

# **Project Leyden: Premain**

**Case study: Spring Boot PetClinic**

**Vladimir Ivanov, Oracle**

**09/12/2023**

# Premain: Current Status



- JVMLS talk (video): <https://www.youtube.com/watch?v=lnth19Kf-x0>
- Repository: <https://github.com/openjdk/leyden/tree/premain>
- Samples: <https://github.com/openjdk/leyden/tree/premain/test/hotspot/jtreg/premain>
  - PetClinic: <https://github.com/openjdk/leyden/tree/premain/test/hotspot/jtreg/premain/spring-petclinic>

# Training Process

4 steps

1. Dump **class list** for static CDS archive
  - -Xshare:off -XX:DumpLoadedClassList=<file>
2. Create **static CDS archive** based on the class list
  - -Xshare:off -XX:DumpLoadedClassList=<file> -XX:SharedArchiveFile=<file>
3. Dump application training data into **dynamic CDS archive**
  - -XX:SharedArchiveFile=<file> -XX:ArchiveClassesAtExit=<file> -XX:+RecordTraining
4. Dump application compiled code into **cached code archive**
  - -XX:SharedArchiveFile=<file> -XX:+ReplayTraining XX:+StoreCachedCode  
-XX:CachedCodeFile=<file>

# Training Process

3 artifacts:

- **Static CDS archive**
  - contains classes and heap snapshot
- **Dynamic CDS archive**
  - contains training data
- **Cached code archive**
  - pre-compiled machine code (nmethods)

# Deployment Run

-XX:SharedArchiveFile=<**dynamic\_cds\_archive**>

-XX:+ReplayTraining

-XX:+LoadCachedCode -XX:CachedCodeFile=<**code\_file**>

-XX:+UseRecompilation

# Case study: PetClinic

# Case Study: PetClinic

## Baseline

```
$ java -jar spring-petclinic-3.1.0.jar
```

Started PetClinicApplication in 3.472 seconds (process running for 3.768)

# Case Study: PetClinic

## Problem: Custom Loader

```
$ java -verbose:class -jar spring-petclinic-3.1.0.jar
```

```
...
```

```
[0.694s][info][class,load] ....FutureOrPresentValidatorForMonthDay
```

```
source: jar:file:spring-petclinic-3.1.0.jar!/BOOT-INF/lib/hibernate-  
validator-8.0.0.Final.jar/ loader:
```

```
org.springframework.boot.loader.LaunchedURLClassLoader
```

```
...
```

```
Started PetClinicApplication in 3.472 seconds (process running for 3.768)
```



# Case study: PetClinic (unpacked)

# Case Study: PetClinic

## Baseline (unpacked)

```
$ java -cp ... org.springframework.samples.petclinic.PetClinicApplication
```

```
Started PetClinicApplication in 2.585 seconds (process running for 2.741)
```

# Case Study: PetClinic

## Steps 1-2. Static CDS archive

```
$ java -Xshare:off -XX:DumpLoadedClassList=petclinic.classlist
```

```
-cp ... PetClinicApplication
```

```
Started PetClinicApplication in 2.646 seconds (process running for 2.859)
```

```
$ java -Xshare:dump -XX:SharedClassListFile=petclinic.classlist
```

```
-XX:SharedArchiveFile=petclinic-static.jsa
```

```
-XX:+ArchiveInvokeDynamic
```

```
-cp ...
```

```
(no application run)
```

# Case Study: PetClinic

## Step 3. Training Data

```
$ java -XX:SharedArchiveFile=petclinic-static.jsa  
      -XX:ArchiveClassesAtExit=petclinic-dynamic.jsa  
      -XX:+RecordTraining  
      -cp ... PetClinicApplication
```

Started PetClinicApplication in 1.466 seconds (process running for 1.699)

# Case Study: PetClinic

## Step 4. Cached Code

```
$ java -XX:SharedArchiveFile=petclinic-dynamic.jsa  
      -XX:+ReplayTraining  
      -XX:+StoreCachedCode -XX:CachedCodeFile=petclinic-code.jsa  
      -XX:CachedCodeMaxSize=128M  
      -cp ... PetClinicApplication
```

Started PetClinicApplication in 1.26 seconds (process running for 1.505)

# Case Study: PetClinic

## Step 5. Deployment Run

```
$ java -XX:SharedArchiveFile=petclinic-dynamic.jsa  
-XX:+ReplayTraining  
-XX:+LoadCachedCode -XX:CachedCodeFile=petclinic-code.jsa  
-XX:+UseRecompilation  
-cp ... PetClinicApplication
```

Started PetClinicApplication in 1.073 seconds (process running for 1.296)

# AOT-processed PetClinic (unpacked)

# Case Study: AOT-processed PetClinic

## Baseline

```
$ java -Dspring.aot.enabled=true -cp ...  
    org.springframework.samples.petclinic.PetClinicApplication
```

Starting AOT-processed PetClinicApplication using Java 22-internal

...

Started PetClinicApplication in 2.346 seconds (process running for 2.534)



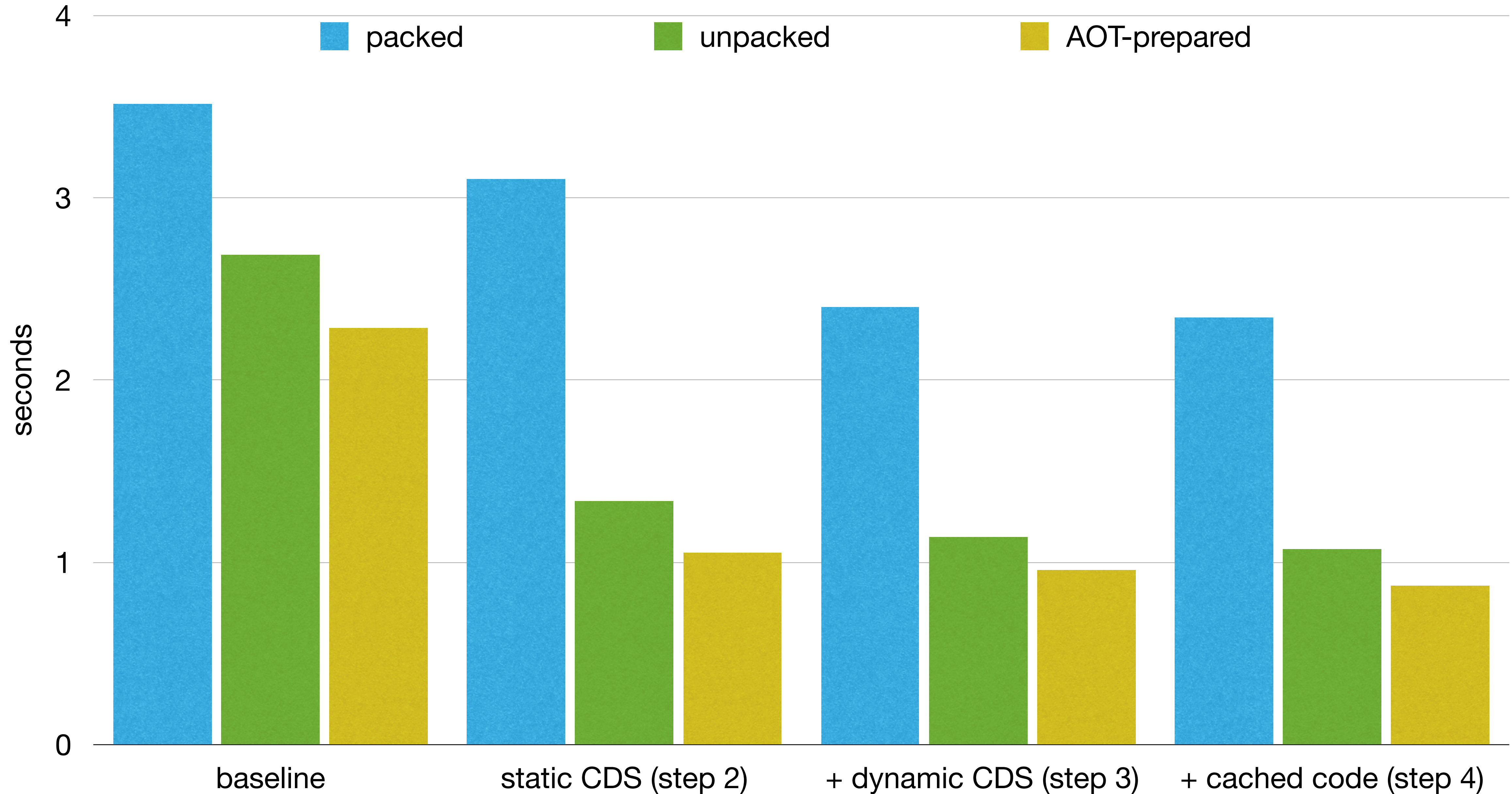
# Case Study: AOT-processed PetClinic

## Step 5. Deployment Run

```
$ java -XX:SharedArchiveFile=petclinic-dynamic.jsa  
      -XX:+ReplayTraining  
      -XX:+LoadCachedCode -XX:CachedCodeFile=petclinic-code.jsa  
      -XX:+UseRecompilation  
      -cp ... PetClinicApplication
```

Started PetClinicApplication in 0.841 seconds (process running for 1.058)

# Spring Boot PetClinic (macosx-aarch64)



# Diagnostic Output

- -Xlog:cds / -Xlog:cds\*
- -Xlog:scc / -Xlog:scc\*
- -XX:+CITime
- -Xlog:init