GraalVM Advisory Board February 2022 Meeting

Date

February 23rd, 2022

Participants

Alan Hayward (ARM), Aleksei Voitylov (BellSoft), Alina Yurenko (Oracle), Bruno Caballero (Microdoc), Chris Seaton (Shopify), Gilles Duboscq (Oracle), James Kleeh (Object Computing), Johan Vos (Gluon), Max Andersen (Red Hat), Michael Simons (Neo4j), Paul Hohensee (Amazon), San-Hong Li (Alibaba), Shaun Smith (Oracle), Thomas Wuerthinger (Oracle), Uma Srinivasan (Twitter)

Agenda

- Alina Yurenko: welcome new Advisory Board member: James Kleeh, Micronaut product development lead at Object Computing
- Alina Yurenko: recent and upcoming project updates
- Max Andersen: Full Java 17 support in 21.x? https://github.com/oracle/graal/issues/3984
  - answered in the ticket, will include @jerboaa's fix in 21.3 (in April)
- Max Andersen: ARM plans/efforts? Mandrel team are exploring here but still tricky to get hands on missing test environments
  - Thomas Wuerthinger: ARM is still a priority, a few people on the team are working on it;
  - we recently also got new hardware that we use for testing
  - Bernhard is working on it (https://github.com/oracle/graal/issues/2666)
- Thomas Wuerthinger: Java 18 support — is it highly requested by the community?
  - we assume it will be working for JIT; NI is a separate topic
  - more focus on LTS versions
- San-Hong Li: In 22.0 release notes https://www.graalvm.org/release-notes/22_0/, there are two statements: a. In the 'Platform Updates' section, "Dropped support for Java 8." has been declared. b. But in the 'Native Image' section, it also says that "GraalVM 22.0 will be the final release in which the native-image builder can be executed with JDK 8. "

Can you clarify the differences between a/b? Will GraalVM 22.0 be the final release for Java8 support?

- Gilles Duboscq: a) is closer to correct answer, we will remove this confusing statement from release notes
- JDK8-based builds can be built from source

- San-Hong Li: For the workloads that still need to stay on Java8 longer, do we have some plan/mechanism to support the backport of critical/vulnerability fixes for Java8?
  - there are only Java 11 & Java 17 based CE builds at the moment (we discussed this in August)
- Aleksei Voitylov: Are there plans for improvements to GC in CE?
  - Thomas Wuerthinger: we are working on a few things;
  - in general we think that for microservices concurrent GCs might not be needed
  - Aleksei Voitylov: we are experimenting with Serial GC as well
  - Thomas Wuerthinger: would be interesting to compare existing options with Shenandoah results
- San-Hong Li: what is the Loom support status?
  - Thomas: waiting for Loom support to finalize, e.g. preview feature status
  - we are working on it, but more active work will follow after support matures
  - the way it looks so far, supporting it on the GraalVM side shouldn't be an issue. Comparing to supporting additional GCs, Loom support seems easier to implement.