



Bringing Reactive Applications to the Java Virtual Machine

2.2 Billion Clicks and Beyond: How Hootsuite Modernized Its URL Shortener to Scala / Akka Using the Typesafe Reactive Platform

About Hootsuite

Hootsuite is the world's most widely used social media relationship platform. More than 10 million users, including 744 of the Fortune 1000 companies, trust Hootsuite to manage their social media programs across multiple social networks from one integrated dashboard.

The Challenge

[Ow.ly](#) is Hootsuite's popular URL shortening and click analytics service. Through Hootsuite's web and mobile dashboards, users submit long URLs to Ow.ly in exchange for a trackable, shortened URL for social sharing. To date, Ow.ly has generated several billion shortened URLs and handles several billion clicks per month (thousands of clicks per second during peak usage).

The Ow.ly service is a critical and popular part of the Hootsuite product. As Hootsuite's user base grew, the engineering team found themselves sinking more and more time, effort, and cost into keeping the service performant and reliable.

"Ow.ly's popularity exceeded our expectations," said Beier Cai, Director of Engineering - Platform. "As a result we have had to focus a lot of time, effort, and cost into scaling it. We have seen diminishing returns for the work we put into scaling this service on its current technology stack. Our engineering challenge is to address that scalability issue so Ow.ly continues to be a performant and reliable service; and one on which we can build new functionality."

The root of this scaling challenge is Ow.ly's six-year-old codebase built on PHP and MySQL. PHP processes requests serially, which creates a bottleneck around Ow.ly's performance when the system is making external calls to screen submitted URLs and intensively writing shortened URLs to the database. PHP's limitations for parallel processing required Hootsuite to leverage the Gearman framework for asynchronous recording and aggregating click statistics. In addition to wrestling with PHP's concurrency limitations, Hootsuite's maintenance efforts were hampered by insufficient automated test coverage and a labor-intensive deployment pipeline.

"Our Ow.ly hosting costs also increased faster than we expected," explained Geordie Henderson, VP Engineering, "since our scaling efforts were mostly focused around adding more horsepower, and not as much on refactoring the product to achieve efficiency. Dozens of AWS instances were required to handle peak loads when we knew that a refactored solution could significantly reduce the number of servers required to deliver Ow.ly functionality."

The Ow.ly engineering team's challenge: to modernize this critical part of the platform at a reasonable cost, while maintaining 100% service uptime.

The Solution

Hootsuite turned to Typesafe's Play Framework to refactor and properly scale Ow.ly. Play is a Scala based framework. As a compiled language, Scala required fewer resources than PHP to perform the same work. The event-based architecture that Hootsuite created with Play allowed blocking external calls to scale better: the system no longer needs a whole new OS process for each incoming web request. Moreover, where PHP's lack of concurrency constructs made it difficult to perform parallel operations, Scala and Akka made concurrency trivial.

Today, 100% of Hootsuite's dashboard usage of the Ow.ly API is handled by this new architecture:

- Ow.ly Web Service - a cluster of Play Framework nodes handling the Ow.ly API, the web UI, and URL redirect.
- Ow.ly Data Service - a cluster of Akka microkernel nodes wrapping the legacy MySQL database in a service. Ow.ly web nodes' communication with the Ow.ly data nodes uses a custom protocol based on ØMQ. Wrapping the database in a service will allow Hootsuite to seamlessly upgrade the Ow.ly data store in the future.

Hootsuite is working to replace its Ow.ly analytics system with a Scala and Akka solution, which will experience the same benefits the Ow.ly URL shortener service sees thanks to the Play Framework.

With Play powering the Ow.ly service re-write, the Hootsuite engineering team was able to reduce costs, more effectively handle load, scale more efficiently, update their testing and deployment infrastructure, and most importantly, improve performance and reliability for Ow.ly users.

- **Cost Savings** - the Hootsuite team was able to reduce the number of servers by 80%, resulting in major savings on its AWS bill and lower IT Operations complexity.
- **More Requests Per Second** - Hootsuite load-tested Ow.ly at an order of magnitude more requests per second than the current system.
- **Better Testing Layers** - The new system currently has 90% unit test coverage and "being able to easily add layers of unit, functional and integration testing has been a pleasure," said Mike White, Senior Engineer.
- **Scale with Ease** - With Scala performing the same work as PHP but with far less effort, the cost and required work to scale is significantly reduced.
- **Easier / More Enjoyable Development** - Hootsuite found Play to be a very easy-to-use framework for development. "It was quick to get people using it, it promotes writing non-blocking, Reactive-style code, and as a result your apps get more efficient," said Jim Riecken, Senior Engineer. Additionally, RESTful APIs are a first class citizen in Play, and it is simple to integrate 3rd party tools like automated API documentation framework [Swagger](#).

As software companies grow by orders of magnitude, the technologies that drive legacy products need to be replaced to keep scaling costs in line. Hootsuite's modernization of the Ow.ly URL shortening service using the Play Framework is such an example. The Typesafe Reactive Platform is now being contemplated for several future services because it is a solid choice for the scale Hootsuite is today.

*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**.*