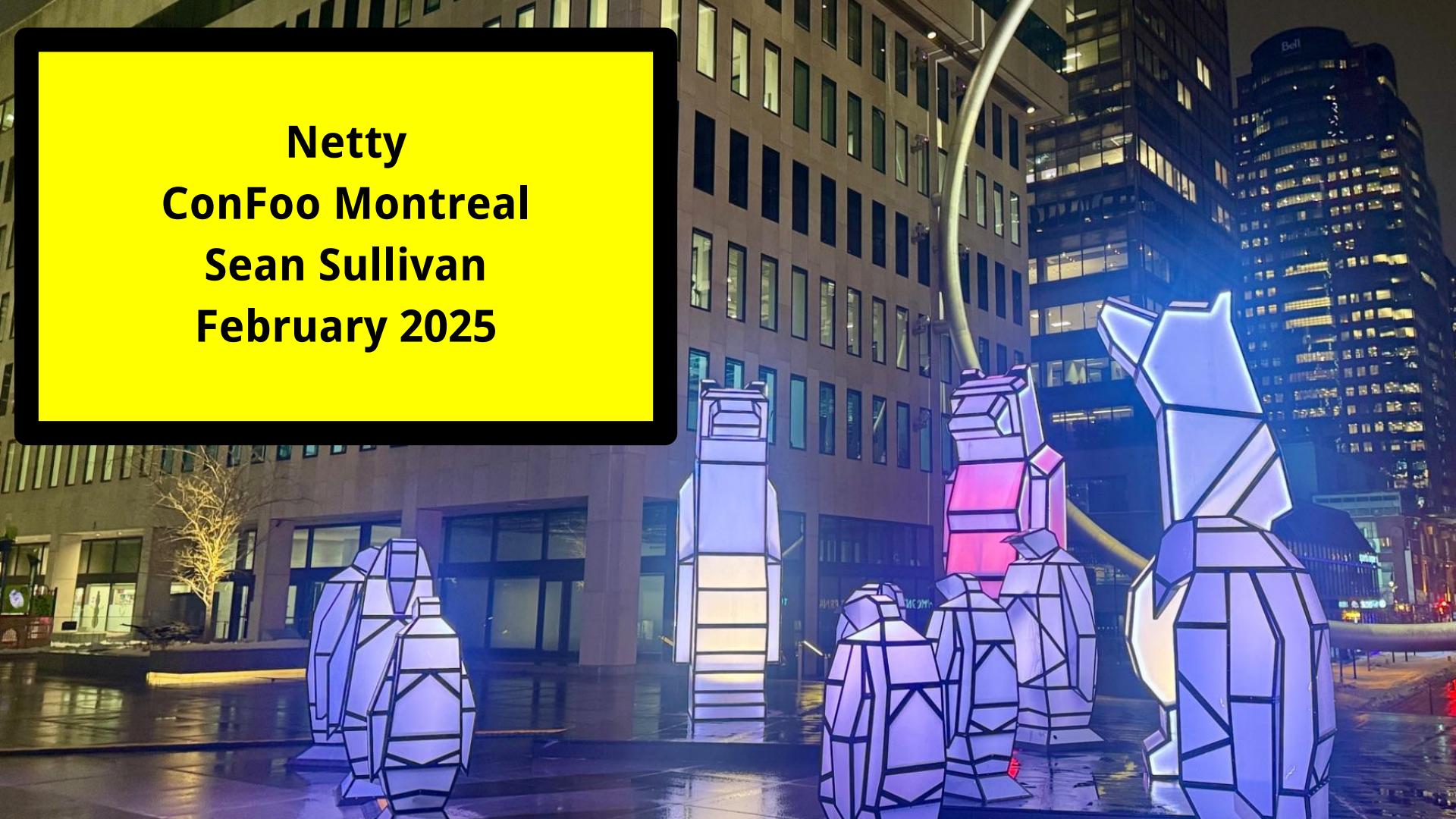
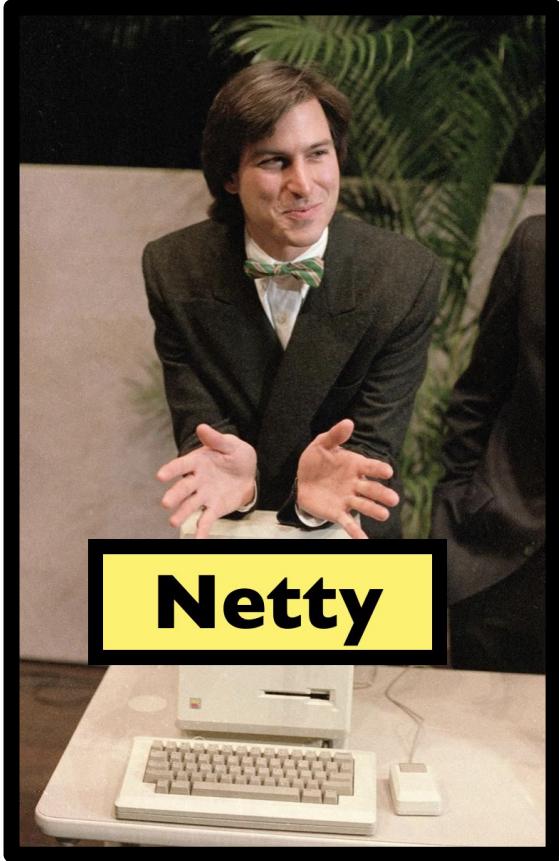


Netty
ConFoo Montreal
Sean Sullivan
February 2025



Agenda



- About me
- Netty project
- Netty in action

About me



Portland Oregon

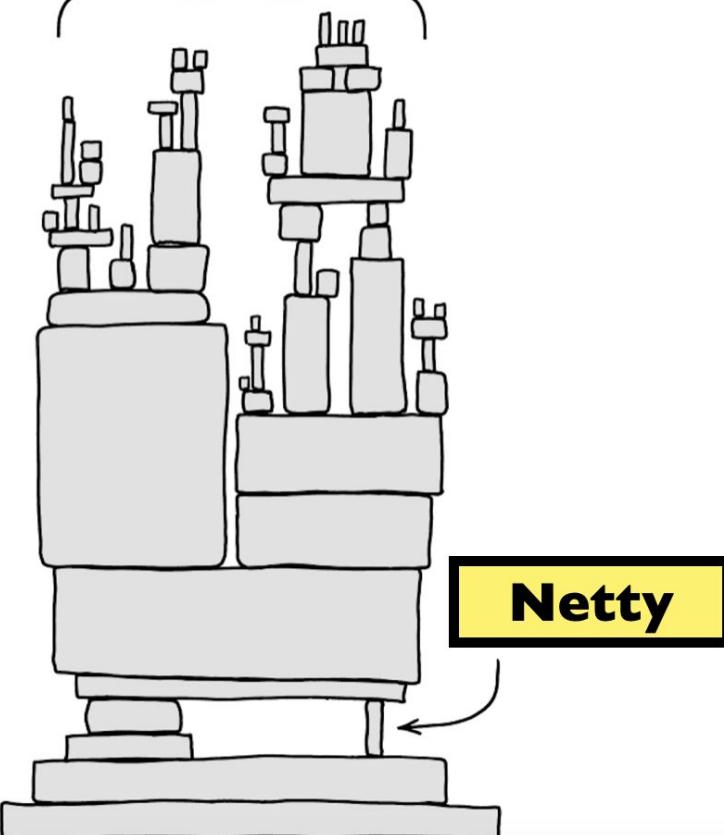


Java since 1996

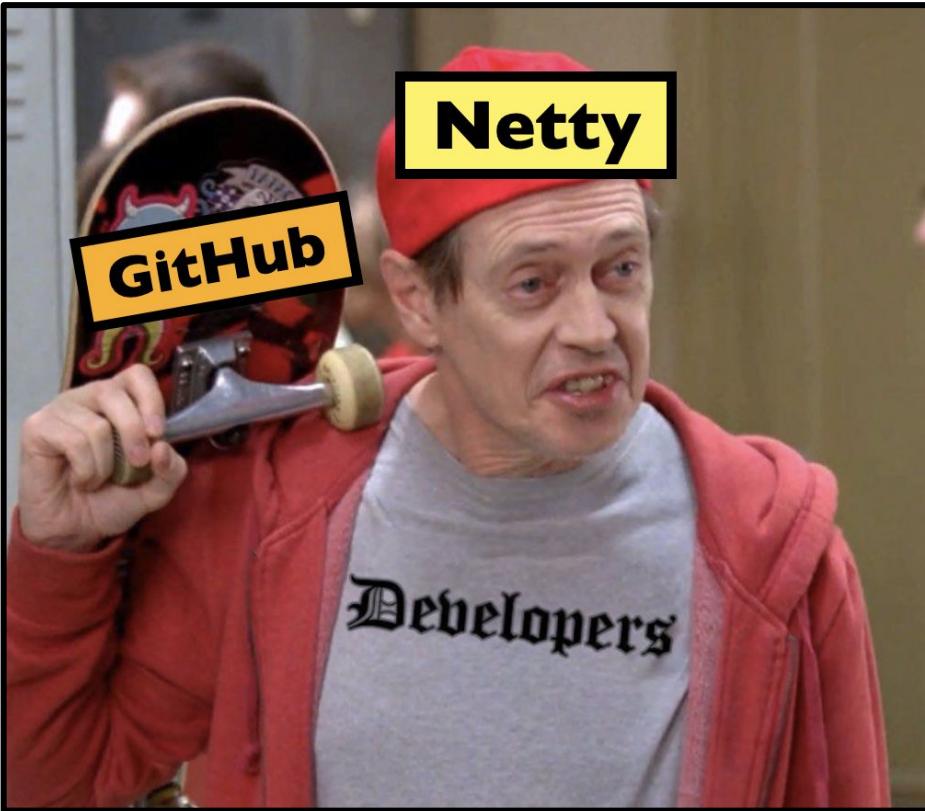


open source contributor

ALL MODERN DIGITAL
INFRASTRUCTURE



*“Netty is an
asynchronous
event-driven network
application framework”*



Netty is
open source

netty / netty



Netty contributor

- **add brotli4j native libs to [codec-http2] pom.xml**

#13177 by sullis was merged on Feb 2, 2023 • Approved ➔ 4.1.88.Final

- **fix brotli compression**

#13165 by sullis was merged on Feb 2, 2023 • Changes requested ➔ 4.1.88.Final

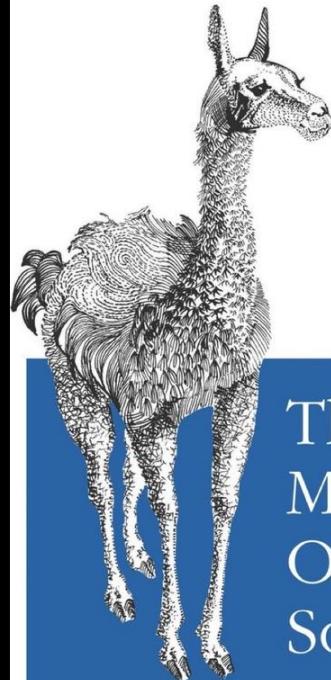
- **add brotli4j aarch64 dependencies**

#13143 by sullis was merged on Jan 23, 2023 • Approved ➔ 4.1.88.Final

- **enable brotli4j in example applications**

#13131 by sullis was merged on Jan 26, 2023 • Changes requested ➔ 4.1.88.Final

Acting out of the goodness of your heart, or something



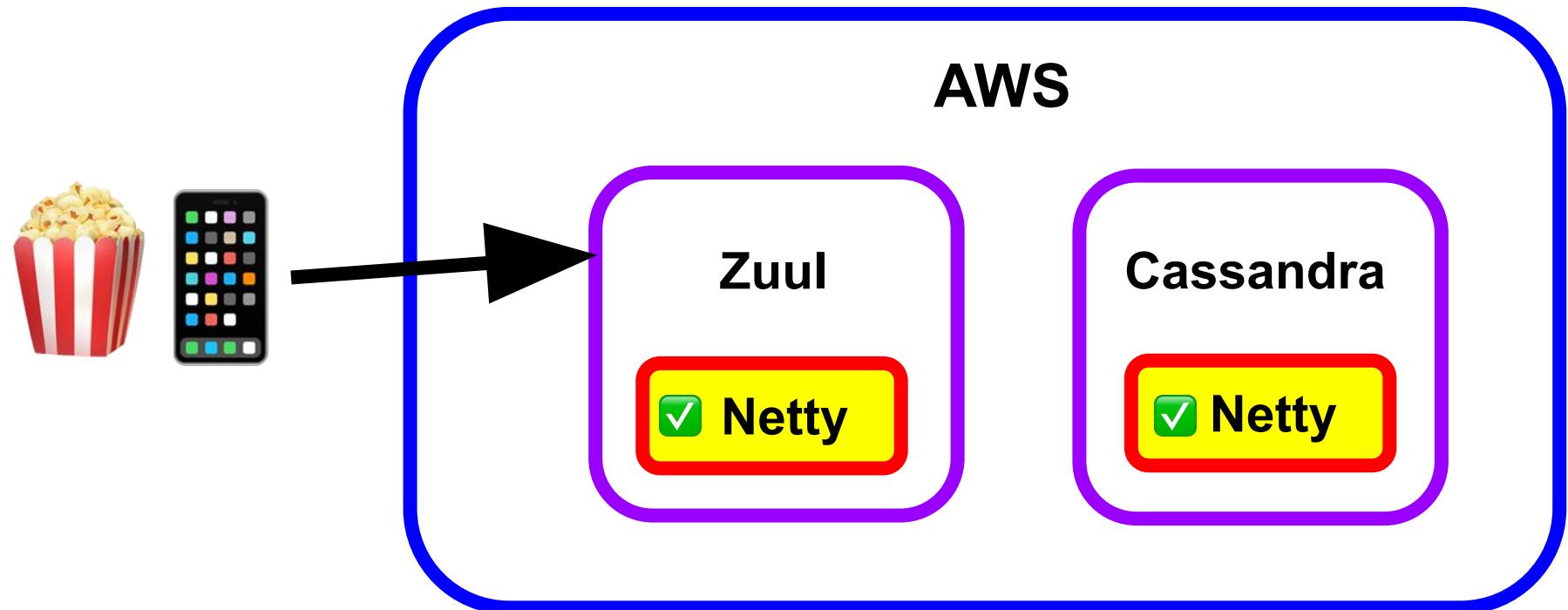
Thanklessly
Maintaining
Open Source
Software

O RLY?

@ThePracticalDev

-  **async-http-client**
-  **reactor-netty**
-  **Netflix Zuul**
-  **Apache Pinot**

Netty at Netflix



Zuul is built on Netty

zuul / zuul-core

```
package com.netflix.zuul.netty.server;
```

Zuul: Server.java

```
import io.netty.bootstrap.ServerBootstrap;
```

```
ServerBootstrap serverBootstrap =  
    new ServerBootstrap().group(
```

Zuul server channel initializer

```
protected void initChannel(Channel ch) throws Exception {  
    // Configure our pipeline of ChannelHandlers.  
    ChannelPipeline pipeline = ch.pipeline();  
  
    storeChannel(ch);  
    addTimeoutHandlers(pipeline);  
    addPassportHandler(pipeline);  
    addTcpRelatedHandlers(pipeline);  
    addHttp1Handlers(pipeline);  
    addHttpRelatedHandlers(pipeline);  
    addZuulHandlers(pipeline);  
}
```

Zuul channel handlers

```
pipeline.addLast(new IdleStateHandler(0, 0,  
    idleTimeout, TimeUnit.MILLISECONDS));
```

```
pipeline.addLast(new CloseOnIdleStateHandler(registry, metricId));
```



Zuul: close on idle handler

```
public class CloseOnIdleStateHandler  
    extends ChannelInboundHandlerAdapter {  
  
    private final Counter counter;
```

Zuul: close on idle handler

```
@Override  
public void userEventTriggered(ChannelHandlerContext ctx, Object evt)  
    super.userEventTriggered(ctx, evt);  
  
    if (evt instanceof IdleStateEvent) {  
        counter.increment();  
        ctx.close();  
    }  
}
```

Zuul push messaging

AWS re:Invent

ARC334

Scaling Push Messaging for Millions of Netflix Devices

Susheel Aroskar
Senior Software Engineer
Netflix Inc/Edge Gateway

aws re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

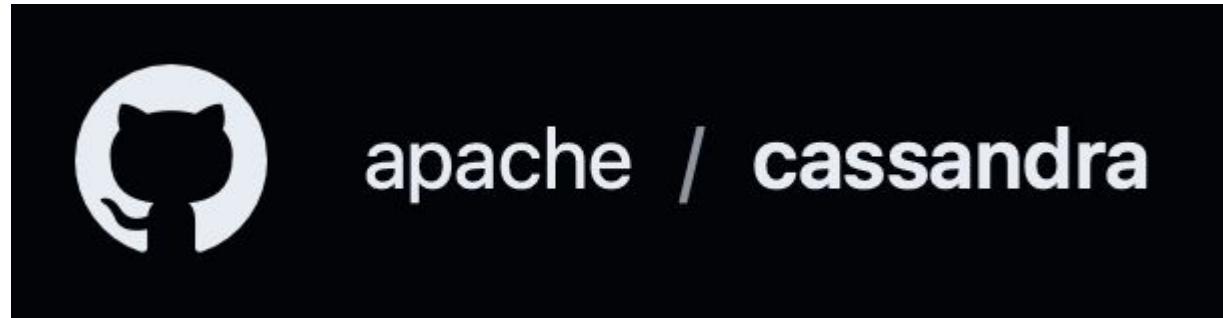
AWS re:Invent 2018: Scaling Push Messaging for Millions of Netflix Devices (ARC334)

Zuul WebSockets

```
import io.netty.handler.codec.http.websocketx.TextWebSocketFrame;  
import io.netty.handler.codec.http.websocketx.WebSocketServerProtocolHandler;
```

```
@Override 1 usage  ↳ Susheel Aroskar  
public ChannelFuture sendPushMessage(ChannelHandlerContext ctx, ByteBuf mesg) {  
    final TextWebSocketFrame wsf = new TextWebSocketFrame(mesg);  
    return ctx.channel().writeAndFlush(wsf);  
}
```

Let's talk about Cassandra



Cassandra is built on Netty

cassandra / src / java

```
package org.apache.cassandra.net;
```

Cassandra: SocketFactory.java

```
import io.netty.bootstrap.ServerBootstrap;  
  
ServerBootstrap newServerBootstrap()  
{  
    return new ServerBootstrap().group(  
}  
;
```



- Apache ActiveMQ Artemis
- Apache Bookkeeper
- Apache Camel
- Apache Cassandra
- Apache CXF
- Apache Druid



- Apache Flink
- Apache NiFi
- Apache Pinot
- Apache Pulsar
- Apache Spark
- Apache Zookeeper



- Armeria
- Micronaut
- Netflix Zuul
- OpenSearch

And many others...

- Apple Servicetalk

- Azure SDK for Java

- AWS SDK for Java (v2)

- async-http-client

- grpc-java

- Expedia Styx

- Play Framework

- reactor-netty

- vert.x

- gatling



why is Netty popular?



blocking I/O



non blocking I/O

JDK classes

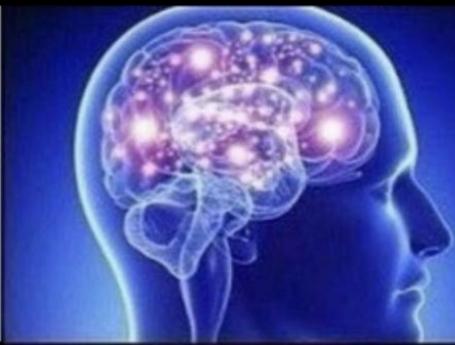
java.net

Class ServerSocket

java.nio.channels

Class ServerSocketChannel

`java.net`



`java.nio`



`io.netty`



Netty core concepts

io.netty.channel.Channel

A nexus to a network socket or a component which is capable of I/O operations such as read, write, connect, and bind

Netty core concepts



io.netty.channel.ChannelPipeline

A list of ChannelHandlers which handles or intercepts inbound events and outbound operations of a Channel

Netty core concepts

io.netty.channel.ChannelHandler

Handles an I/O event or intercepts an I/O operation, and forwards it to its next handler in its ChannelPipeline

Netty core concepts



io.netty.buffer.ByteBuf

A random and sequential accessible sequence of zero or more bytes

High performance buffer pool

*Netty 4 introduces a high-performance buffer pool
which is a variant of jemalloc*

<https://netty.io/wiki/new-and-noteworthy-in-4.0.html>

High performance buffer pool

-  *Reduced GC pressure incurred by frequent allocation and deallocation of the buffers*
-  *Reduced memory bandwidth consumption incurred when creating a new buffer which inevitably has to be filled with zeroes*
-  *Timely deallocation of direct buffers*

Netty uses native code

```
#include <jni.h>

// JNI initialization hooks.

jint netty_unix_register(JNIEnv* env, const char* packagePrefix);
void netty_unix_unregister(JNIEnv* env, const char* packagePrefix);
```

Platform specific libraries

The screenshot shows a GitHub repository page for 'Graylog2 / graylog2-server'. At the top, there are navigation icons for issues, pull requests, and code. Below these are statistics: 1.6k issues, 109 pull requests, and 1.6k code commits. A prominent yellow box highlights the text 'add Netty arm64 dependencies'. At the bottom, a purple button indicates a merge by 'bernd' from a branch into 'Graylog2:master'.

Graylog2 / graylog2-server

<> Code ⚡ Issues 1.6k Pull requests 109 Actions

add Netty arm64 dependencies

Merged bernd merged 6 commits into Graylog2:master from

Example: linux-aarch_64

```
<dependency>
  <groupId>io.netty</groupId>
  <artifactId>netty-transport-native-epoll</artifactId>
  <classifier>linux-aarch_64</classifier>
</dependency>
```

Netty transports

Core



JDK NIO

Native transports



epoll



io_uring



kqueue

Netty optional libraries

Native transports

- epoll transport
- io_uring transport
- kqueue transport

Other

- tcnative boringssl
- brotli4j
- zstd-jni



io_uring support in Netflix Zuul

support Netty io_uring transport #1146

Merged

argha-c merged 27 commits into `Netflix:master` from `sullis:netty-`



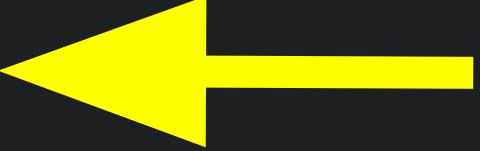
<https://github.com/Netflix/zuul/pull/1146>



Socket options

DefaultSocketChannelConfig.java

```
// Enable TCP_NODELAY by default if possible.  
if (PlatformDependent.canEnableTcpNoDelayByDefault()) {  
    try {  
        setTcpNoDelay(true);  
    } catch (Exception e) {  
        // Ignore.  
    }  
}
```



TCP_NODELAY

TCP_NODELAY

Marc's Blog

It's always TCP_NODELAY. Every damn time.

<https://brooker.co.za/blog/2024/05/09/nagle.html>

TCP_NODELAY

*The first thing I check when debugging latency issues in distributed systems is whether **TCP_NODELAY** is enabled*

TCP_NODELAY

*if you're building a latency-sensitive distributed system running on modern datacenter-class hardware, enable **TCP_NODELAY** (disable Nagle's algorithm) without worries*

Netflix Zuul socket options

The screenshot shows a GitHub repository page for `Netflix / zuul`. The main navigation bar includes links for Code, Issues (245), Pull requests (34, highlighted with an orange underline), Actions, and a grid icon. A prominent purple button at the bottom left indicates the pull request has been Merged. The title of the pull request is "Properly apply socket options #1638". The commit message below it states: "jguerra merged 1 commit into master from jg/options" followed by a merge icon.

<https://github.com/Netflix/zuul/pull/1638>

Netflix Zuul socket options

```
serverBootstrap.option(ChannelOption.SO_BACKLOG, 128);
serverBootstrap.childOption(ChannelOption.SO_LINGER, -1);
serverBootstrap.childOption(ChannelOption.TCP_NODELAY, true);
serverBootstrap.childOption(ChannelOption.SO_KEEPALIVE, true);
```

<https://github.com/Netflix/zuul/pull/1638>



HTTP content compression

Netty compression support

 gzip

 snappy

 brotli

 zstd

HttpContentCompressor

```
import io.netty.handler.codec.http.HttpContentCompressor;
```

```
pipeline.addLast(new HttpContentCompressor((CompressionOptions[]) null));
```

Netty HTTP compression in Zuul

```
@Test  
void gzipOnly()
```

```
@Test  
void brotliOnly()
```



Netty pitfalls





Cookie class



io.netty.handler.codec.http.Cookie



io.netty.handler.codec.http.cookie.Cookie



HttpContentCompressor



```
new HttpContentCompressor()
```



```
new HttpContentCompressor((CompressorOptions[]) null)
```



Netty and Java 21

Aborted (core dumped)

Netty 4.1.98 + Epoll transport + Linux



Netty and Java 21



Norman Maurer

@normanmaurer

...

We released [#netty 4.1.99.Final](#) today as an emergency release. A bug in netty could crash the JVM when using JDK21+.

As JDK21 was just released and people might start upgrading to it we consider upgrading netty critical. See netty.io/news/2023/09/2...

5:34 AM · Sep 28, 2023 · 6,138 Views

<https://netty.io/news/index.html>



Netty resource leaks



Netty project

Reference counted objects

<https://netty.io/wiki/reference-counted-objects.html>

Since Netty version 4, the life cycle of certain objects are managed by their reference counts, so that Netty can return them (or their shared resources) to an object pool (or an object allocator) as soon as it is not used anymore.

<https://netty.io/wiki/reference-counted-objects.html>



“The disadvantage of reference counting is that it is easy to leak the reference-counted objects”

source: Netty wiki



Resource leaks are
difficult to debug



Sept 28-30, 2017
thestrangeloop.com

Direct Memory Leaks

Servers slowly use more memory
Eventually RPS drops to zero
Extremely difficult to debug



Netflix Zuul 2.x

September 2017

Netty resource leak detector



Leak Detector levels:

- DISABLED
- SIMPLE
- ADVANCED
- PARANOID

Best practices to avoid leaks

*“Run your unit tests and integration tests at
PARANOID leak detection level”*

<https://netty.io/wiki/reference-counted-objects.html>



can we implement a
JUnit 5 Extension?



nettyplus / **netty-leak-detector-junit-extension**

netty-leak-detector-junit-extension

Public

Netty leak detection ★

<https://github.com/nettyplus/netty-leak-detector-junit-extension>



nettyplus / **netty-leak-detector-junit-extension**



NettyLeakDetectorExtension.java



NettyLeakListener.java

JUnit 5 example

```
<dependency>
    <groupId>io.github.nettyplus</groupId>
    <artifactId>netty-leak-detector-junit-extension</artifactId>
    <version>0.0.6</version>
</dependency>
```

pom.xml

JUnit 5 example

```
import io.github.nettyplus.leakdetector.junit.NettyLeakDetectorExtension;  
import org.junit.jupiter.api.Test;  
import org.junit.jupiter.api.extension.ExtendWith;  
  
@ExtendWith(NettyLeakDetectorExtension.class)   
class FooTest {  
    @Test  
    void testSomething() {  
        // ...  
    }  
}
```



NettyLeakDetectorExtension

```
leakListener = new NettyLeakListener();
ByteBufUtil.setLeakListener(leakListener);
```



AsyncHttpClient / **async-http-client**

use Netty leak detector extension #1932

Merged

hyperxpro merged 1 commit into `AsyncHttpClient:main` from `sullis:n`



<https://github.com/AsyncHttpClient/async-http-client/pull/1932>



reactor / reactor-netty

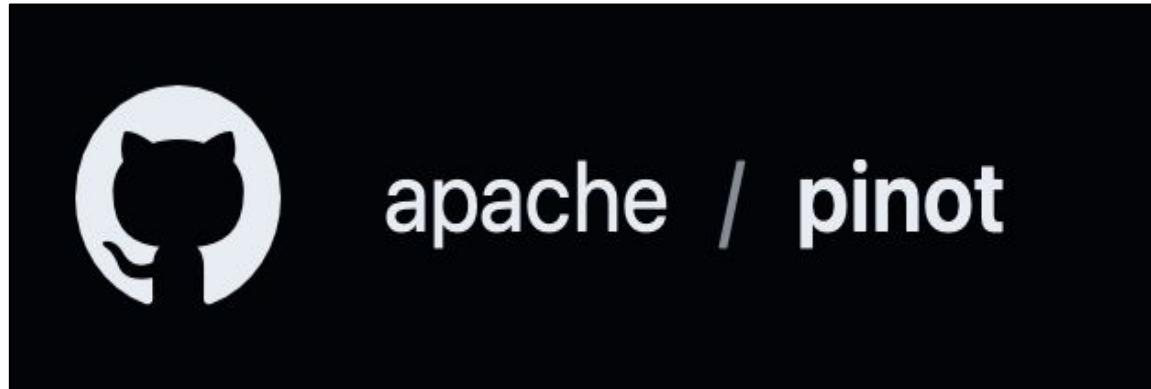
enable Netty leak detector extension

Draft

sullis wants to merge 1 commit into `reactor:main` from `sullis:No`

<https://github.com/reactor/reactor-netty/pull/3064>

Let's talk about Apache Pinot





Pinot uses Netty

```
<dependency>
    <groupId>io.netty</groupId>
    <artifactId>netty-bom</artifactId>
    <version>${netty.version}</version>
    <type>pom</type>
    <scope>import</scope>
</dependency>
```



Pinot: QueryServer.java

```
import io.netty.bootstrap.ServerBootstrap;
```

```
ServerBootstrap serverBootstrap = new ServerBootstrap();
```



Pinot leak detection

enable Netty leak detection #12483

 Merged

Jackie-Jiang merged 2 commits into apache:master from sullis:sean/testng-ne



<https://github.com/apache/pinot/pull/12483>



Native transports

enable Netty native transports by default

⋮ Draft

sullis wants to merge 1 commit into apache:master from sullis:sean/n

<https://github.com/apache/pinot/pull/12577>



Let's talk
about security

October 2023



CVE-2023-44487

“HTTP 2 Rapid Reset”

October 2023

HTTP/2 Zero-Day vulnerability results in record-breaking DDoS attacks

10/10/2023



HTTP/2 Rapid Reset

“Ensure relevant web server and operating system patches are deployed across all Internet-facing Web Servers”





Netty and HTTP/2 Rapid Reset



Norman Maurer
@normanmaurer

...

#netty 4.1.100.Final was released to fix the "HTTP/2 Rapid Reset Attack (CVE-2023-44487)" . If you use #netty to for a #http2 server we urge you to upgrade as soon as possible! netty.io/news/2023/10/1... #cve

9:08 AM · Oct 10, 2023 · 7,153 Views

October 10, 2023



Spring Boot and Netty

Spring Boot web servers

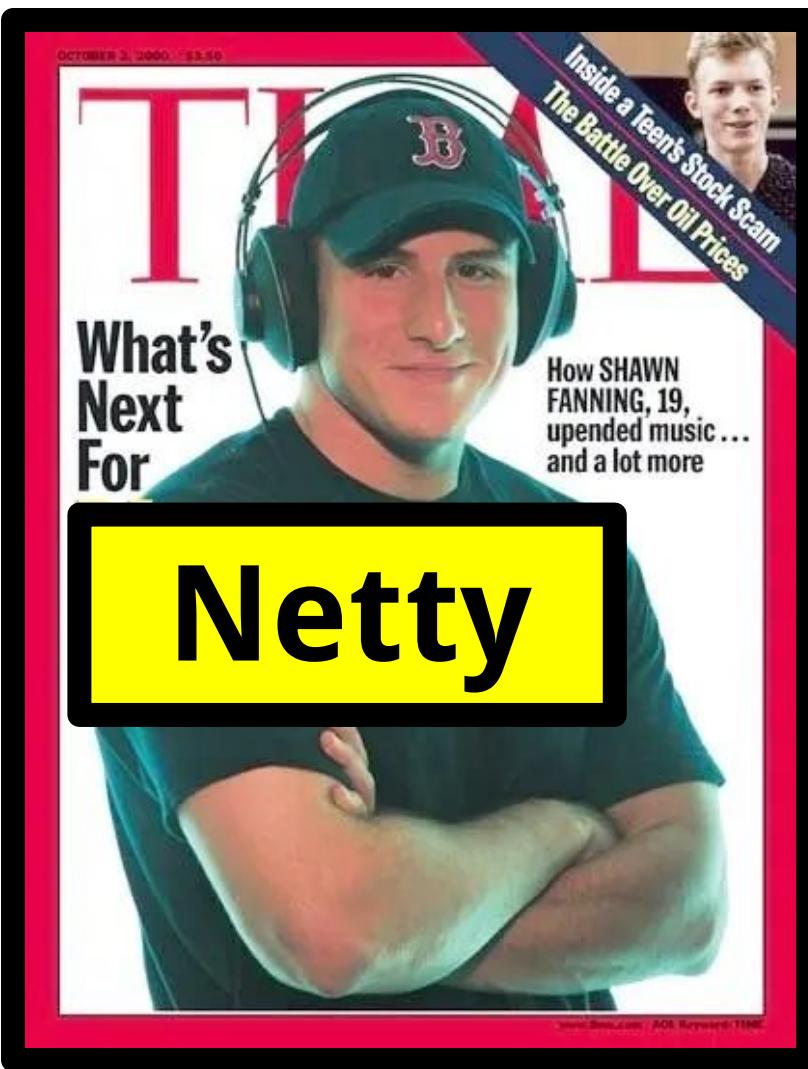
- ◆ JettyWebServer
- ◆ NettyWebServer
- ◆ UndertowWebServer
- ◆ TomcatWebServer

Spring Boot with Netty

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



what's next for Netty?



● Netty 4.2

● Netty 5.0

Netty 4.2 branch

4.2 ▾



This branch is 28 commits ahead of, 7 commits behind 4.1 .

Netty 4.2 and io_uring



Norman Maurer
@normanmaurer

...

Happy Friday! Our `#io_uring` based transport is on the way to get merged in the 4.2 branch of `#netty` and will be part of `#netty` 4.2.0. With the new design of io registration APIs in `#netty` 4.2 it should be trivial to also support `#file` io with `#io_uring`

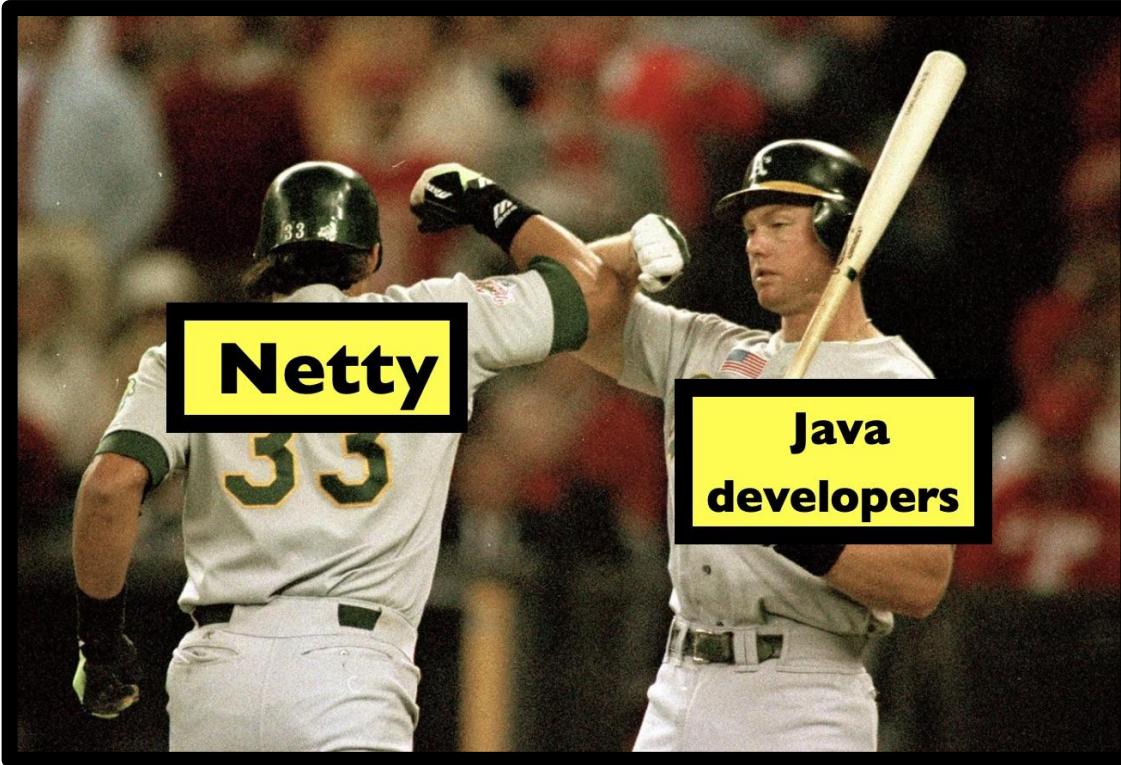
Final thoughts

- ✓ Netty is everywhere
- ✓ Netty is extensible
- ✓ Netty is awesome

Questions?



The End

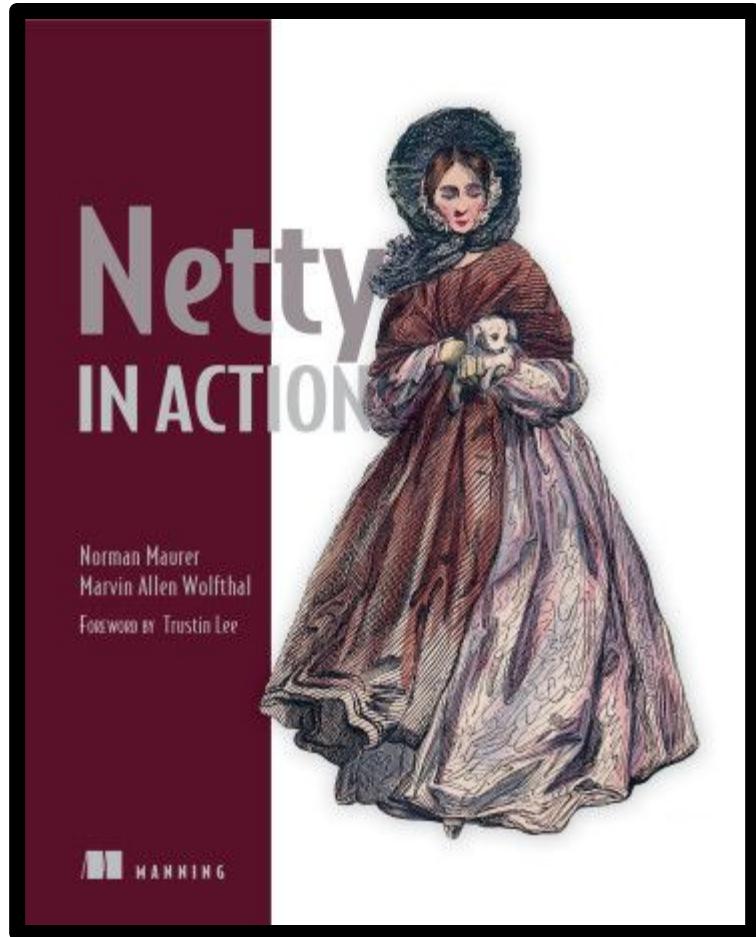




Bonus content



Learning Netty



published December 2015



covers Netty 4.x

The image is a composite of two parts. On the left, a man stands behind a podium at a conference. The podium has a sign that reads "DEVOXX BELGIUM". Behind him is a large screen displaying the text "The IO framework that propels them all" and the name "Stéphane Landelle @ DEVOXX". On the right, there is a slide with a dark blue background featuring a network graph. At the top, it says "Gatling" with its logo. Below that, the word "Netty" is written in orange. Underneath "Netty", the same text "The IO framework that propels them all" is displayed. At the bottom of the slide, the name "Stéphane Landelle @ DEVOXX" appears again. The overall background of the image is a colorful, abstract scene of what looks like a microscopic view of cells or a planetary surface.

The IO framework
propels them all

Gatling

Netty

The IO framework that
propels them all

Stéphane Landelle @ DEVOXX

ING Google Cloud aws JET BRAINS ORACLE DEVOXX™

Netty, the IO framework that propels them all By Stephane LANDELLE

Stéphane Landelle @ Devoxx Belgium 2023



Need for speed: Boosting Apache Cassandra's performance using Netty

Dinesh A. Joshi
✉@dineshjoshi
dinesh.joshi@gatech.edu



**Devoxxua 2018: Need for speed: Boosting Apache Cassandra's performance using Netty
(Dinesh Joshi)**

Dinesh Joshi @ Devoxx Ukraine 2018



DEVOXX™

Netty - One Framework to rule them all by Norman Maurer

Norman Maurer @ Devoxx Belgium 2016