GraalVM Advisory Board August 2021 meeting

Participants

Alan Hayward (ARM), Aleksei Voitylov (BellSoft), Alina Yurenko (Oracle), Bruno Caballero (Microdoc), Chris Seaton (Shopify), Eric Sedlar (Oracle), Gilles Duboscq (Oracle), Jason Greene (Red Hat), Johan Vos (Gluon), Michael Simons (Neo4j), Paul Hohensee (Amazon), Roxana Bradescu (Oracle), San-Hong Li (Alibaba), Shaun Smith (Oracle), Thomas Wuerthinger (Oracle), Uma Srinivasan (Twitter)

Announcements

- New members joining the Advisory Board:
    - Paul Hohensee joining as a permanent representative from Amazon;
    - Aleksei Voitylov joining as a representative from BellSoft.

Project Updates

- **Native Build Tools**
    - New GraalVM Native Image Maven & Gradle plugins intended to simplify building and testing native images and bring JUnit testing support
    - Testing support was developed in collaboration with the JUnit, Micronaut, and Spring teams
    - More feedback from community is welcome
- **Native Image compilation process improvements**
    - We see community members reporting compilation process improvements, such as build times (up to 20%-30%), memory usage and binary size.
- **JFR Support in Native Image**
    - Initial support in 21.2, extended support in upcoming releases
    - Developed in collaboration between Oracle Labs and Red Hat
    - San-Hong Li: do you plan to add JFR Event Streaming support?
        - Thomas: we will discuss it with the Native image team and follow-up in the next meeting;
    - Thomas: what about support for other monitoring tools like OpenTelemetry — should we provide support for it as well? Feedback would be helpful.
        - **to-do**: follow-up on this with the Red Hat & Spring Boot teams
- **Class pre-definition in Native Image**

    - Classes that need to be loaded at run time can be made available to the static analysis so that they are included in the closed-world analysis;
    - This feature was developed in collaboration between Oracle Labs and Alibaba
    - Aleksei Voitylov: What’s the license of those plugins?
        - Thomas Wuerthinger: It’s The Universal Permissive License (UPL)
- **Other updates**
    - Performance improvements thanks to new optimizations, such as speculative guard movement optimization (+ 4.2% geomean throughput score of GraalVM Community on SpecJVM2008)
    - Truffle: new compilation queuing heuristic enabled by default, which improves the warmup time of the polyglot runtime on many workloads
    - Compatibility and performance updates to JavaScript, Python, Ruby, R
    - Multiple contributions to TruffleRuby from Shopify

Q&A

1. San-Hong: To better recognize the people's contributions, do we (the GraalVM community) consider setting up the different **project roles, such as Author, Committer, Reviewer** (examples in OpenJDK) according to participant's contribution?
        - Thomas: we invite contributors to the GraalVM GitHub organization, which is probably similar to the Author role; this shows that they are a part of the community/contribute
        - over long-term we are working on additional tooling, e.g. to add a Reviewer role that will allow for example to merge changes
        - we can follow-up on this in the next meeting
2. San-Hong: We(Alibaba) start working on the work for the standalone static analysis tool. Can you guys give us **fast feedback on the PRs in the queue**? The below PRs are pending review:

oracle/graal#3672

oracle/graal#3672

oracle/graal#3666

oracle/graal#3666

  - we requested reviews and will follow-up on PRs directly

3. Johan Vos: I recommend **to not link to internal numbers/issues/PR's in public docs** (issues/PR's) as external people can't access those anyhow.
    - Thomas; agree; let's discuss such issues/PRs directly
    - Maybe instead we can have public information linked internally in our issue tracker
4. Uma Srinivasan: Thank you for the rapid support and help with fixing https://github.com/oracle/graal/issues/3568
5. Uma Srinivasan: Would like to understand the plans for **Autovectorization support in CE**. TwitterVMTeam had an intern this summer who resurrected the 2019 prototype and provided performance enhancements to it. Seeing more stability and performance with micro-benchmarks relative to C2.
    - Thomas: yes, there are still plans for this
    - the Guard Movement optimization in 21.2 should contribute to performance improvements in Community as well
    - the next step will be adding some basic autovectorization for loops

- Uma: One more q on Autovectorization support. Are you working on codegen support only for x86 or for ARM as well?
    - Thomas: we are working on both

6. Jason Greene: Just a **reminder on a PR that provides some useful analysis capabilities** with neo4j graph output. It seems to have passed review, but waiting for merge.

https://github.com/oracle/graal/pull/3128

  - to follow up - Alina (done)

7. Gilles Duboscq: We are planning to **stop building JDK8-based builds**; share with us your feedback/concerns if there are any.