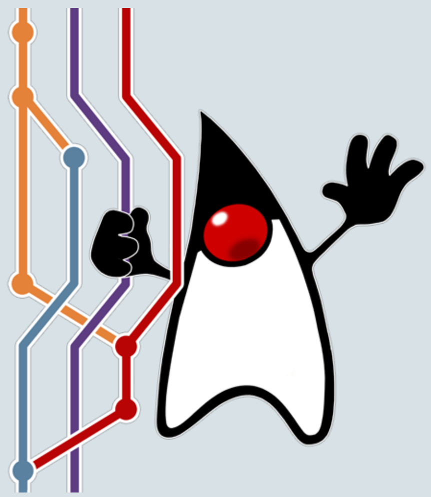
Update on Project Skara and Git

Source code management options for the JDK



Joseph D. Darcy (darcy, @jddarcy)
Joint work with Erik Duveblad (ehelin) and Robin Westberg (rwestberg), among others
Java Platform Group, Oracle
OpenJDK Governing Board meeting
October 2019,

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Preface: bylaws and hosting

- Section 6 of the OpenJDK Bylaws:
  “A Project may have web content, one or more code repositories, and one or more mailing lists. Projects are expected to operate in an open, transparent, and meritocratic manner. Their alignment with these principles will be monitored by the Governing Board.”
- Appendix A of the OpenJDK Bylaws:
  “The data stored in any infrastructure provided for use by Community members must be made available by some means that enables, without undue effort, the construction of a complete functional clone of that infrastructure and its data as seen by the entire Community.”

Background

- Call for Discussion: New Project: Skara -- investigating source code management options for the JDK sources, July 2018
- *Investigating source code management options for the JDK sources,* OpenJDK Committers' Workshop August 20181
- Project created September 2018
  - Mailing list: skara-dev
  - Wiki
- *Update on Project Skara and Git*, OpenJDK Committers' Workshop August 2019

Mirror, mirror

https://github.com/openjdk/

- Git mirrors of hg repos of consolidated JDK lines of development (and a few other repos from codetools) including:
  - jdk/jdk
  - jdk13
  - jfx
  - amber
  - loom
  - panama
  - shenandoah
  - tsan
  - …

Imported repos reformatted commit messages

- Current:

```
8225035: Thread stack size issue caused by large TLS size
Summary: Adjust thread stack size for static TLS on Linux when AdjustStackSizeForTLS is enabled.
Reviewed-by: dholmes, fweimer, stuefe, rriggs, martin
Contributed-by: jeremymanson@google.com, fweimer@redhat.com, jianglizhou@google.com
```

- Reformatting works better with git tooling including `git log`
- Multiple authors supported
- Git has separate notions of author and committer; could be used for backports too

- Proposed

```
8225035: Thread stack size issue caused by large TLS size

Adjust thread stack size for static TLS on Linux when AdjustStackSizeForTLS is enabled.

Co-authored-by: Florian Weimer fweimer@redhat.com
Co-authored-by: Jiangli Zhou jianglizhou@google.com
Reviewed-by: dholmes, fweimer, stuefe, rriggs, martin
```

“The Skara tooling is now open source”, June 2019

https://github.com/openjdk/skara

“The Skara tooling includes both server-side tools (so called "bots") as well as several command-line tools:

- git-jcheck
- git-webrev
- git-defpath
- git-fork
- git-pr
- git-translate
- git-info
- git-skara”

New candidate JEP 357, July 2019

Migrate from Mercurial to Git

Summary

Migrate the OpenJDK Community's source code repositories from Mercurial (hg) to Git.

Goals

- Migrate all single-repository OpenJDK Projects from Mercurial to Git
- Preserve all version control history, including tags
- Reformat commit messages according to Git best practices
- Port the jcheck, webrev, and defpath tools to Git
- Create a tool to translate between Mercurial and Git hashes

Second JEP to determine hosting provider, TBD

- *Not* looking to change other infra (bug tracker, etc.)
- Support different working styles and provide continuity
  - Allow traditional mailing lists and webrevs to be used
  - Native hosting provider interactions too
  - Command line support
- Avoid vendor lock-in
  - Bi-directional mirroring of pull request comments and email threads

Gaining trial experience with Git and Skara

- Project Loom is now doing its development on github with Skara
- Discussion of moving jmc to github
- OpenJFX has moved to github with Skara, including reviews, etc.
  - Pull requests accepted and integrated
  - Skara team responding to feedback from new users

Model for using a hosting provider

- A developer has one more clones on the hosting provider *server*; self-service, done quickly
- These can then be cloned locally and worked on
- Pushes are made to the personal clones
- A *pull request* (PR) is made from the personal clone to the master
  - Bots and people review the request
  - Bots then handle the actual push
  - Users would *not* push directly to master
- Need an account on the provider to integrate; otherwise, require a sponsor

Future steps

- Gain experience from projects using Skara on github, tuning support
- Build-out additional boundary systems
  - Skara analogue of submit repo
  - JBS jira integration/hg-updater replacement
- Publish second JEP
- More hg ⇒ git transition tips from Skara team

Questions?