Reducing the Size of Your Java Run-Time Image

FOSDEM 2025

Severin Gehwolf

Principal Software Engineer

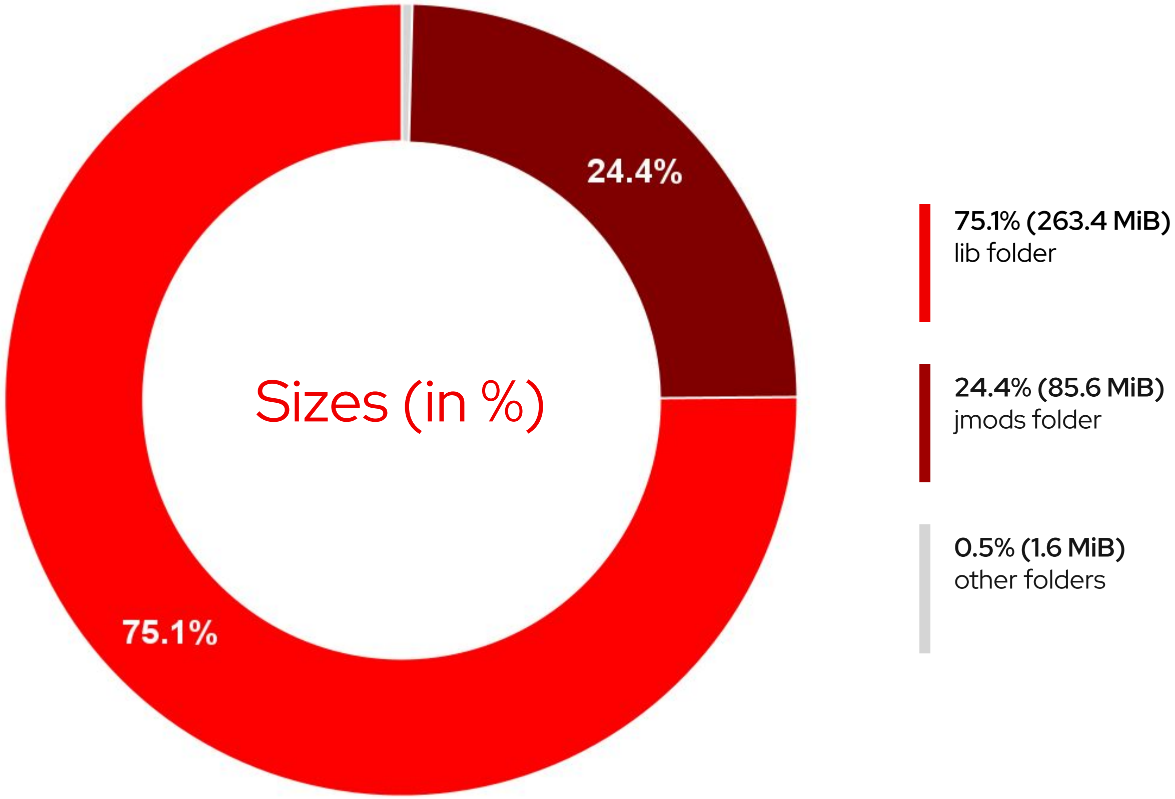
Looking back: Size of the JDK

… or what JEP 493 changed

JDK 23: File sizes

Top level folder of the extracted JDK archive

JDK 23.0.1*



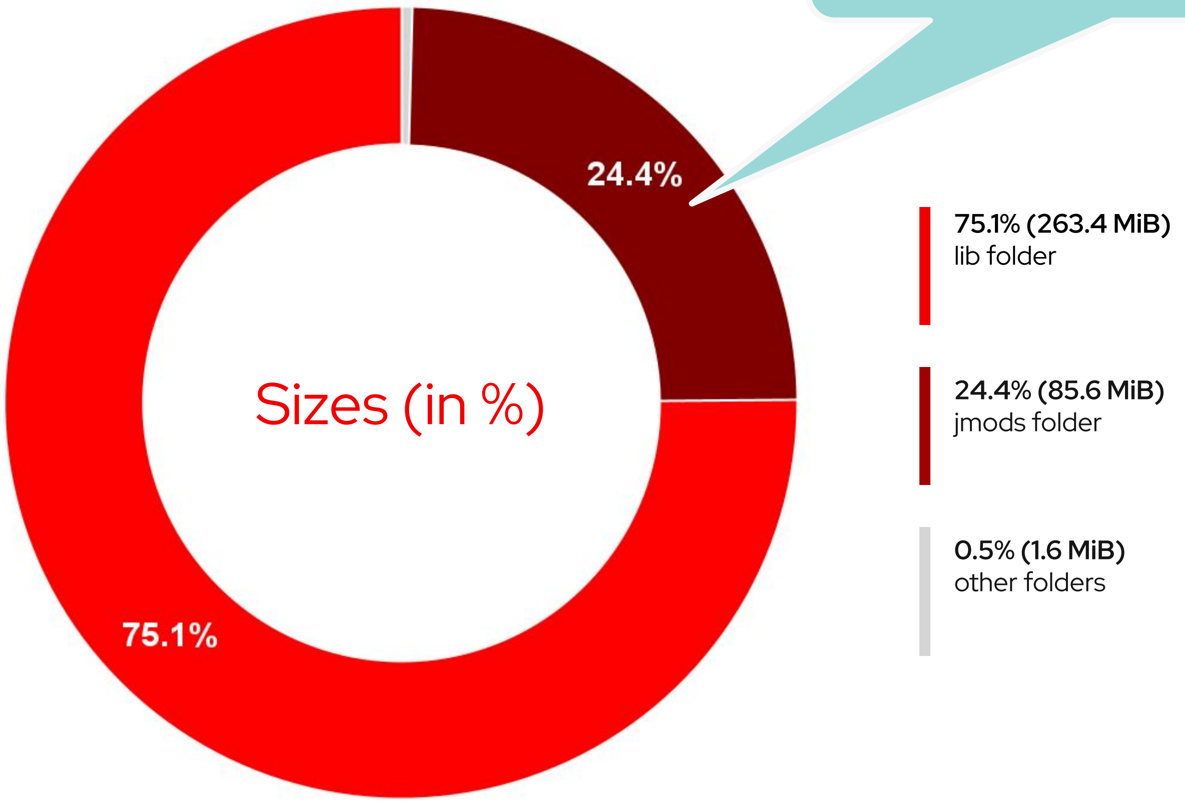
Source:
*) JDK 23.0.1: https://api.adoptium.net/v3/binary/version/jdk-23.0.1%2B11/linux/x64/jdk/hotspot/normal/adoptium Analyzed on Linux. Other platforms look similar.

JDK 23: File sizes

JDK 23.0.1*

JEP 493 removes this

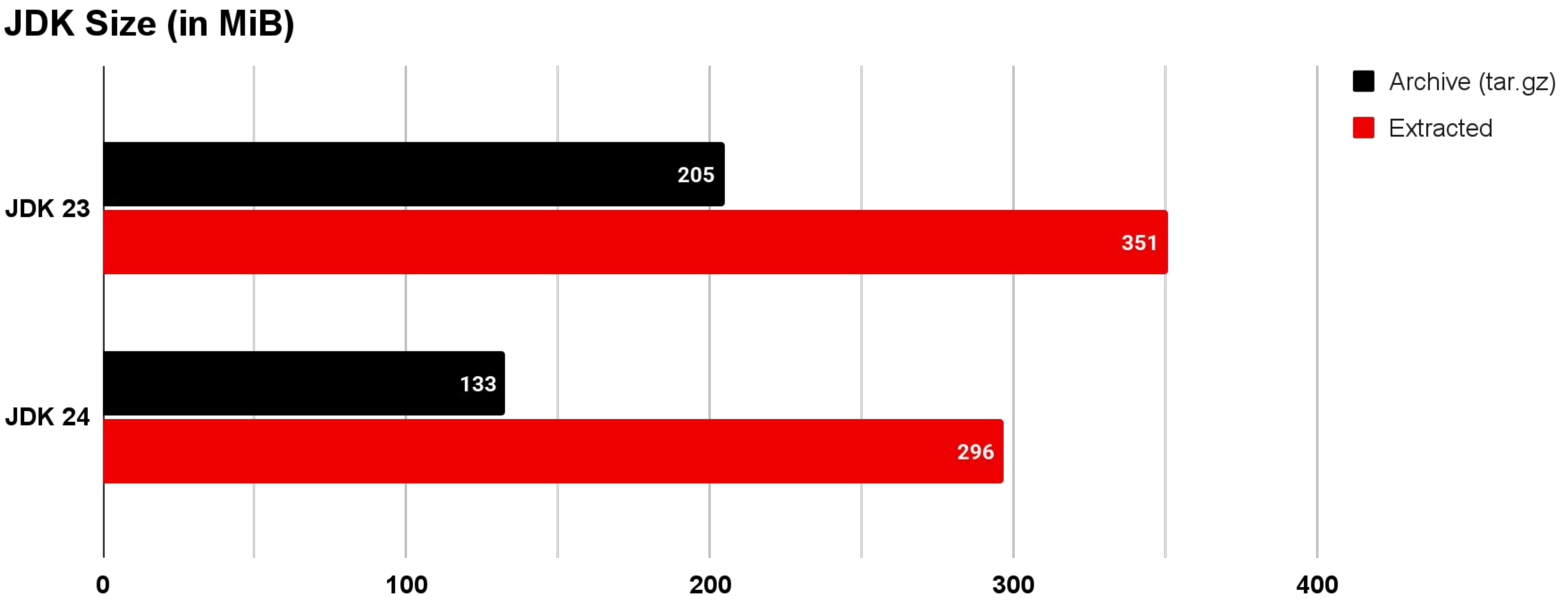
Top level folder of the extracted JDK archive



Source:
*) JDK 23.0.1: https://api.adoptium.net/v3/binary/version/jdk-23.0.1%2B11/linux/x64/jdk/hotspot/normal/adoptium Analyzed on Linux. Other platforms look similar.

On-disk size comparison of JDKs

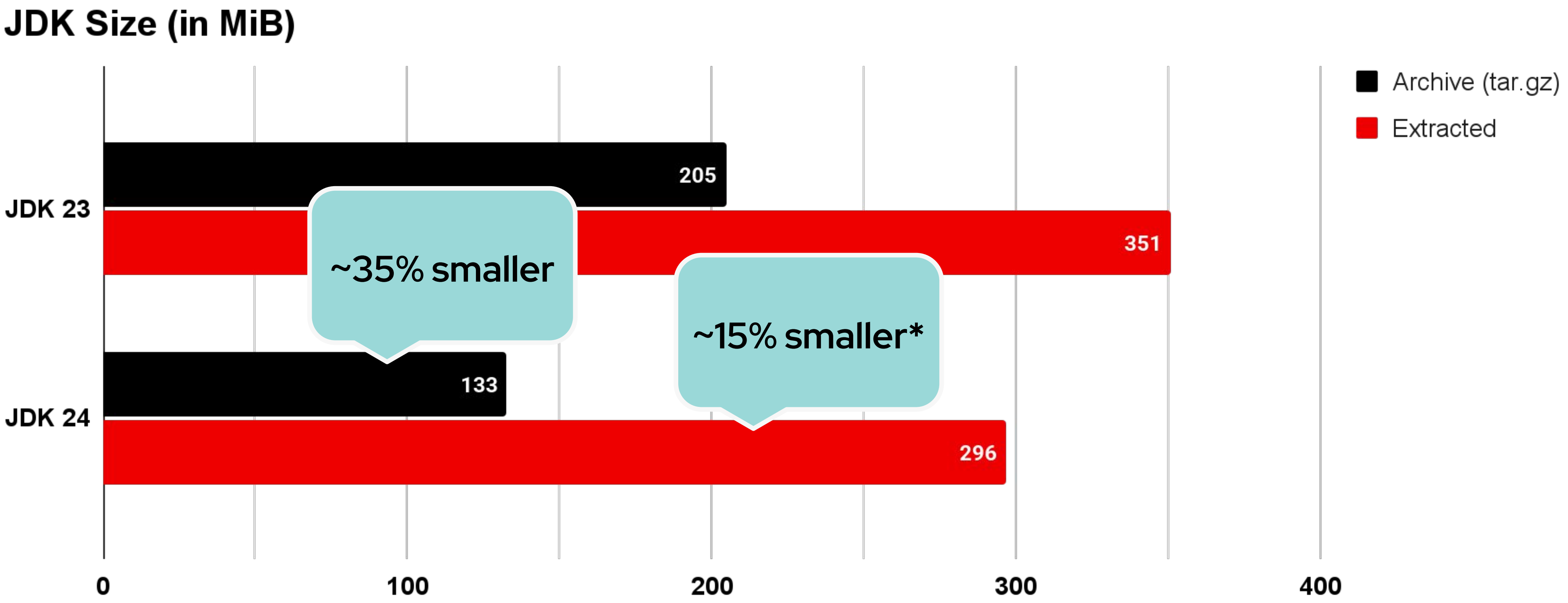
JDK 23.0.1 vs JDK 24+31 (ea) with JEP 493 enabled



Source:
Eclipse Temurin JDK 23.0.1 and Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
JDK 23: https://api.adoptium.net/v3/binary/version/jdk-23.0.1%2B11/linux/x64/jdk/hotspot/normal/adoptium

On-disk size comparison of JDKs

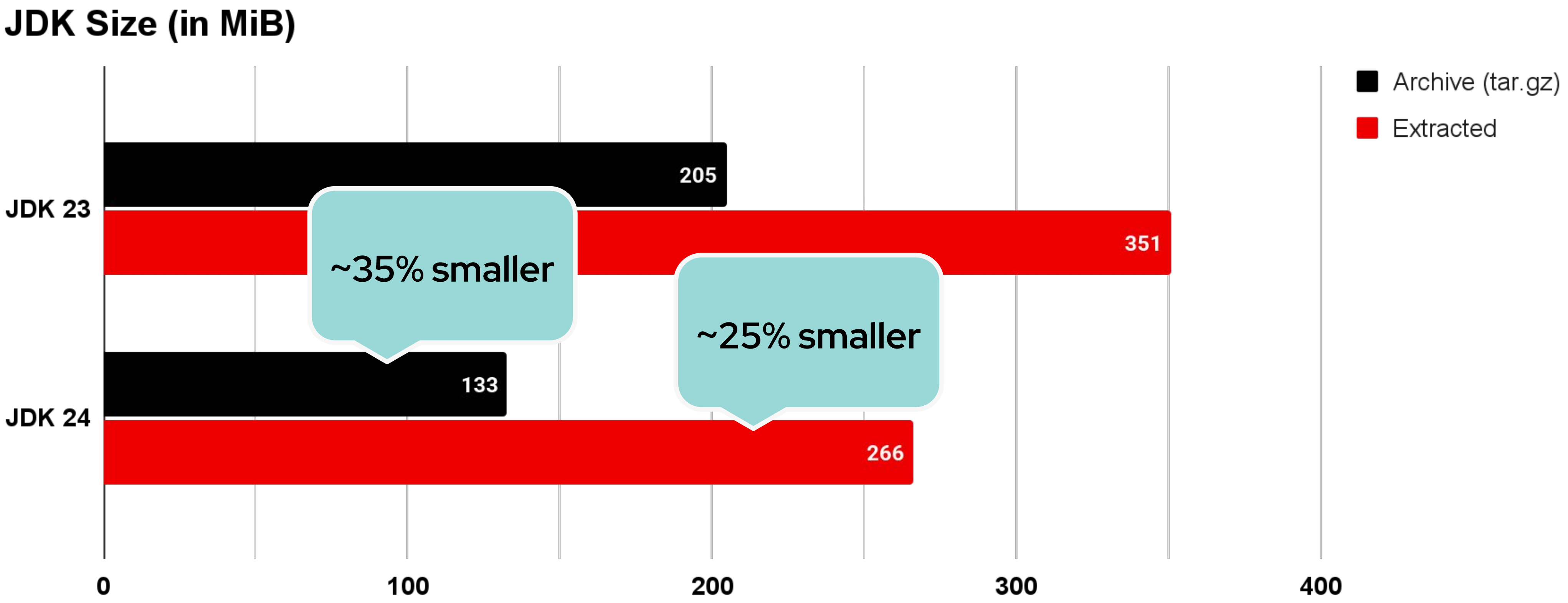
JDK 23.0.1 vs JDK 24+31 with JEP 493 enabled



Source:
Eclipse Temurin JDK 23.0.1 and Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
JDK 23: https://api.adoptium.net/v3/binary/version/jdk-23.0.1%2B11/linux/x64/jdk/hotspot/normal/adoptium
*) Actually 25% smaller when comparing same files as in JDK 23 outside JMODs, but JDK 24 with JEP 450 also includes *_coh.jsa shared archives amounting to ~30MB

On-disk size comparison of JDKs

JDK 23.0.1 vs JDK 24+31 (JEP 450 clean**)



Source:
Eclipse Temurin JDK 23.0.1 and Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
JDK 23: https://api.adoptium.net/v3/binary/version/jdk-23.0.1%2B11/linux/x64/jdk/hotspot/normal/adoptium
**) JEP 450 - new in JDK 24 - includes two new *_coh.jsa files in lib/server which in total amount to ~30MB. Removing those files for a better comparison to JDK 23, we arrive at the JEP 493 advertised number

Custom Run-Time Images

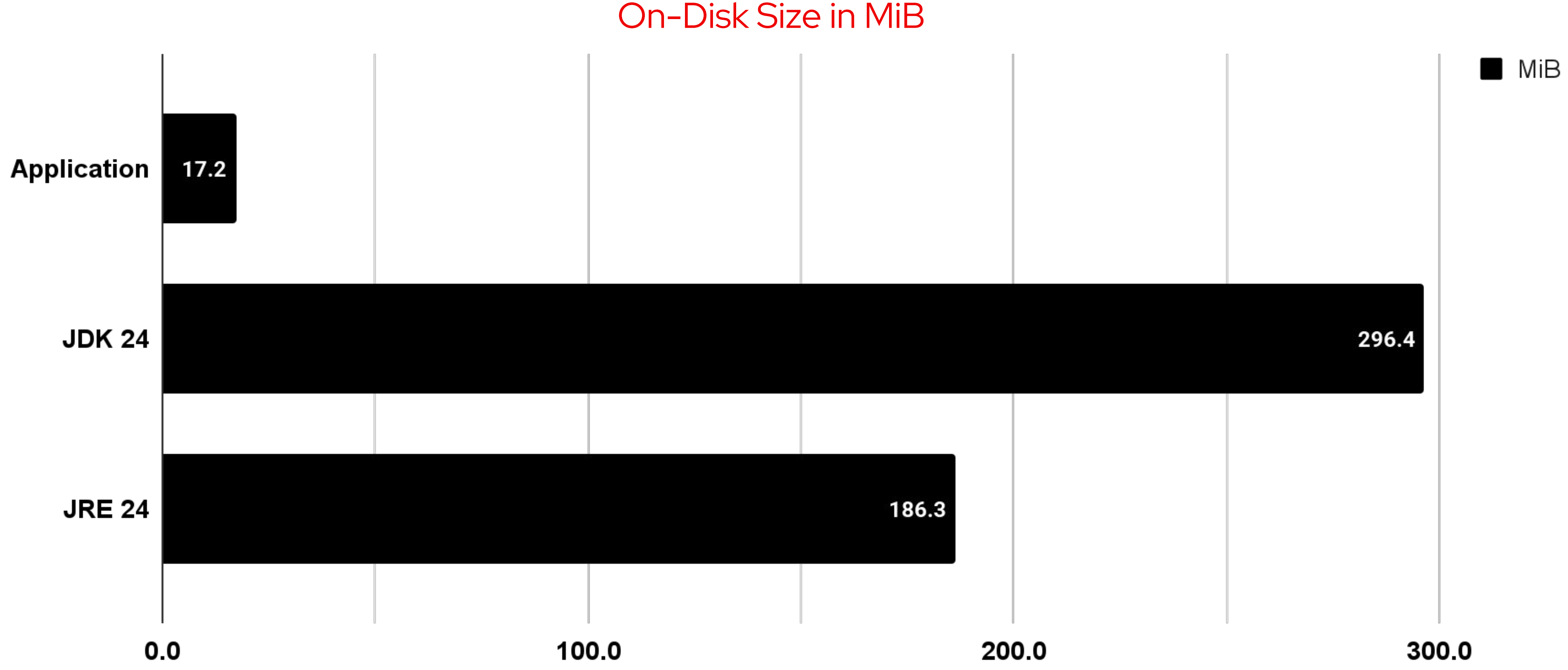
Consider the following scenario:

- REST application*) running on classpath
- Application requires JDK modules **java.logging, java.naming, jdk.unsupported**
- Desire to bundle application with a suitable Java runtime

Source:
*) Quarkus sample application from: https://github.com/quarkusio/quarkus-quickstarts/tree/main/rest-json-quickstart, Guide at: https://quarkus.io/guides/rest-json,

Option 1:
Bundle full JDK with Application

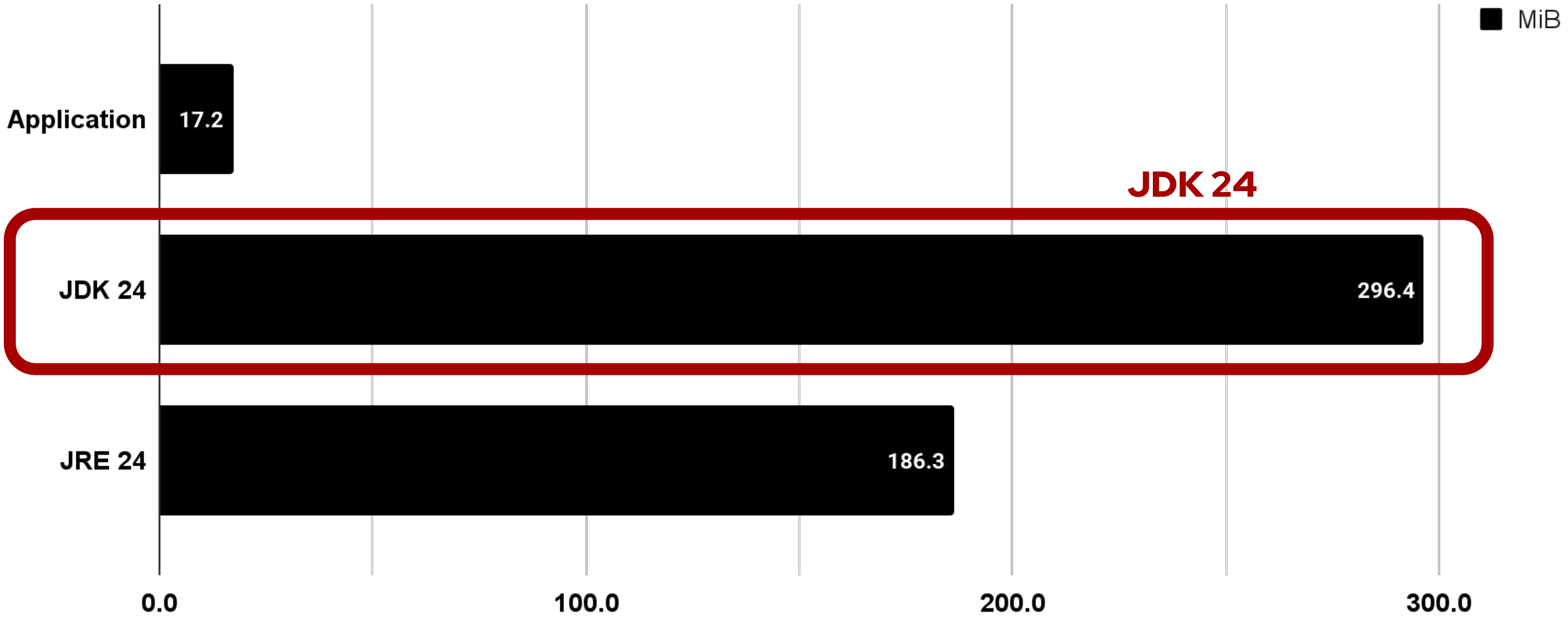
Size characteristics of application and JDK 24



Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
JRE 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jre/hotspot/normal/adoptium

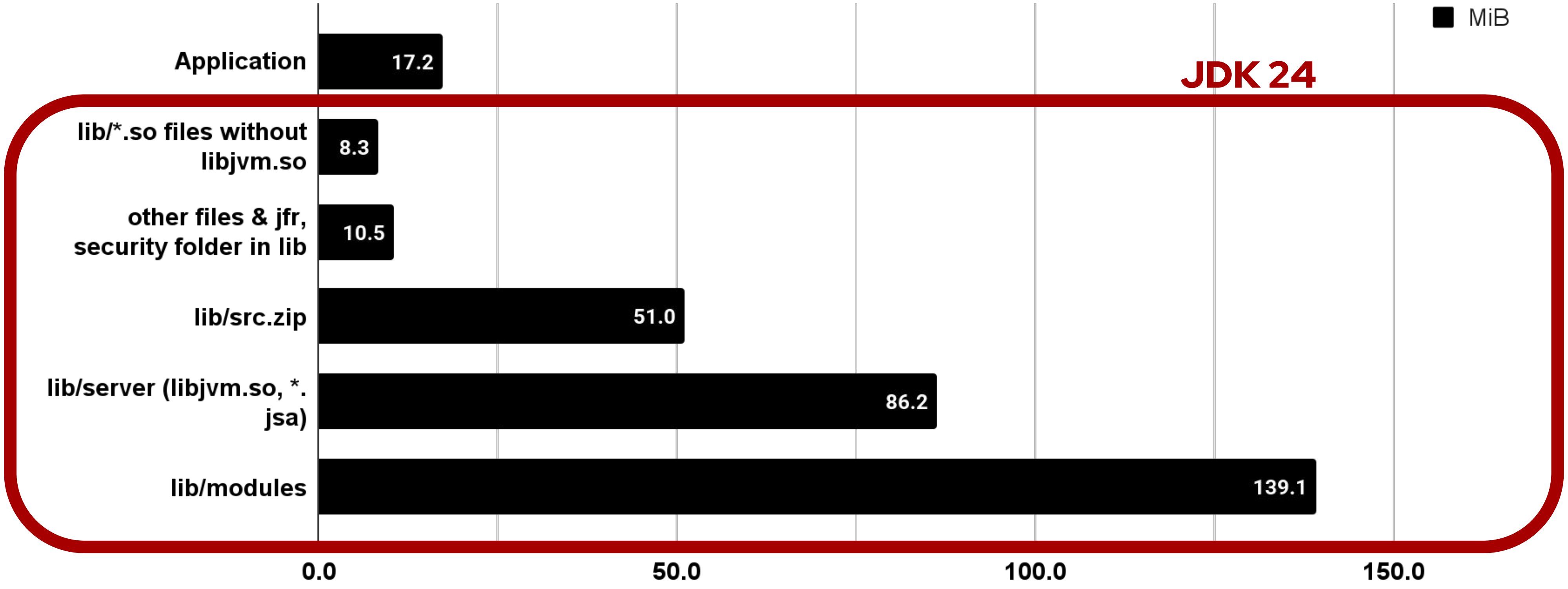
Size characteristics of application and JDK 24

On-Disk Size in MiB



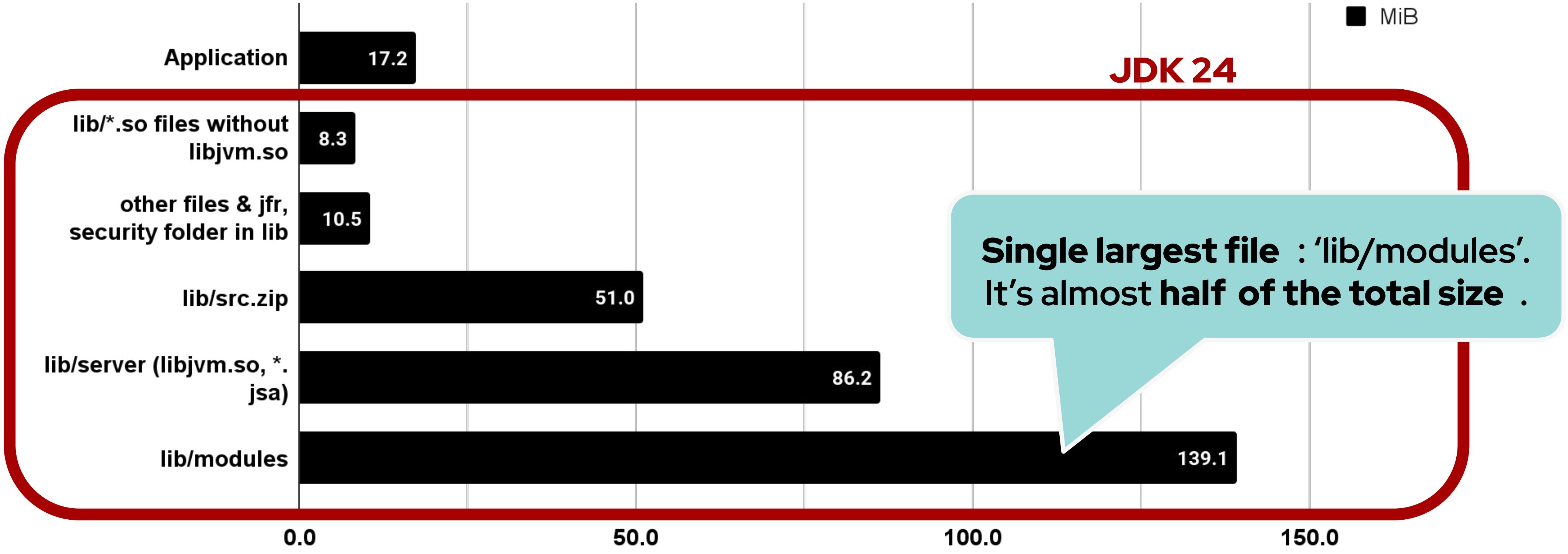
Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
JRE 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jre/hotspot/normal/adoptium

Size characteristics of application and JDK 24 - details



Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium

Size characteristics of application and JDK 24 - details



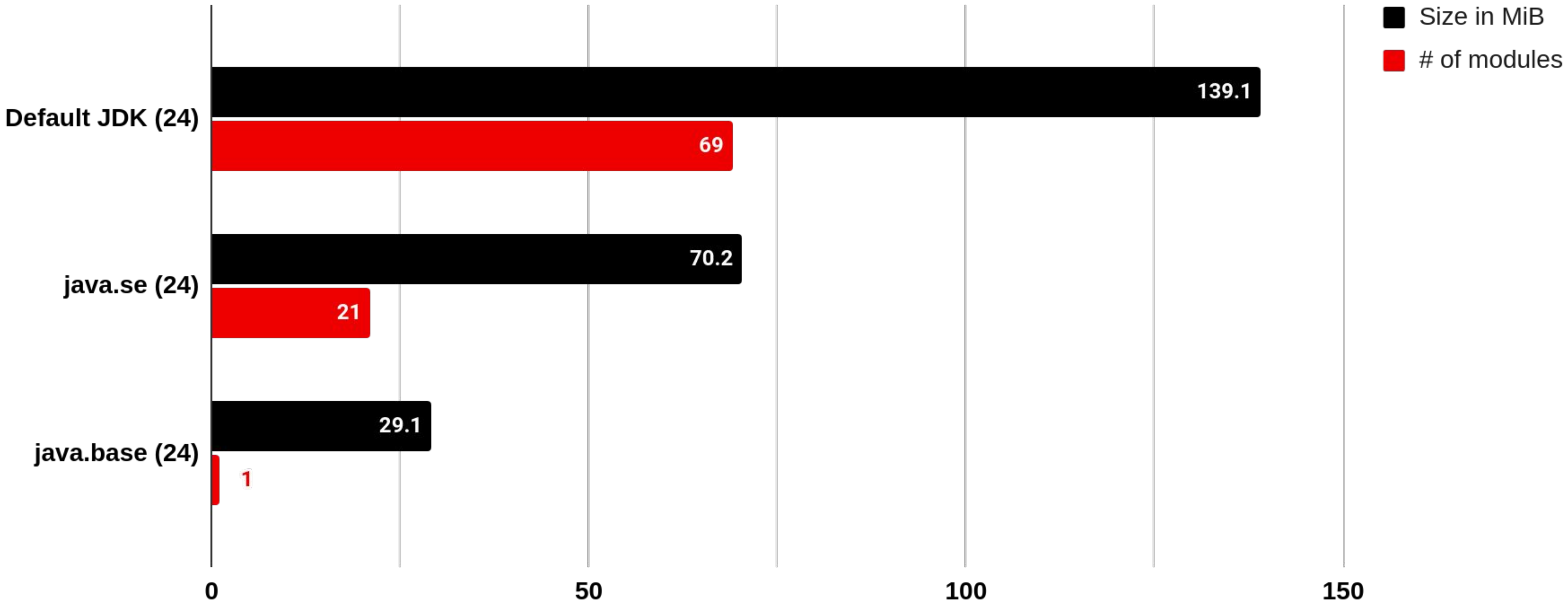
Single largest file : 'lib/modules'. It's almost **half of the total size** .

Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium

Bundle full JDK/JRE?
Probably **NOT**.

Observation: lib/modules file size

Size in MiB vs. number of included modules



Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
Used: java –list-modules | wc –l for the # of modules. Created images with 'jlink –add-modules <label> –output <label>-image`

Option 2: **Custom Runtime** with only **select modules**

Enter jlink

JEP 493 enabled JDK

```
>  API_URL=https://api.adoptium.net/v3/binary/latest/24/ea/linux/x64/jdk/hotspot/normal/eclipse
> curl -sL $API_URL -o jdk.tar.gz
> mkdir jdk
> tar -xf jdk.tar.gz -C jdk --strip-components=1
> jdk/bin/jlink --help | tail -n2
Capabilities:
    Linking from run-time image enabled
```

JEP 493 enabled JDK build

Creating custom run-time images

with JEP 493

```
> jdk/bin/jlink \
    --add-modules java.logging,java.naming,jdk.unsupported \
    --add-modules jdk.crypto.cryptoki,jdk.jdwp.agent,jdk.jfr \
    --output runtime-for-rest-app

> runtime-for-rest-app/bin/java --list-modules
java.base@24-beta
java.logging@24-beta
java.naming@24-beta
java.security.sasl@24-beta
jdk.crypto.cryptoki@24-beta
jdk.jdwp.agent@24-beta
jdk.jfr@24-beta
jdk.unsupported@24-beta
```

Source:
Eclipse Temurin JDK Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium

Creating custom run-time images

with JEP

Modules required by application

```
> jdk/bin/jlink \
    --add-modules java.logging,java.naming,jdk.unsupported \
    --add-modules jdk.crypto.cryptoki,jdk.jdwp.agent,jdk.jfr \
    --output runtime-for-rest-app

> runtime-for-rest-app/bin/java --list-modules
java.base@24-beta
java.logging@24-beta
java.naming@24-beta
java.security.sasl@24-beta
jdk.crypto.cryptoki@24-beta
jdk.jdwp.agent@24-beta
jdk.jfr@24-beta
jdk.unsupported@24-beta
```

Source:
Eclipse Temurin JDK Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium

Creating custom run-time images

with JEP

Modules required by application

```
> jdk/bin/jlink \
    --add-modules java.logging,java.naming,jdk.unsupported \
    --add-modules jdk.crypto.cryptoki,jdk.jdwp.agent,jdk.jfr \
    --output runtime-for-rest-app

> runtime-for-rest-app/bin/java --lis
java.base@24-beta
java.logging@24-beta
java.naming@24-beta
java.security.sasl@24-beta
jdk.crypto.cryptoki@24-beta
jdk.jdwp.agent@24-beta
jdk.jfr@24-beta
jdk.unsupported@24-beta
```

Extra modules:

crypto, monitoring, debugging

Source:
Eclipse Temurin JDK Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium

Creating custom run-time images

with JEP 493

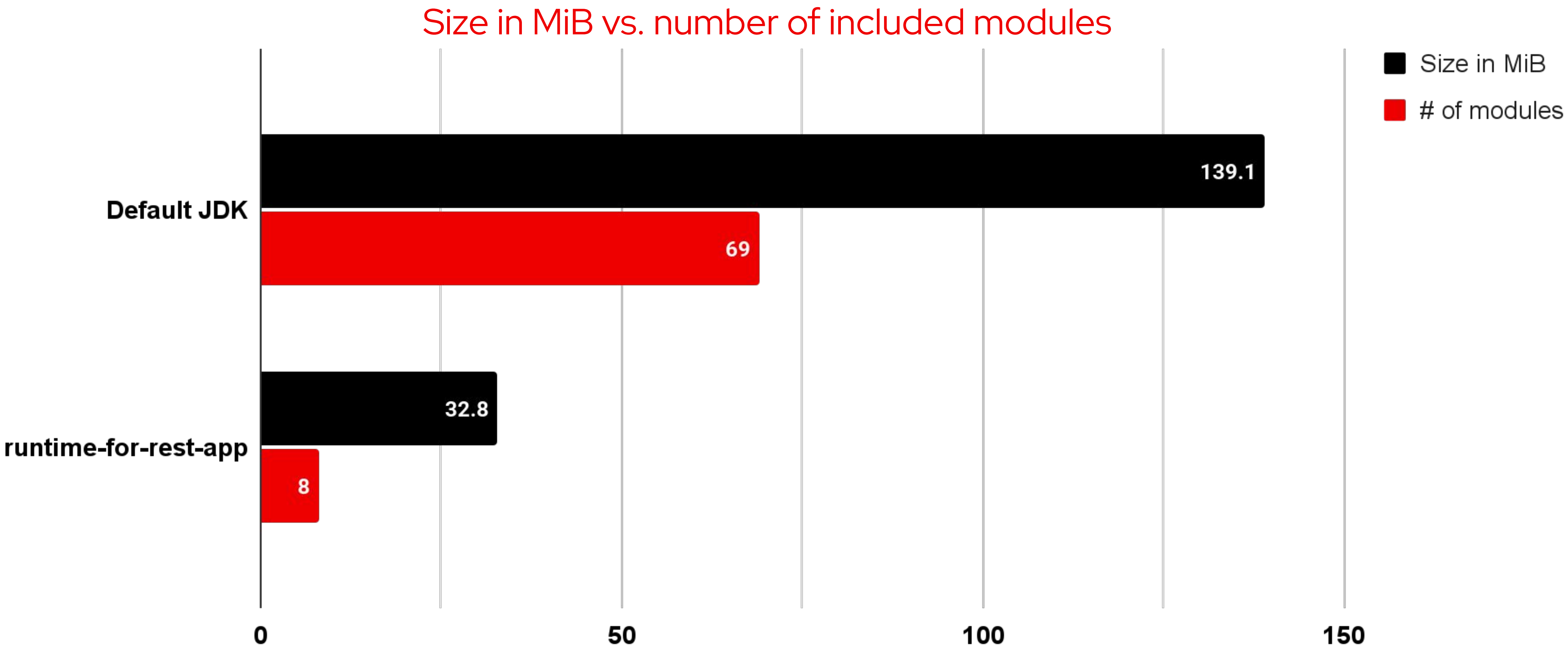
```
> jdk/bin/jlink \
    --add-modules java.logging,java.naming,jdk.unsupported \
    --add-modules jdk.crypto.cryptoki,jdk.jdwp.agent,jdk.jfr \
    --output runtime-for-rest-app

> runtime-for-rest-app/bin/java --list-modules
java.base@24-beta
java.logging@24-beta
java.naming@24-beta
java.security.sasl@24-beta
jdk.crypto.cryptoki@24-beta
jdk.jdwp.agent@24-beta
jdk.jfr@24-beta
jdk.unsupported@24-beta
```

Use `java –list-modules` to list included modules in image

Source:
Eclipse Temurin JDK Eclipse Temurin JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium

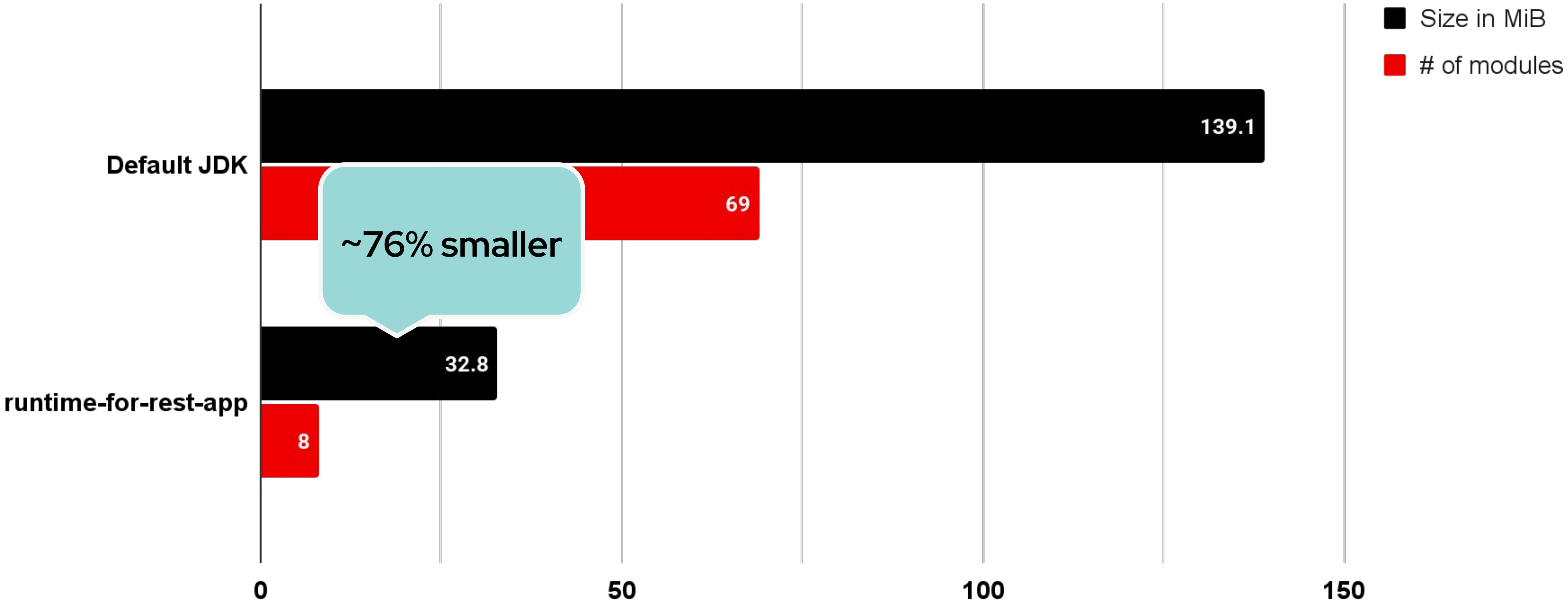
Default JDK vs custom runtime: lib/modules



Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
Used: java –list-modules | wc -l for the # of modules. Default JDK 24 modules image vs. modules image created using commands on slide 29

Default JDK vs custom runtime: lib/modules

Size in MiB vs. number of included modules

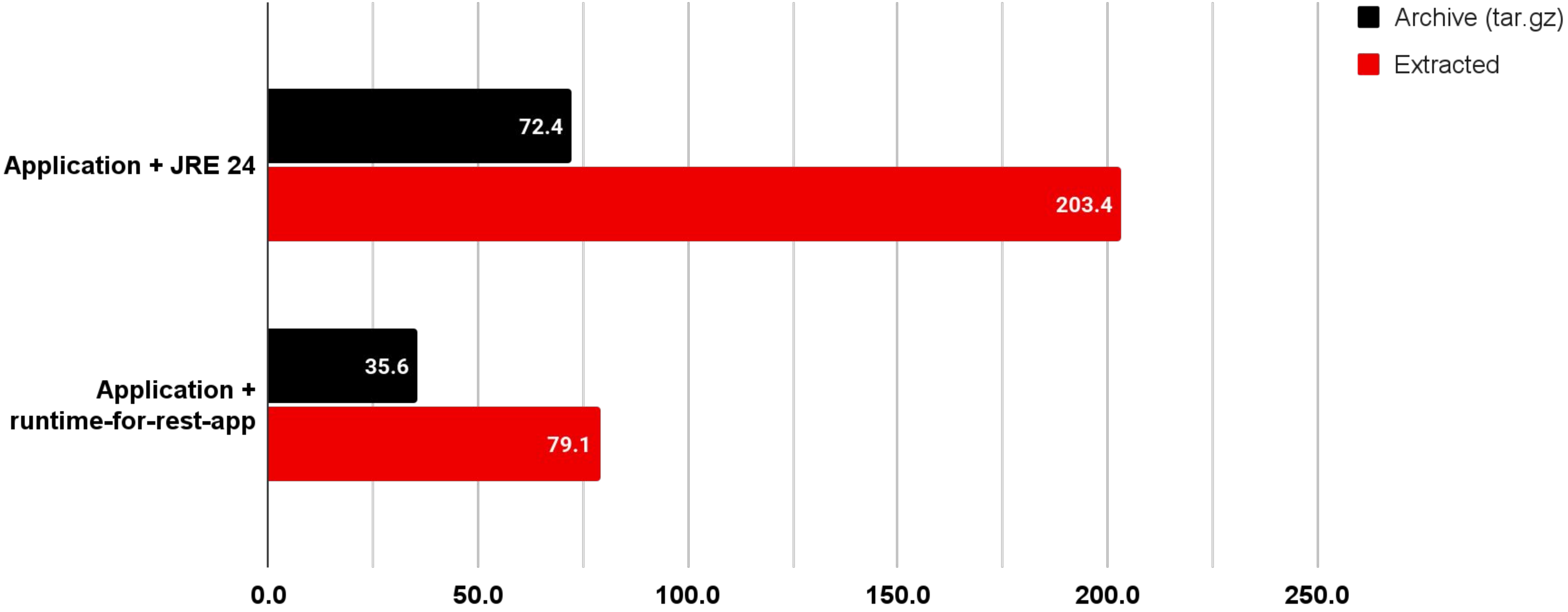


~76% smaller

Source:
Eclipse Temurin JDK JDK 24+31 (ea)
JDK 24: https://api.adoptium.net/v3/binary/version/jdk-24%2B31-ea-beta/linux/x64/jdk/hotspot/normal/adoptium
Used: java –list-modules | wc -l for the # of modules. Default JDK 24 modules image vs. modules image created using commands on slide 29

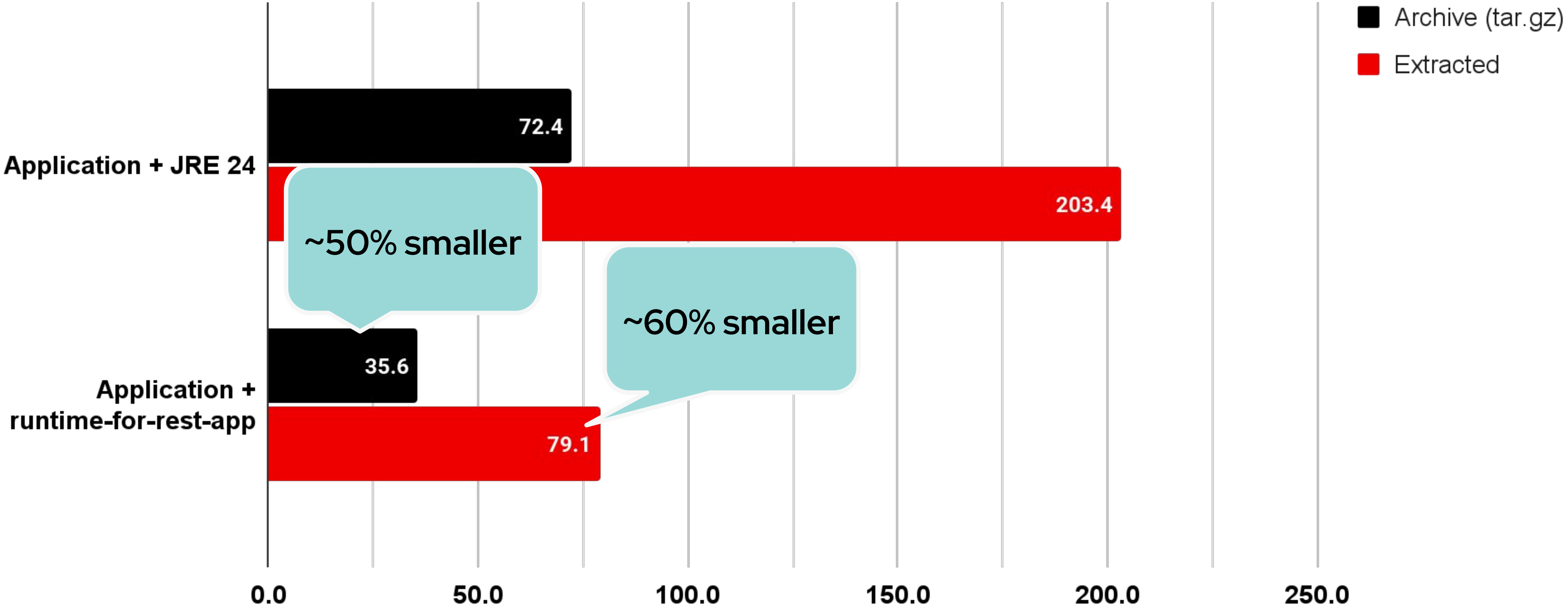
Default JDK vs custom runtime

Size in MiB



Default JDK vs custom runtime

Size in MiB



~50% smaller

~60% smaller

Recap and a look ahead

Key takeaways

Lessons learned:

- **JDK 24 got smaller** *)

  ~35% reduction in archive size

- **Custom Runtime approach achieves similar results** for your application bundle

  ~50% reduction in size to bundled JRE

*) with JEP 493 enabled

Key takeaways

Lessons learned:

- **JDK 24 got smaller** *)

  ~35% reduction in archive size

- **Custom Runtime approach achieves similar results** for your application bundle

  ~50% reduction in size to bundled JRE

What now?

- Using jlink was never easier (no more JMODs)
- Go create a custom runtime for your application today!

*) with JEP 493 enabled

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

x.com/RedHat

Acknowledgement:
Part of this work was performed in the context of the EU funded project AERO with project number 101092850

Funded by
the European Union