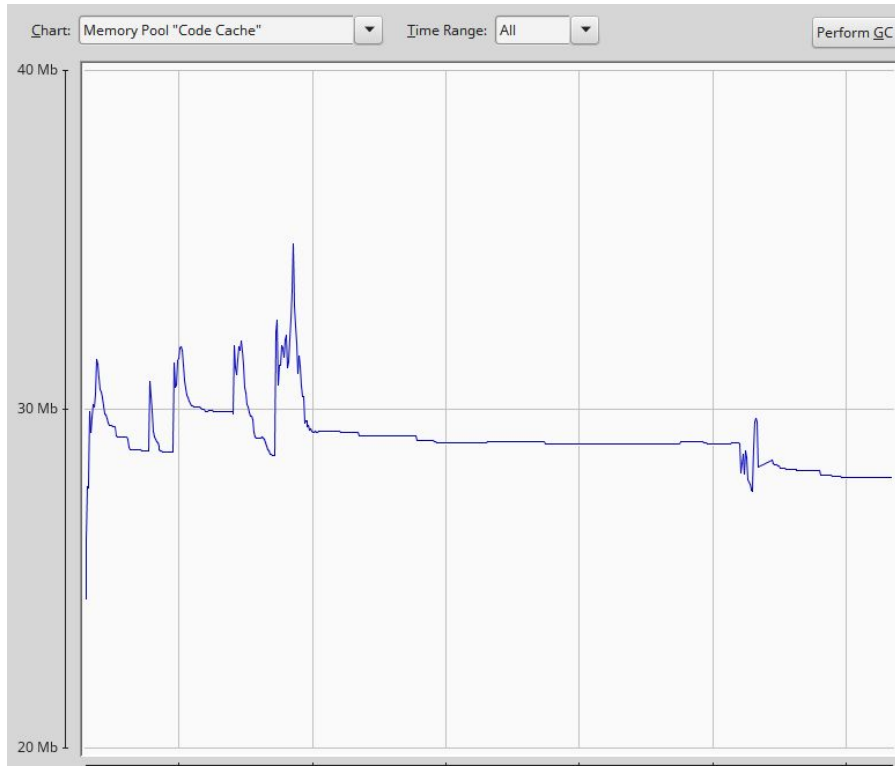
**-XX:ReservedCodeCacheSize=240M**

Chart: Memory Pool "Code Cache" ▼

Time Range: All ▼







# JVM Defaults

Java 7	Java 8
Server compilation	Tiered compilation
Code cache: 40 MB	Code cache: 240 MB



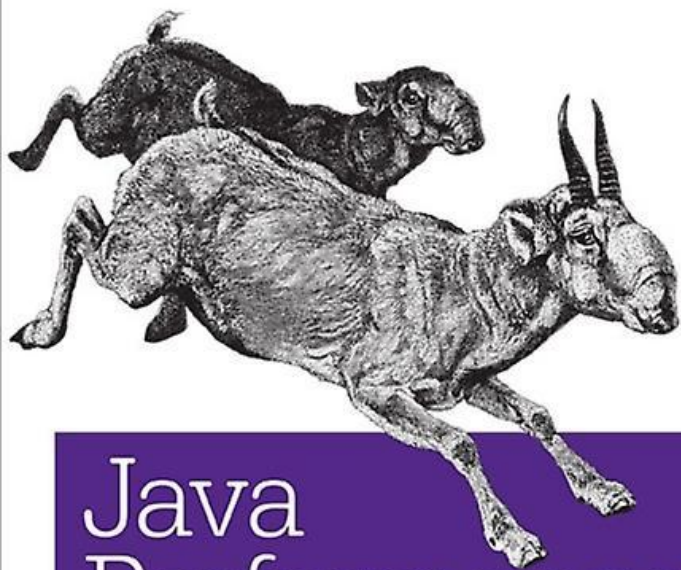
Toolbox



JVM tunings



O'REILLY®



# Java Performance The Definitive Guide

GETTING THE MOST OUT OF YOUR CODE

Scott Oaks



Compliment

Invest into  
knowledge!



Brown bag  
sessions



Library



Lazy days



Courses



Hackatons



Conferences



Thank  
you!