

Bachelor's Thesis

Using Virtualization for Building Images from Native Image Bundles for Deterministic Reproducibility

Student: Dominik Mascherbauer (k12005607)
Supervisor: Prof. Dr. Hanspeter Mössenböck
Co-Supervisor: Dr. Christian Wirth (Oracle Labs)
Start: March 15, 2023

**o.Univ.-Prof. Dr. Dr.h.c.
Hanspeter Mössenböck**
Institut für Systemsoftware

P +43 732 2468 4340
F +43 732 2468 4345
hanspeter.moessenboeck@jku.at

Sekretariat:
Karin Gusenbauer
Ext. 4342
karin.gusenbauer@jku.at

Linz, March 13, 2023

GraalVM Native Image is a solution for ahead-of-time compiling Java applications based on a closed-world-assumption. The created binaries can be executed independently of any JVM on the target system.

To deterministically¹ reproduce the build of an image, so-called *bundles* can be used. A bundle comprises all necessary inputs for the image creation process (class files, configuration, native build tools such as the C compiler including their versions, the provided flags, environment variables, etc.), so that a practically identical image can be recreated later on a different machine. While a bundle can provide for that, it cannot *guarantee* the independence of any input from the development machine, because, despite of specifying the inputs used for the build, the image builder still has access to the whole file system and to the network resources of the host that runs the builder.

The goal of this bachelor's thesis is to improve on this current state.

1) Implement the bundle creation process in a pipeline that uses a virtualization solution such as Docker to ensure that only the input actually configured is used. This guarantees independence of the bundle from the development environment, but is only available on Linux-based systems.

2a) Implement the same pipeline based on the JVM and its tooling capabilities (JVMTI).

2b) Make sure that the pipeline used in 2a cannot access input from the host system in a non-sanitized way, e.g., that it does not access files or environment variables that were not explicitly provided by the configuration.

¹ Native Image builds are inherently non-deterministic due to specified behavior of Java, e.g., random Hashtable ordering or undefined static constructor initialization order in the context on concurrent class loading. While this does have a relevant impact on the determinism of built images, it is an orthogonal problem to the determinism in this thesis that appears due to non-deterministic input

Step 1 is the core part of this thesis. Step 2 is an extension that allows its application to a larger set of operating systems and makes it more generally applicable. Step 2b can guarantee that Step 2 is as correct as the variant implemented in Step 1; it is not expected that this part is fully completed as part of this thesis. However, it should be shown that it is possible to implement it with some simple examples; a full implementation can be left for future work

The progress of the thesis should be discussed bi-weekly with the supervisor or the co-supervisor. The normal duration of a bachelor's thesis is 6 months. The mark "sehr gut" is only possible if the software and the written thesis are submitted not later than September 15, 2023. Within the first few weeks of the work a rough time schedule should be developed and discussed with the supervisor.