Making online education accessible with Typesafe

Coursera's remarkable online education service delivers high quality classes directly to users desktops, and turned the company into an overnight success. Delivering any service in a scalable, seamless manner however, is hard to do without the right tools. Coursera decided to utilize the Typesafe Reactive Platform to handle the massive scale and concurrency issues they faced.

About Coursera

Coursera believes in connecting people to a great education so that anyone around the world can learn without limits. Coursera is an education company that partners with the top universities and organizations in the world to offer courses online for anyone to take, for free. Their technology enables their partners to teach millions of students rather than hundreds. Coursera envisions a future where everyone has access to a world-class education, which so far, has only been available to a select few. They aim to empower people with education that will improve their lives, the lives of their families, and the communities they live in.

The Business Problem - Scaling to support 6.3 million students

Having a code base that was primarily PHP-based turned out initially to be a good strategy for Coursera because their courses and applications could be developed and deployed quickly. As time progressed, however, a growing customer base of over 6.3 million students, a catalog of over 568 different courses and a quickly expanding development team, began to highlight PHP's weaknesses, and it started to become clear that it wasn't going to work as a long-term development platform. The team, lead by Brennan Saeta, decided it was time to start looking for alternatives that met both the physical and developer scalability needs foreseen in Coursera's future.

Revision 1.0

Evaluating Technologies - Deliverability and developer productivity

The team members drew on their personal experiences to help determine which tools to look at and quickly came to consensus on a strategy. There would be two parallel evaluation tracks that would exercise the tools in real-world scenarios. The technologies that would be evaluated were:

Django and Python
Typesafe Reactive Platform

This strategy would enable the team to gain an understanding of the fit and finish of the tools in their development environment.

Django and Python

Even though the Django/Python route was amenable to rapid development, it presented a number of issues over time. The biggest turned out to be:

Dynamic typing

The Coursera team wanted better performance, yet still needed developer productivity to be as high as possible. Dynamic typing was deemed an impediment whereas static typing enabled substantial developer productivity.

Performance

The Python stack turned out to be even slower than the PHP stack it was intended to replace.

Concurrency

Futures, asynchronous execution, and simply having the JVM platform (that has been designed for concurrency) was a huge driving force for moving beyond Python and PHP.

A number of tools and services were built on the Django/Python stack, however new services are built on the Typesafe Reactive Platform.

Typesafe Reactive Platform

The Typesafe Reactive Platform track was proceeding more successfully. The components utilized in the evaluation were Play Web Framework and the Scala programming language to provide technical parity between the evaluations. Play Framework is based on a lightweight, stateless, web-friendly architecture that features predictable and minimal resource consumption for highly scalable applications. Scala is a general purpose programming language designed to express common programming patterns in a concise, elegant, and type-safe way. It smoothly integrates features of object-oriented and functional languages, enabling Java and other programmers to be more productive.

Initially, there was hesitation around Scala. It was perceived as a “new” language that would be difficult for developers to get up to speed on. Additionally, the team was skeptical about the maturity of Play. The evaluation put both concerns to rest. Brennan Saeta, a Software Engineer on the team, had been utilizing both technologies for nearly two years. He was able to demonstrate the robustness of Play and ease of Scala by quickly deploying Coursera’s first Play application in May of 2012. This single application showed such value and technical traction within Coursera that all new services in Coursera’s service-oriented architecture are written on the Typesafe Reactive Platform as a result. The team has built tooling around Play so that it’s very easy to use and works seamlessly within the Coursera environment.

The Eventing System, while more simple than other applications, has become the most stable application that we run. It's incredibly reliable.

Brennan Saeta,
Software Engineer, Coursera

Massive concurrency, minimal resource usage

There were a few things that the team explicitly pointed to that would be difficult, if not impossible, to accomplish in another framework or another language, such as the PHP and Python stacks.

> In other stacks, performing operations concurrently involves *forking a whole new process for every request*, which presents massive overhead. Play, on the other hand, allows massive concurrency with *minimal* resource usage.
>
> Concurrency, being able to do things in parallel, background threads, pipelines, or queues in memory is something that would have been impossible in another framework. It's so easy in Play and Scala, however, that it's been a huge win for Coursera.

Coursera is a big fan of Futures and use them extensively too.

Play combines useful and powerful libraries without being opinionated. It is very extensible and allows us to plug in our own custom logic or tools and features to make it all work seamlessly.

Brennan Saeta,
Software Engineer, Coursera

Moving Forward - Going "all in"

Because the Typesafe Reactive Platform runs on the JVM, it's very easy to use existing JARs and packaging tools. This allows Coursera to continue to utilize Artifactory to share code within the organization, for instance.

Utilization of the other components of the Typesafe Reactive Platform is expanding too, with both Akka and Slick being utilized in projects. Akka is a toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications. Slick is a modern database query and access library for Scala, where you can write your database queries in Scala instead of SQL, thus profiting from the static checking, compile-time safety and compositionality of Scala.

Starting out, the IDE support was not as good as the Java IDE's, but that is rapidly improving with Typesafe's work on Eclipse and the support of other vendors like JetBrains' IntelliJ IDEA.

Daniel Chia,
Software Engineer, Coursera

Akka and Amazon SES, or the Amazon Simple Email Service, are very heavily utilized in Coursera's email sending service. When instructors need to send an email to their tens or hundreds of thousands of students, they want to do that as fast as possible and in parallel. While Amazon SES works reliably most of

the time, when it fails, Akka's Actors ensure applications are not sending not above the rate limits, and that each email goes through correctly.

The infrastructure team liked that Anorm did not abstract too heavily from the database layer, but found that Slick gave them much more flexibility and power while still making it fairly obvious what kind of SQL query is being done. Slick is very explicit about when a SQL query is actually executed. While it looks as if a developer is manipulating the entire database in memory, it's actually being done via SQL. Coursera has a lot of code currently in Anorm, but are migrating it all to Slick at this point, and already have a number of projects in production serving user traffic. Underneath the hood they are using the c3p0 database connection pool as opposed to what Play is bundled with, which is BoneCP.

We are going "all in" on sbt too. We have configured our build process to leverage sbt 0.13's new performance capabilities. We have a thirty-plus project, single sbt build file that builds almost all of our Scala. We have our own plug-ins, and have done significant customization. I for one have been very pleased with it.

Brennan Saeta,
Software Engineer, Coursera

Conclusion

Currently at Coursera there are approximately twenty projects built with Play. About half of the applications are internal and the other half are actually handling user traffic and serving up course content. The only reason an accurate count can't be given is that so many new applications and tools pop up so frequently! In closing, Brennan states "Though we've hit some rough spots in Scala and Play along the way, we're willing to invest in it for the long-term, as the entire Typesafe Reactive Platform is on an amazing trajectory."

*We measured just recently; we're serving **tens of millions** of requests per day using Play.*

Brennan Saeta,
Software Engineer, Coursera

*The **Typesafe Reactive Platform** is a modern software platform that makes it easy for developers to build scalable software applications. It combines **Play Framework**, **Akka** runtime, the **Scala** programming language, and robust developer tools in a simple package that integrates seamlessly with existing Java infrastructure. Commercial support and maintenance is available for the Typesafe Reactive Platform through the **Typesafe Subscription.***