



Bringing Reactive Applications to the Java Virtual Machine

Sharethrough chooses Scala to build industry's first native ad management system

Sharethrough is building a native advertising platform based on the belief that traditional display advertising is underperforming and will be replaced by advertising formats that are non-interruptive, with seamless visual integration on the websites on which they appear. They chose Scala and the Typesafe Platform to help meet their functionality and performance objectives.

About Sharethrough

Sharethrough's technology platform powers in-feed "native" ad placements, enabling publishers to manage and sell native ads on web and mobile. Sharethrough believes the future of advertising will be about thoughtfully integrating brand content on sites people love to visit, not slapping ads in cookie-cutter boxes, banners or auto-playing videos.

The Sharethrough team is comprised of a dedicated group of professionals with deep experience across the advertising, technology, media and creative industries. Founded in the heart of Silicon Valley at Stanford University, Sharethrough now spans the country with offices in San Francisco, New York and Chicago.

Background

Prior to 2012, the first iteration of Sharethrough's advertising platform relied on an integration between their end-to-end campaign trafficking and analytics platform (a monolithic Ruby on Rails application) and a third-party ad server. In order to realize their vision of powering native advertising across the open web, they realized they were going to have to build their own ad server - no existing solution supported the targeting, templating and analytics capabilities they required.

Building on three years of experience powering video advertising on hundreds of websites, Sharethrough knew what they would need in terms of functionality, performance and monitoring. Being long time practitioners of the Lean Startup methodology, they started small and focused on building early pieces of the core templating and targeting technology as a simple, horizontally scalable trafficking tool in Sinatra, deployed on Heroku. As the platform quickly gained traction, they realized that a move from Sinatra (and

Heroku) was going to happen sooner than originally planned. Given that there would be significant performance requirements and that the critical portion of their infrastructure was going to be in the service layer, Sharethrough decided that the time was right to examine other languages and frameworks.

Subsequently in 2012, Sharethrough began evaluating languages and architectures for their next-generation service stack.

Considerations

Early on in development, the objectives were primarily around performance, stability and extensibility. Low latency is required for happy users and the lofty goal of shooting for 99% of transactions returning in under 50ms was set. Sharethrough needed their algorithms to be efficient, scaling at worst linearly with the quantity of content in the system, facilitating low latency and reducing infrastructure costs. Starting small and having the ability to scale meant choosing proven and battle-tested technologies; nobody wants to be first to jump onto a brand new platform only to see them fail at scale and having to figure out what went wrong. With a new product in a new advertising ecosystem, Sharethrough values the ability to quickly capture learnings, rapidly deploying new algorithms as they find out what works for partners, customers and users. Finally, as Sharethrough grows their engineering team, they aim to ensure that no team is blocked waiting for another. They do this with a comprehensive test suite, which limits the need for a dedicated QA team and a service-oriented architecture, which isolates each unit of work and allows them to parallelize efforts.

With these things in mind, the decision was made early that the next generation architecture would be built on the JVM as it offers the most mature, robust and performant platform for building production applications; JRuby, Scala, and Java all were candidates. The following considerations apply to these languages for Sharethrough's particular set of circumstances, and are not universal, applies-in-all-situations comparisons.

JRuby: Sharethrough has long been a Ruby shop so the first language evaluated was JRuby. JRuby would allow existing Ruby developers to quickly develop and deploy a new platform bringing along existing development practices and test frameworks. Sharethrough concluded that prior familiarity was outweighed by a number of issues; first, JRuby felt split between the worlds of Java and Ruby, notably with dependency management being isolated in Bundler for Ruby and Maven for Java – also an issue for library selection (do you use pure Java or pure Ruby libraries?). Working with Ruby didn't foster thinking in a thread-oriented mindset for the team, and would require them to make certain that the gems they depended on were also thread-safe. Sharethrough continues to use Rails for integral portions of their stack, just not the service layer.

Java: Sharethrough are using a significant amount of Cascading (and therefore Java) for their Amazon Elastic MapReduce jobs and have long felt the pain of the amount of boilerplate required and felt it impacted their ability to iterate. Given the importance of the algorithmic work Sharethrough is doing, using a language that naturally facilitates that style of thinking is important (i.e. functional) and they prefer to use Java for object-oriented modeling. Ultimately Scala's support of both functional and object-oriented paradigms won out.

Scala: Sharethrough's first exposure to Scala came as multiple facets of their data pipeline in both the authoring of Hive UDFs and in the use of Twitter's Scalding for creating Hadoop jobs. Sharethrough developers were quickly productive with Scala, which can be attributed to their familiarity with Java, Scala's expressiveness and lack of boilerplate code. Scala's seamless support of Java puts more than 15 years of mature Java libraries at their disposal (perhaps most importantly for Sharethrough, `java.util.concurrent`).

The concise syntax and blend of object oriented and functional paradigms allows them to use one language for two very different purposes: object-oriented and mathematical modeling.

Architecture

A detailed discussion of Sharethrough's architecture is beyond the scope of this case study, however some important choices as they relate to the choice of Scala are within scope. Sharethrough decided they would focus on extracting small, single concern services. Twitter's Finagle and Typesafe's Play and Akka Frameworks were the perfect choices for this architecture. The front-end ad server was built using Finagle; the asynchronous nature of Finagle coupled with the level of abstraction made it the ideal tool for accepting incoming impression requests and fanning out calls to the necessary backend services. Backend services are built using a combination of Finagle and Play. Play provides an ideal level of abstraction for more complex services that have to expose multiple RESTful resources. Using Akka for background processes gave them the ideal tool for writing services such as the automated creative optimizer.

Conclusion

After evaluating many JVM based languages Sharethrough has found Scala to be the best tool for the requirements of their next generation native advertising platform. Utilizing the strengths of frameworks built on Scala such as Akka, Play and Finagle, they have delivered a service-oriented architecture capable of serving thousands of requests per second, with 99th percentile latency of less than 50ms.

Sharethrough highly recommends consideration of Scala if you're requirements dictate a language that delivers reliable performance and rapid development.

*The **Typesafe 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 Platform through the **Typesafe Subscription**.*