



Bringing Reactive Applications to the Java Virtual Machine

HolidayCheck's journey with Typesafe

Problems scaling out new international sites dictated a change in HolidayCheck's application development platform and the adoption of a new approach to developing applications.

About HolidayCheck

The HolidayCheck idea dates back to 1999 when Markus Schott and Sascha Vasic, frustrated with the lack of authentic information when choosing a hotel, decided to create a website that would give others the opportunity to read real opinions and reviews instead of basic catalogue descriptions. Since the 2004 launch of the official company in Switzerland, HolidayCheck has seen rapid growth and expanded internationally. Today HolidayCheck is available in 10 languages worldwide.

Each month 20 - 30 million people visit HolidayCheck websites to learn about hotels and destinations by viewing feedback from over 10 million user opinions. Information is available on more than 300,000 hotels and 100,000 points of interest in the holiday destinations.

The Business Problem

HolidayCheck is the largest German speaking online travel site/agency, having been online now for more than ten years. In their peak-booking season they have over one hundred million page impressions per week. The platform started with a typical LAMP stack, which worked well for the initial site as it was meeting usual business goals and classical web business KPIs like Shopper Conversion and Look2Book.

Problems arose, however when the team tried to duplicate the code to deploy slightly different products for different markets – the biggest problems being related to platform/code complexity, stability and performance. Christopher Schmidt, Senior Lead Architect responsible for the online platform was tasked with designing, developing and implementing a new architecture based on a new technology stack.

The Evaluation

In 2010, Christopher attended the first Scala Days in Lausanne, Switzerland where he was introduced to the Scala programming language and the Akka runtime. 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. Akka on the other hand is a runtime for building highly concurrent, distributed, and fault tolerant event-driven applications.

Christopher's platform reworking requirements included a major objective to deploy a new platform or change an existing one with new/changed front-end applications within a matter of days – a lofty goal considering the complexities involved.

Christopher felt that both the Scala and Akka held a great deal of promise for his project, and began to investigate them more closely. For Scala and Akka to become the building blocks for Christopher's new architecture however, there were additional technical requirements that needed to be met:

- High performance websites, like Twitter, Yahoo, LinkedIn demonstrated the scale.
- Dynamic cloud and cluster deployments as opposed to "static" or predefined configurations.
- Polyglot persistence instead of a one size fits all MySQL.
- Loosely coupled and massively parallel, asynchronous from top to bottom.
- Utilizing industry standards instead of JSR stuff,
- Simplicity regarding APIs, semantic and usage, pragmatism (instead of dying in beauty).

There was no dedicated evaluation to use Scala or not, it simply was clear that Scala added significant value (because of Functional Programming, the collection API, Traits) so we used it. We wanted to use Actors for parallel processing, so Akka was a natural choice.

*Christopher Schmidt,
Senior Lead Architect, HolidayCheck*

HolidayCheck uses Scala and Akka for their applications, however it's difficult to build applications without relying on third-party frameworks. The HolidayCheck team has utilized Zookeeper, Storm, Tomcat/ Jersey (for REST), Neo4j, MongoDB, Kafka, Hadoop and one of Christopher's own frameworks - neo4j-scala.

The passport to productivity

With time-to-market a driving factor for Christopher, it was a huge risk moving the company from PHP/MySQL to Scala/JavaScript/NoSQL, with the worst case scenario leaving them unable to deploy new products in time. To help mitigate the risk, Christopher engaged Typesafe to perform several Scala and Akka classes for his developers in Lausanne and London in order to get them up to speed quickly.

HolidayCheck has about 50 developers, divided into 9 teams, using mostly Scrum or Kanban (both with the release train concept). The average development project length is around 2-3 months.

The team has been using Scala and Akka for around fifteen months now. As previously mentioned, they came from a LAMP stack - not all developers that could develop there, are able to develop in Scala with Akka - at least until now. Interestingly enough, good JavaScript developers have fewer issues picking up

with Functional Programming and Scala than PHP developers. Functional Programming and Scala are complex environments and one thing that Christopher would change if given the chance to do things again would be ensuring that the teams are more evenly matched from a Scala experience standpoint, as the more advanced team members were using features that the less experienced did not yet know how to use properly. After the fifteen-month development effort the team is beginning to realize an ROI. Looking at the company as a whole however, most projects with higher skilled developers already show the advantages of Scala/Akka or functional programming in general.

Areas for improvement

Christopher mentions that he and his team clearly underestimated the effort needed to go from Scala 2.9 to Scala 2.10 - binary compatibility issues dictated that the frameworks and libraries he utilized needed to be migrated to Scala 2.10 at the same time which took more time.

Version fragility is a major concern for Typesafe and is something that we're attacking from multiple angles. Scala 2.10.x has been fully backwards and forwards binary compatible, so you can mix and match libraries compiled for any Scala 2.10.x version and run them on Scala 2.10.y. This a major improvement over older (pre-Typesafe) Scala releases, which didn't offer any kind of compatibility, essentially.

In Scala 2.11.x, we've focused on reducing the standard library (by modularizing - scala-library-jar shed about 1 MB, or 20%) so that there's a smaller surface area that needs to be kept stable. In addition, Typesafe will build as many open-source projects and libraries as possible to ensure that compatibility issues are minimized and remedied prior to the final release.

Eventually, we'd like to offer binary backwards compatibility between major releases for the core of the standard library, but we're not there yet.

*Adriaan Moors
Scala Tech Lead, Typesafe*

Christopher notes that Scala 2.10's extensive feature set is a double-edged sword - developers tend to try to utilize as many of these features as possible which leads to multiple ways that problems can be solved. Engineering managers need to pay attention to the functionality being utilized and reign in their teams to produce consistent code just the same as with other languages.

Conclusion

Christopher notes a number of points gleaned from his use with Scala and Akka:

- Good developers want to code Scala, which is very good for recruiting.
- Scala is an enabler for great frameworks using DSLs.
- The collection API is the one of Scala's most valuable capabilities.
- You need skilled developers, as Scala is a sophisticated language.

- The possibilities of Scala separate good and bad developers more clearly.
- The Scala world sometimes tends to reinvent wheels.
- Scala/Java interoperability is not an issue and works seamlessly.
- CoffeeScript and Scala are a good fit.

HolidayCheck doesn't utilize advanced functional patterns and are very productive using the more basic functionality. Christopher is very happy with his decision to use the Typesafe Reactive Platform.

I am sure that without Typesafe, we would not have the confidence to use Scala and Akka in our business or platform. Good to know that there is someone who can help!

*Christopher Schmidt,
Senior Lead Architect, HolidayCheck*

*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**.*