Scala Center updates
Q1 2021 Advisory Board meeting

Scala Center team: Julien Richard-Foy, 60%; Jamie Thompson, 100%; Vincenzo Bazzucchi, 100%; Fengyun Liu, 100%; Adrien Piquerez, 100%; Meriam Lachkar, 100%; Katja Goltsova, part-time intern; Maxime Kjaer, part-time intern; Sébastien Doeraene, 100%; Darja Jovanovic, 100%, Vincent Derouand, 40% since March 1st; VirtusLab team: Tomasz Godzik, 100%; Wojciech Mazur, 100%.

At a glance

MOOCs

@julienrf @vincenzobaz

Effective Programming in Scala

We have finished recording all the lectures (~10 hours of video in total). We are now finalizing the assignments and instructions so that we will be ready to publish the course along with the release of Scala 3.0.0 in the coming months.

The course Effective Programming in Scala has been designed to train non-Scala programmers to become ready to work in Scala. It teaches the essentials parts of the language and the common practices.

Other courses

We upgraded all the assignments of the [Scala Specialization](#) and the [Akka course](#) to Scala 3.0.0-RC1. We reviewed and polished the port of the assignments to Scala 3. The release of Spark 3.2 snapshots allowed us to test the assignments for the Big Data course, although breaking changes in the reflection API in Scala are holding back the migration of the `timeusage` assignment which depends on the typed Dataset API of Spark. We are looking for workarounds to address this issue which we believe could help Scala 3 early adopters relying on Apache Spark to upgrade their codebases.

Google Summer of Code

@vincenzobaz, @darjutak

We were accepted as an Organization for GSoC 2021! ([Blog post](#))

Our team of mentors committed to guiding and welcoming students as well as officially signing up for the event. We communicated about our incredible team of 19 mentors and the 29 project ideas that they proposed. We keep mentors up-to-date via a mailing list, and they can exchange ideas in an internal Slack workspace.

We received interest in the Contributors forum and mentors are reporting being contacted by enthusiast students. Student applications officially start on Monday, March 29th.

Let's Talk About Scala 3 Video Series

@vincenzobaz @bishabosha @liufengyun @julienrf @sjrd

February and March were important months for our video series.

In February we recorded with Jamie (whose video is out) and Fengyun (scheduled to go public after Easter Break). In March we recorded content for Julien's and Sébastien's videos.

"[Pick any card, or Write Better Data Structures with Scala](#)" is a video for beginners on how to improve the precision of their data structures to avoid validation/run-time errors. It also thoroughly covers enums. The video was released on 18th March 2021.

The next videos to be released are about:

- Compiler plugin development in Scala 3
- Following the recommended versioning scheme in libraries with sbt-version-policy
- Best features of Scala 3 for Scala.js users

We are tuning the process iteratively and these two months were extremely positive. The collaboration with 47 Degrees is becoming smooth and we are getting used to the pace and rhythm of the project. The lessons from the "Speaking on camera" training seem to help each team member in different ways, but in general it has helped speakers feel more comfortable and has provided us with tips that improve the quality of the videos.

We planned recordings for April which will see Meriam and Anatolii (from LAMP) in the studio.

Scala3-migrate

@mlachkar

Scala3-migrate is a tool to migrate projects from the latest 2.13 version to the latest 3.0.0. During the last two months, we focused on releasing the project which required fixing blocking bugs, high memory usage, and improving log messages.

We released the first public version v0.3.1 with the following features:

- Migrate both scalacOptions and libraryDependencies
- Fix the syntax and migrate the code to make it compile in Scala 3

We wrote a specific page about Scala3-migrate in the migration guide.

Feedback is especially welcome for this project.

Migration Guide

@vincenzobaz @MaximeKjaer

We started to work on addressing specific tasks and blockers encountered by Spotify during the migration of the Scio project to Scala 3. The team has been focusing on getting to know the codebase as well as the tasks and planning the project.

The goal of the project is to document and communicate about the migration of a large codebase which relies on advanced language features such as metaprogramming, reflection and code generation.

Scala 3 Compiler improvements

@bishabosha @liufengyun @sjrd

In this quarter we have been focusing on improving stability as we reach the release of Scala 3. We have been helping the core Scala 3 team at LAMP with fixing as many bugs as possible, as well as stabilizing the TASTy format.

We have also significantly improved the experience in case of TASTy file version mismatch. We have designed a new header format for a TASTy file. It is now aware of experimental TASTy releases, and stores the compiler version that produced the TASTy file. Users will now receive instructions on what to do if the TASTy version of a library dependency does not match what the compiler expects. Those improvements were released in Scala 3.0.0-RC1.

Other areas that received particular attention are:

- F-bounds, improving compatibility with Scala 2
- Java interoperability
- Pattern matching
- Metaprogramming API, notably with scala.deriving.Mirror
- Initialization checker
- Scala.js support

Scala Debug Adapter

@adpi2

We released the first version of the Scala Debug Adapter, an implementation of a DAP server for the Scala language on the JVM. As part of this project, the sbt-debug-adapter makes it possible to debug an sbt project in Metals using sbt as the build server. It is already available in Metals out-of-the-box.

sbt

@adpi2

We have been very active in stabilizing sbt 1.5.0 release candidates. If no serious issue is found by April 3rd 2021, sbt 1.5.0 final will be released, with stable Scala 3 support.

Scalafmt and scalameta

@tgodzik

We have worked on improved support for Scala 3.

- The Scalameta parser now fully supports Scala 3 syntax
- Scalafmt now supports all new constructs, except optional braces

Metals

@tgodzik

We published two bugfix releases of Bloop: [v1.4.7](#) and [v1.4.8](#).

We released Metals [v0.10.0](#), with multiple improvements and features:

- Insert inferred type refactoring
- Type decorations for definitions
- Scala 2.13.5 and 3.0.0-RC1 support
- Multiple Scala 3 support improvements

TASTy Reader For Scala 2

@bishabosha

As Scala 3 reaches its final release, we have been stabilising changes to the TASTy format, which need to be ported to Scala 2. We also decided that efforts to support Scala 3 exclusive features of TASTy in Scala 2 should only be added with demand from users.

Highlights

We added support for Scala 3.0.0-RC1, released in Scala 2.13.5 on February 22nd. This release brings a new header format for TASTy files, which is aware of experimental releases, and a string for the compiler version that produced the TASTy. Users are now given instructions on what to do if the TASTy version of a library dependency does not match the compiler's expectation.

We also fixed a bug in mixing Scala 2 and 3 macros in TASTy: if the Scala 3 implementation was an inline method, and not a macro, then the Scala 2 macro would not be discovered and the user would see an error.

Scaladex

@adpi2

We moved Scaladex to a new virtual machine with better settings and stricter security rules. We upgraded Elasticsearch to 7.10.2 to improve the performance.

TASTy manipulation library

@cache-nez

We added support for unpickling more trees, including ones requiring some support for Symbols. The next steps will involve unpickling types and implementing the subtyping relationship.

Scala.js

@sjrd

We published [Scala.js 1.5.0](#). This was mostly a bugfix release, including fixes for some important bugs related to dynamic module loading, which was introduced in 1.4.0.

The main non-bug fix improvement is the introduction of custom JS function types. It is now possible to declare custom definitions for JavaScript functions with arbitrary apply signatures. Notably, this allows to declare types for JS functions with varargs.

We have fixed a number of issues related to floating point numbers. In particular, the %e, %f and %g conversions in java.util.Formatter/String.format now produce exactly the same results as on the JVM. This improves portability, and makes it easier to write cross-compiling tests in some instances. These fixes will be part of the next release.

Scala Native

@wojciechmazur

After the release of Scala Native v0.4.0, we have been fixing follow-up issues. We also worked on link-time conditions, allowing us to discard branches of code based on the used platform. Based on that functionality we have started working on first-class Windows support. We also worked on linking Scala Native projects as shared libraries.

Highlights

Support for shared libraries

We have been working on emitting shared libraries from Scala Native. The model is based on annotating methods that should be treated as entry points in shared libraries. Each method with `export()` annotation would be accessible in the resulting library and would follow C naming conventions. This will enable developers to use Scala Native dynamic libraries with other compiled languages like C/C++, Rust or Go.

Link-time conditions

We introduced a set of link-time resolved properties, and based on them we introduced conditions resolved at link-time. With a new type of conditional branching, we are able to define platform-dependent inclusion of code. This feature allows us to include support for both Unix-based and Windows operating systems in a single library, without the need for further cross-building and creating complex definitions of library dependencies. This will be the basis for Windows support.

Miscellaneous
- Run the upstream scala/scala tests in the Scala Native CI
- We identified and fixed performance regressions introduced in v0.4.0

Communication and Management

@darjutak

- Published 4 blog posts:

  [Preventing Version Conflicts with versionScheme](#)

  [Tuples bring generic programming to Scala 3](#)

  [Scala 3 Developer's Preview Survey Results](#)

  [GSoC 2021 Here we come!](#)

- Started publishing this year's project's roadmaps, [Contributors threads](#)

- Facilitated the [2nd Organizers Summit](#) during the ScalaLove conference, mid Feb.

- Started "Let's talk about Scala 3" video series, with high view rate and great feedback. We will release 1 video every 2 weeks, and have the material up to October 2021 covered.

  [Setting up a dev environment with Coursier](#), by Eric Loots, Lunatech
  Released 4th March, by today 2,229 views, 89 likes, 13 comments (and answered).

  [Pick any Card: Write Better Data Structures with Scala](#), by Jamie Thompson, Scala Center
  Released 18th March, by today 1,266 views, 92 likes, 14 comments (and answered).

  [Compiler Plugin Development in Scala 3](#), by Fengyun Liu, Scala Center.
  To be released tomorrow (31st March).

- Employed Vincent Derouand as a communication specialist for 40%, 1 year mission, started 1st of March.
- In the process of engaging Valée Pedroni as a communication intern, 80%, 1 year mission starting on 15th of April 2021.
- In the process of choosing one more associate through the Swiss unemployment program Syni, for 80% for a 6 month mission. We hope to choose a candidate ASAP.

  The communication team will be tasked with:

  - Document and standardise current communication processes

- Design and coordinate with the team and partner organisations a communication strategy for the Scala Center

- Train members to create and produce communication deliverables in line with the strategy and standards (blogs, reports, updates, other)

- Identify and map out the key stakeholders and media institutions

- Organize/facilitate/communicate about events (Contributors summit, Organizers summit, online Scala sprees, other)

- Lead the Diversity & Inclusion project(s)

- Management team participated in the vision workshop, in process of creating the Scala Center 5-year strategy
- Signed SwissBorg as an Affiliated member
- In process of signing with Databricks as an Advisory board member