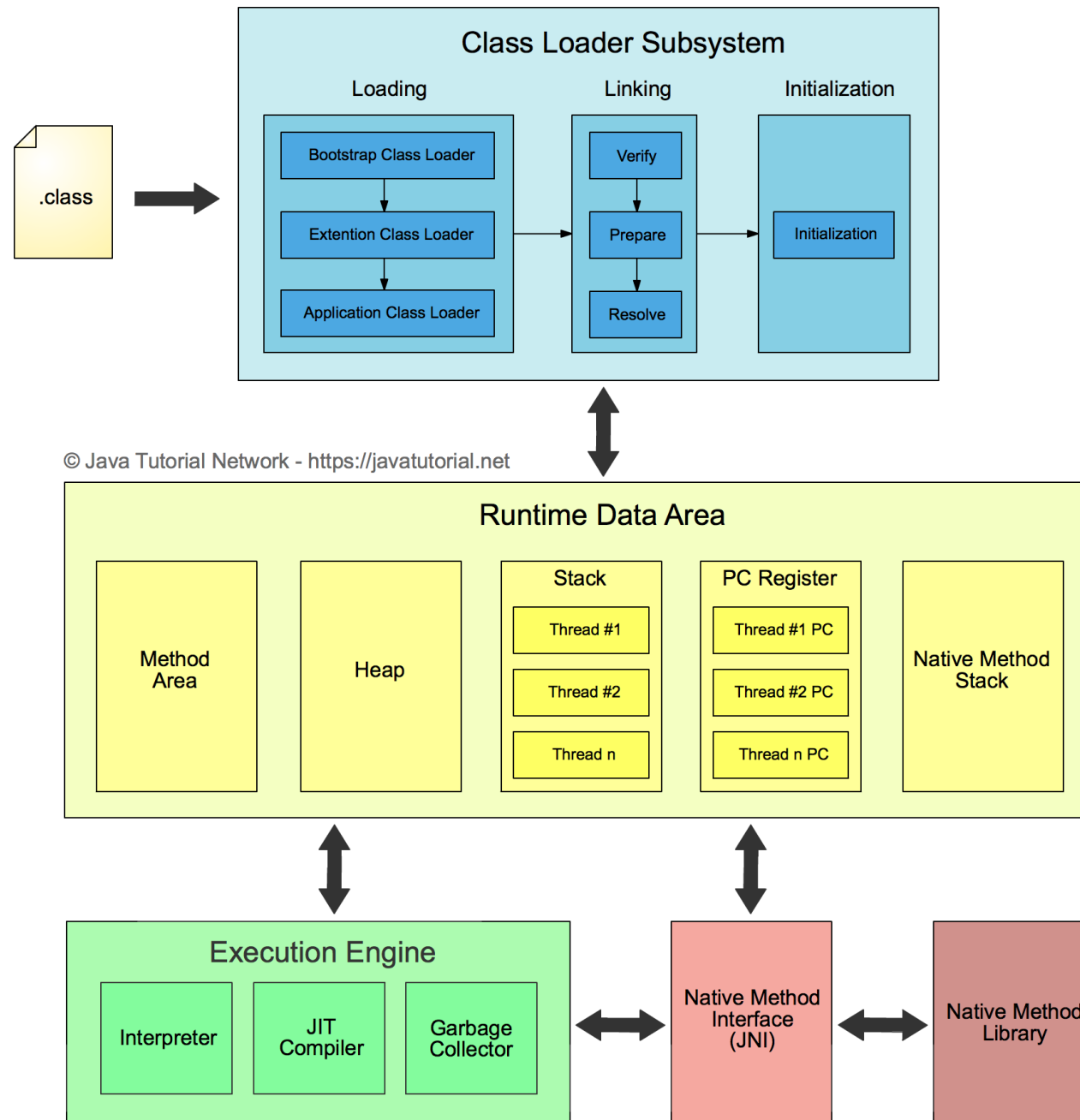


How Class Data Sharing Can Speed up Your Java Application Startup





Rudy De Busscher

- **Payara**

- Technical Sales
- Developer Advocate

- **Involved in**

- Committer of MicroProfile
- Committer in Eclipse EE4J groups
- Java EE Security API Expert group member



@rdebusscher



<https://blog.payara.fish/>
<https://www.atbash.be>

Agenda

- JVM internals related to Class
- Why fast startup important?
- Class Data Sharing
 - Basics
 - Demos
- Conclusion

Using a class

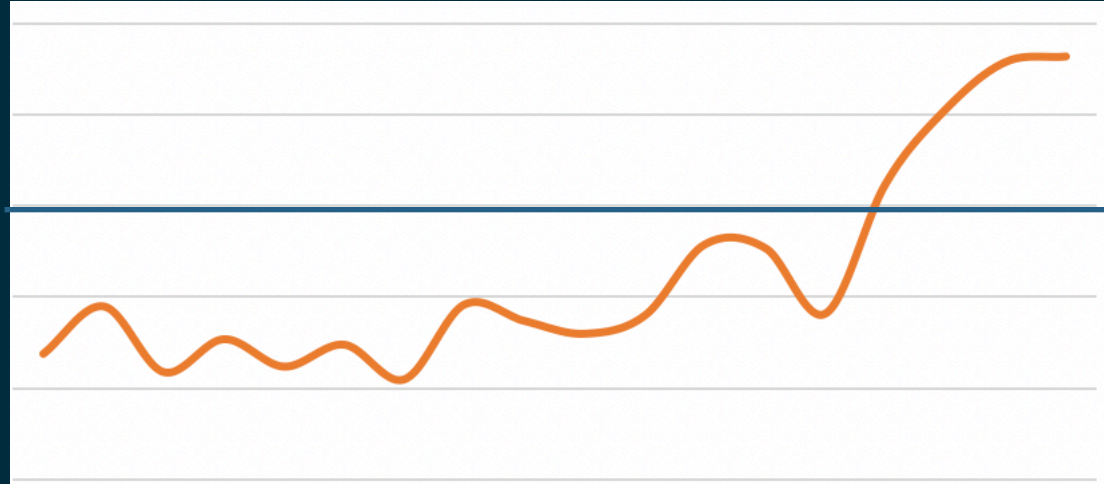
- `new MyClass();`
- Load Java Byte code (through the Classloader)
- Verification of Byte code
- Class file derivation (can result in *IncompatibleClassChangeError*)
- Preparation (~ variable initialisation)
- Resolution (resolve symbolic references)
- Interpreter uses code

Improvements?

- Can we improve this?
 1. —noverify option
 - Opens door for malicious byte code.
 2. Native compilation
 - Slow
 - Lower peak performance
 - Obscurity with reflection
 - No JVTMI, Java Agents, ...
 3. Class Data Sharing - CDS

Why Fast Startup Important?

- Large surge in requests



- React fast
- Avoid refused connection during scale up.
- (Not cold startup problem with serverless)

Class Data Sharing - CDS

- Load, verify, derive
 - Perform upfront
 - Store info in memory mapped read-only file
- *Loading* class is much faster
- Memory requirements are lower (can be shared)
 - Hello world -> 437 classes
 - Payara Micro -> 15.000 classes

Using CDS

- OpenJDK
 - -XX:DumpLoadedClassList=
 - -XX:SharedArchiveFile=
 - -XX:share:dump|on

A photograph of a workspace on a dark wooden desk. On the left is a silver laptop with a black keyboard. To its right is a light blue ceramic mug filled with a brown liquid, likely tea. Below the mug are a pair of white earbuds with black cables. A large white diamond shape is overlaid on the right side of the image, containing the word "Demo" in orange.

Demo

Gain

35%

- OpenJDK

	Startup Time	Memory
Plain	8,2 sec	49,8 Mb
CDS	5,2 sec	49,5 Mb

Docker images

- Several steps required
 - Multi stage Docker file
- Shared Archive file can be in Image
 - Layer is reused
- Shared Archive file can be outside
 - On a volume
 - Extra step in build or use *podman build*

A photograph of a workspace on a dark wooden desk. On the left is a silver laptop with a black keyboard and a trackpad. To its right is a light blue ceramic mug filled with a brown liquid, likely tea. Below the mug are a pair of white earbuds with black cables. A white diagonal shape, resembling a folded piece of paper, is overlaid on the right side of the image.

Demo

“Pitfalls”

- Class path issues
 - Classpath must be identical when using CDS than when CDS is prepared
- Dynamic proxies can't be used
- Pre 1.6 class formatted files can't be used.
- Investigate issues -> *-Xlog:load+class*
- Shared Archive file is version and OS dependent

Payara Reef: Community Growth Program

Let us help you spread the word about our open source software. Join the Reef!

We'll Support You With:

- Event, JUG, conference sponsorship
- Freebies, swag, handouts, speakers
- Promotion and advertising of events and articles
- Community forum

Learn More:

www.payara.fish/reef

Thank You

**Download the open source software:
payara.fish/downloads**

**Get Started:
payara.fish/get-started**