



Bringing Reactive Applications to the Java Virtual Machine

Yap.TV Goes Reactive

Transitioning from Ruby to the Typesafe Reactive Platform, Yap.TV manages increases in traffic and new problem domains.

About Yap.TV

[Yap.TV](#) is your social guide to what's on TV and iTunes. See what's new on TV, browse top shows, discover what your Facebook friends are watching, and get your fix with a personalized TV Feed full of media from your favorite shows.

About SoftwareMill

[SoftwareMill](#), Typesafe Consulting partner, is a software house based in Poland that specializes in delivering high-quality, customized software solutions. Their clients come from the telecommunications, banking, logistics and entertainment industries. A wide range of systems can be found in their [portfolio](#), starting from high-performance mobile application backends, through fault-tolerant messaging gateways, big-data reporting solutions and credit management systems.

The Challenge

SoftwareMill and Yap.TV began working together in September 2011. At the time, the Yap.TV server was entirely Ruby-based. While this worked in the beginning, as the user base grew, high load on the servers, especially during prime time TV, caused significant issues with server performance and scalability.

Not only was this degrading the end-user experience, but as new features were waiting in the backlog to be developed, key engineering resource time was becoming dominated by scaling servers and operations.

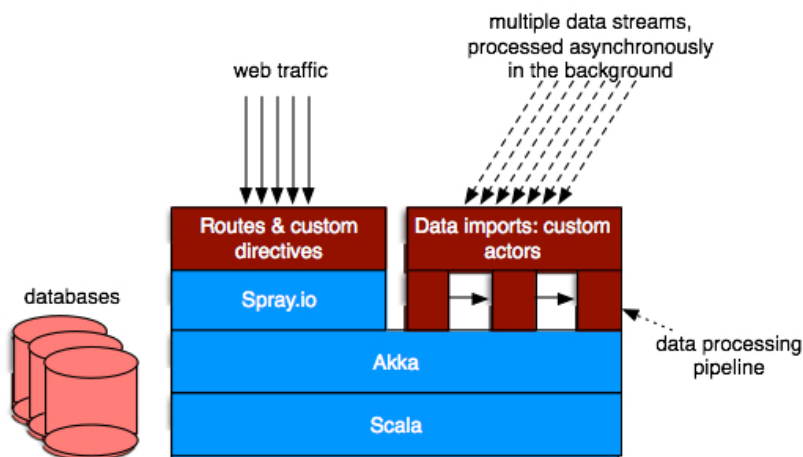
Apart from serving traditional web traffic, Yap.TV also required an increasing number of background imports, taking together data from various sources. To handle the ever-growing amount of data to be processed, a performant (but at the same time - easy to program and safe) concurrent programming model was crucial.

The Solution

Yap.TV decided to re-engineer their backend systems with the help of software development from SoftwareMill. They formed a joint backend team to tackle the problems. The platform of choice was the Typesafe Reactive Platform. We evaluated a couple of alternatives, but the elegance and type-safety of Scala, the concurrency (actor) model used by Akka, the speed and power of Spray.IO in building REST services and the overall asynchronous model of the Typesafe Platform made it a clear winner.

To ensure quick delivery, SoftwareMill gradually replaced portions of Yap's REST API, starting from some simple endpoints to get deployment right, and then proceeding to the ones with the highest congestion. They promptly found that Spray.io, combined with the Scala language, enabled them to rapidly develop new web service endpoints. Type-safety saved the SoftwareMill team from a lot of debugging and the ability to create composable abstractions kept their code concise and easy to maintain.

To implement the data processing pipeline, SoftwareMill used the Akka concurrent programming framework. Handling multiple, asynchronous data streams and processing them in real-time became not only possible, but also feasible to implement in a reasonable timeframe and a clear, understandable way. The Actor programming model itself, and the fact that instead of thinking about threads and synchronization, one has to think about actor behaviors and interactions, which happen asynchronously, made programming this system much easier.



Importing data from third-party services (which in case of Yap.TV, include Twitter, Facebook, EPG and iTunes data, among others) is an inherently error-prone process, hence detecting problems early and recovering from them is crucial. Here again Akka was a good fit, due to the supervisor hierarchies and the let-it-crash philosophy.

Thanks to the versatility of the Scala platform, SoftwareMill was able to use and easily integrate with Java-based systems, such as ElasticSearch for implementing TV show search and Mahout for an initial version of our recommendation system (which they ultimately replaced with a custom-written pure-Scala solution).

At each point of the development process, we kept the quality standards high. All of the code was peer-reviewed, not only to find bugs and design problems as early as possible, but also to disseminate the knowledge about the code, hence reducing code ownership. All of SoftwareMill's code is also unit- and (where this makes sense) integration-tested.

The Scala ecosystem encourages a simpler deployment process, without the need to maintain application servers or such, which are commonplace when using e.g. Java. All of our services are packaged as fat-jars, and run as first-class daemons. Our automated Chef-based deployment infrastructure takes care of distributing new code to the servers on which it should be run. Thanks to that we can deploy multiple times per day, and the users will only notice the improved feature set, no downtime!

The Result

SoftwareMill's helped Yap.TV develop a high-performance, resilient and scalable backend. With the help of the Typesafe Reactive Platform (Scala, Akka, Spray.io) Yap.TV now has a solid server platform that is ready to accommodate not only increases in traffic, but also brand new problem domains.

“When selecting a partner to help us re-architect our backend infrastructure, we wanted to work with a team and a platform that share our high quality of standards. SoftwareMill and the Typesafe Platform enabled us to do just that, while building out a new solution that allowed us to scale and grow in new directions.

Trevor Stout, CEO at YAP.TV.

*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**.*