Scala Center updates
Q4 2019 Advisory Board meeting

Scala Center team: Jorge Vicente Cantero, 100% until November 15; Julien Richard-Foy, 60%; Jamie Thompson, 100%; Darja Jovanovic, 80%; Sébastien Doeraene, 100%; VirtusLab team

At a glance

MOOCs

@julienrf

We have published all the assignments of all our online courses in their updated version (Scala 2.13, except for assignments that use Spark, which are kept on 2.12). For the reactive course, we also bumped the Akka version to 2.6.0.

In parallel, we have been working on improving the content of the existing functional programming courses, both to include new features of Scala 3 and to simplify the curriculum. In particular, we gave a lecture on “type-directed programming” (implicits) with EPFL students, and created a new assignment on this topic. We plan to release these changes when Scala 3 will be released.

Finally, we have been discussing with Scala companies about the curriculum of a new Scala course which would be more business oriented (understand, less theoretical, more practical).

TASTy Reader for Scala 2 (SCP-018)

@bishabosha

We have been working on the TASTy Reader for Scala 2 and made good progress. We recently started integration testing with community libraries.

As a highlight of the progress: using the TASTy reader, it is possible to extend and use in Scala 2 the following trait, compiled with Dotty 0.20.0-RC1 and using lambda types:

```
trait Reader[Ctx] extends Monad[[X] =>> Ctx => X] {
  def [A, B](r: Ctx => A) flatMap(f: A => Ctx => B): Ctx => B
  def pure[A](x: A): Ctx => A
}
```

We have also introduced TASTy Test into our workflow, to build a regression testing for Scala 2 code that depends on code compiled with Scala 3. It supports assertions about compile errors as well as validating code that should work. This allows more productive time spent iteratively adding support for new language features, and finding incompatibilities at the binary level.

We expanded testing to an sbt project, compiling with the bootstrapped scalac with TASTy Reader, and depending on community libraries compiled with Scala 3. This will help discover gaps in language support for real world code. For example we can compile and run a ZIO application when the library is published for Dotty.

We also abstracted components of TASTy into a separate library, `ch.epfl.lamp:tasty-core`, cross-compiling with Scala 2 and 3, starting with definitions for byte serialisation/deserialisation and semantic names for tags used to define trees in TASTy. Future work will add a flexible framework on top for tools to work with trees and symbols, starting with more shared implementation between Scala 2 and 3.

SemanticDB for Scala 3

@bishabosha, @olafurpg, @odersky

We implemented support for generating SemanticDB in the Dotty compiler, so that we can bring Metals to Scala 3.

This involved adding a small implementation of protocol buffers within the Dotty compiler to produce SemanticDB files, which use the binary format. These files contain sections for metadata about Scala definitions and usages of them. Together, these will be used by Metals to enable features such as “go to definition” and “find occurrences”.

VirtusLab will be responsible for using the SemanticDB generator for Dotty to build support in Metals.

Metals

VirtusLab

- Automated integration with main build tools such as sbt, maven, gradle, mill
- Document symbol highlighting
- Rename and goto implementation
- Document folding
- Support for running and debugging
- Improvements to code completion
- Numerous bug fixes
- Promoting Metals on 3+ conferences and expanding Metals into new niches like support for Eclipse and Eclipse Theia that results in web-only IDE based on Gitpod

Bloop

@jvican

We finished the development of bloop 1.4.0, which is scheduled to be released on Wednesday December 18, with an accompanying Metals release shortly after that. Here are the highlights of this release:

- Scala 3 support in Bloop for Metals to work
- Built-in support for debugging Java and Scala applications
- Full implementation of “offloading compilation from sbt”, including the infrastructure required to quickly implement the same thing from other build tools such as Gradle
- Implementation of bloopgun, a JVM-based nailgun client; it can be run via coursier or compiled to native code with GraalVM.
- Installation changes: Bloop can now be used and installed without Python, which means it is only a coursier invocation away
- Hydra support (collaboration with Triplequote)
- Performance fixes to scale to very big codebases, in compilation, build loading, testing and running

Bloop has received a lot of contributions from the community during the past few months and its adoption is growing steadily.

Scala Libraries and Documentation

@julienrf

- Minor follow-up work on Alvin Alexander’s contribution to the documentation (#1564).
- We investigated the causes of unreliability of scaladex, which sometimes misses some projects’ releases. We fixed a race condition and are still monitoring what’s going on (#550).

Scala.js

@sjrd

We published Scala.js 0.6.29 shortly after the last AB meeting, with support dynamic **`import()`** calls and a new major optimization (see last report). We also published Scala.js 0.6.31 3 weeks ago, with facade types for JavaScript's BigInts and an implementation of `java.util.IdentityHashMap` (both by external contributors). Scala.js 0.6.30 was declared dead-on-arrival due to a mistake in the release process.

More significantly, we finalized the changes that we wanted to integrate into Scala.js 1.0.0, and published Scala.js 1.0.0-RC1. We helped a few core projects of the community to upgrade and publish 1.0.0-RC1, notably mill, µTest and sbt-web-scalajs.

We discovered a few mistakes in the handling of versions for post-v1.0.0, which we fixed. A bug report in 0.6.x caused us to reconsider a detail of the IR encoding for Java static methods, which we also changed. We will need a v1.0.0-RC2 to validate those changes, although they have no user-visible impact. Artifacts for 1.0.0-RC2 are already on Maven Central, and the announcement should be out on December 13 (accessible from the news page).

We encourage all Scala.js users who are able to do so to test the RC before the end of January 2020. If no critical issue is found by then, Scala.js 1.0.0-RC2 will become 1.0.0 final.

scalajs-bundler

@julienrf, @sjrd

We updated scalajs-bundler to support Scala.js 1.0.0-RC1 and published version 0.16.0. That version of scalajs-bundler should also support subsequent RCs and 1.0.0 final out of the box.

Scala Sphere

@darjutak

Website

- Helped organise the Scala spree and ScalaBridge
- Co-organised the third Scala Contributor Summit
- Lead the “Community issues” and “Organisers” topic at the Contributors Summit
- Helped during the conference

The VirtusLab team has been the main organiser of the Sphere.it conference, including the Scala Sphere (the Scala track) in Krakow, Poland for the last 4 years. This year they proposed to sponsor the Contributors Summit during the Scala Days in June and to continue the conversation by colocating the following Summit with the Sphere.it conference. Thanks to their initiative and backing, we were capable of bringing about 100 contributors together to speak about many important topics, such as:

- Growing community
- Data science
- Scala 3
- Open source
- Enterprise
- Ecosystem metrics
- Community
- Scala Meetup and Conference Organisers
- Moderators

We invite you to read the “Scala Contributor Summit(s)” blog post for more information on this general topic.

Given the feedback, we will keep the Summits going, colocating with ScalaDays and other conferences that attract numerous contributors.

SIP meetings

@darjutak

- October private meeting
- November public meeting: minutes
- Coming up: December public and March 2020 3-day retreat

The SIP Committee had a complex task of analysing about 50 changes to the language, proposed by the Dotty team, between November 2018 and 2019. This November the SIPs are entering a new stage of the process which is still being discussed: now that we discussed most of the features with the community involvement, what is the best path forward? While we are discussing the details, we are all in agreement that the SIP Committee has to be involved in assessing the changes, giving recommendations, act as a body that can communicate the process and changes with the wider public.

Scala Days 2020 PC + CFP

@darjutak

- Coordinated the Scala Days 2020 succession
- Formed the PC
- Released the CFP (please inform your teams)

ScalaDays 2020 will take place in Seattle, May 18-22, and in Berlin, July 1-3. The organisers will be: Lightbend, Trifork, and 47 degrees. The Scala Center will lead the Program Management and surrounding events, be available for consultation and support, but not nearly as implicated as for the 2019 edition.

Internal organization

@darjutak

- Creating the mid-term strategy (will present during the meeting)
- Hiring 3 full time engineers
- Exploring the student projects option (also, more during the meeting)

Moderation

During the Scala Contributors Summit, during the “Community” topic, about 15 contributors and organisers came to a conclusion that:

- Moderation is the key to online community communication success
- Creating a Moderators Summit is an excellent step forward
- There should be a wider committee composed of community managers from different programming languages

We are dedicated to pursue the Moderators Summit in the next 6 months, more updates coming.