



Bringing Reactive Applications to the Java Virtual Machine

Huffington Post's new Publishing Platform on Typesafe

Huffington Post's amazing popularity spawned the need to make numerous technology decisions rapidly, resulting in a patchwork publishing and content management system that recently turned eight years old. A new, integrated platform was needed to propel them through the next stage of growth.

About Huffington Post

The Huffington Post is a leading online news source founded by Arianna Huffington, Ken Lerer, and Jonah Peretti. Contributors include more than 3,000 bloggers - from politicians and celebrities to academics and policy experts - who contribute in real-time on a wide range of topics.

The site started as a politics blog, but now covers more than 20 categories, including media, business, entertainment, living, style, sustainable "green" living, world news, technology, nonprofits, college life, and comedy. It began launching local versions in 2008 and increasingly is experimenting with social networking features.

In February 2011, Huffington Post was acquired by AOL.

The Business Problem

The technological landscape has changed radically since Huffington Post's blogging and publishing platform was forked from the original Movable Type codebase, with entire technology paradigms falling in and out of favor with the digerati.

"Eight years of growth allowed our company to accumulate a lot of technical debt and we were really paying the price," explains Adam Denenberg, Huffington Post's Vice President of Engineering. "Our publishing platform was incredibly fragile. The infrastructure was very tightly coupled; things were hard coded all over the place. And it was at the point where it was really preventing us from being agile and developing more quickly."

Adam decided that it was time for a change. His goal: Getting Huffington Post ready for the next wave of users who create thousands of hours of video, billions of Likes, and millions of tweets each day. This required a modernized codebase, built upon the success of the past, yet preparing Huffington Post for a future where mobile devices are more powerful than desktop computers, social media is the only media, and everyone with an Internet connection can stream content 24/7.

Taking Stock

Before embarking on radical change, Adam took time to evaluate the core problems with the current systems, and broke them into three main categories:

Tools

The tooling used by The Huffington Post's editors consisted of a number of involved steps. One example of this pervasive issue was with the Slide Show Creator. Slide shows are excellent at generating revenue, however, creating one requires going into different tools, copying and pasting special codes from multiple screens onto new screens, which sometimes doesn't work correctly and other times requires republishing an entire entry.

Clearly new tools were needed to allow editors to operate more seamlessly and efficiently than the current system allowed.

Scalability

Huffington Post relies very heavily on caching. They needed a platform that could sustain massive amounts of traffic and support their planned global deployments. With editions for Italy, Spain, Quebec, Japan and Germany – the pressure was on to spin-up multiple, international data centers, on a massive scale.

Ease of Deployment

Multi-tenancy factors in, too. The legacy platform consisted of different instances of the Huffington Post that were being launched internationally, however, they are all part of a monolithic code base. What Huffington Post actually needed was a multi-tenant view of the world so that all the international editions could have their own instance of the Huffington Post that was isolated from a security, infrastructure, security and management standpoint.

Picking the right Platform

Adam's responsibilities cover HuffingtonPost.com, all of the international editions, Huff Post Live, advertising platforms, moderation and community platforms, editorial systems, and real-time statistics and machine learning technology. Whatever technology platform he picked would need to bring synergy across the product set.

Since Huffington Post is a member of the AOL family, Adam looked at what other business units were using. A notable standout was the Patch group; they had done a large rewrite from Ruby to Scala and Play Web Framework from Typesafe and were very happy. 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. Play Framework is an MVC-based web framework that focuses on developer productivity, modern web and mobile applications, and predictable, minimal resource consumption resulting in highly performant, highly scalable applications.

Fundamentally, the biggest issue is scalability, and with two and a half billion page views a month Huffington Post gets a lot of traffic. At any given time they have at least 150,000 concurrent users, a commenting platform, all these services, and they rely heavily on caching. A key aspect of scalability is efficient resource utilization, and with the very large vertical boxes with lots of CPU's and RAM, Adam and his team felt that they would probably not take advantage of all of them using PHP.

"We went through a lot of discussions," recalls Adam. "We said, 'Okay, look, we could probably develop a lot quicker with other languages and pay a performance penalty. But what languages and frameworks are out there that could give us the scale that we need, but, also allow us to develop somewhat rapidly?'"

The team investigated Erlang, but it didn't feel like it was mature enough for building Web platforms, so they started talking about the JVM and the discussion came back around to Scala and Play.

With a lot of the big media companies like Tumblr, Foursquare, Twitter and Gilt Groupe moving to Scala, there was clearly a precedent being set for similar media companies that were achieving really great results with the technology.

A big factor in the decision to use Scala and Play, however was the value that Typesafe brought to the table with commercial support and training from the original authors and architects of the products they would be using.

"Scala was new to the group, so having commercial support and training was a big benefit for Huffington Post," shared Adam. "We've had really good success getting quick resolutions to questions. Plus, when we wanted some help with Slick, a Typesafe consultant came out for four hours. We got ten people in the room and got everyone up to speed. That's the kind of responsiveness and engagement that we really like."

The Team

With a small team of five people divided equally between the front and back end components, productivity is key, as is the ability to leverage available frameworks like AngularJS and Scalatra which aid in front end development and creation of back-end services and API's respectively.

The team engaged with Typesafe for Scala training, which was very helpful to bootstrap the project, and are looking for a longer-term mentor who can provide best practices and approaches to solving problems over the long term.

Moving forward with Athena

The new Huffington Post Platform was dubbed Athena, and is composed of Scala, Play, AngularJS and MongoDB.

Adam and his team took a smart approach to building out their new architecture, with the most visible benefit being that the internal editors at Huffington Post already are utilizing Athena. With a front-end built in Play, the team has created a tool that still speaks to the original backend services, however the front-end experience is radically simple and far easier to use than the mish-mash of existing tools.

John Pavley, Huffington Post's CTO mentioned a specific use case that the Athena system is solving:

“For example, two HuffPost editors want to collaborate on a story in real time. They don't want to email a text document around and they don't want to use Google Docs to collaborate because it takes time and tweaking to move a body of text from one system to another.

Two editors John and Stephen both log into the Athena System. John opens up an article and starts editing it, while Stephen opens the same article. John is informed by a message in the lower right of his screen that Stephen has started to work on the same article. As Stephen edits the article in his web browser, John is given the opportunity to "fight" with Stephen for control. John wisely chooses to allow Stephen to continue writing and sees the changes Stephen has made to the article in real-time. The sweet thing about all this real-time collaboration is that the developer just has to set it up - the AngularJS framework (and Scala) are doing all the hard work. AngularJS monitors the text field in Stephen's web browser, notifies the Scala code back on the server of a change, and the Scala code notifies the text field in John's web browser. The amount of MySQL, PHP, HTML, and JavaScript code required to make this real-time collaboration happen would have been staggering and hard to reuse. Athena's modern technology stack with Scala, MongoDB, and AngularJS, makes it simpler for the developer and easier for the editors: No need to "check out" documents, no arbitrary limits on the number of collaborators, minimal locking of the shared document, no submit button required”.

John adds that “this is a much more human way doing things for everyone involved.”

This approach of building a veneer API over existing services allows a gradual migration from the Legacy Publishing Platform, which enables the team to build out backend services and substitute them over time. These services include dozens of editorial components for cropping images, creating slide shows, creating splashes, etc. that are simply not feasible to deploy all at once.

*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**.*