



Bringing Reactive Applications to the Java Virtual Machine

Kifi is built from the ground up with the Typesafe Reactive Platform

Leveraging Scala, Play and Akka, powerful search engine executes as many as 50 new code deployments per day in a highly aggressive continuous deployment environment

About Kifi

Kifi is an innovative Silicon Valley startup that is creating a new way to search the Internet and store or share information. Kifi allows you to easily keep and tag anything you find online – an article, video, picture, email, – then quickly find it on top of your favorite search engine results. Kifi also surfaces at the top of results relevant that your friends kept, as well. Kifi layers social tools on top of what you keep by including messaging to friends and highlighting and annotation of information found on pages. The Kifi search infrastructure also lets you tag and organize information you are saving, just like in a bookmarking system.

The Challenge

Kifi CTO and co-founder Eishay Smith knew that he was undertaking a significant challenge in building an application that would attempt to perform multiple computationally intensive tasks in an extremely limited time window. What's more, the architecture had to support this level of performance at all scales and be flexible enough to scale up and scale down instantly to meet often unpredictable customer usage patterns. Therefore, the Kifi team built a Reactive application structure to enable rapid iteration, handle massive number of simultaneous users and processes, and maintain a diverse range of services. In short, Kifi needed:

- **Blazing fast response times** throughout the application tiers: People are used to fast search query responses and page loads with traditional search engines like Google. Kifi needed to build an application that could match the same speed and responsiveness. "Our search application is nontrivial. It needs to look at several indices per request and get some auxiliary data from other resources like memcached, databases, etc. To beat Google's response time, we need to target a response in only 30-40 milliseconds," explains Kifi CTO Eishay Smith.
- **Robust support for multithreading** to maintain performance: Kifi needed high concurrency support baked into its application infrastructure. It splits work on several cores on its virtual machines for the same request, to be able to hit the optimal response window.

- **Support for rapid continuous deployments:** Kifi also needed an application infrastructure that could handle a rapidly iterative development cycle and continuous deployments. This meant dozens of new deploys to the complex application stack daily.
- **100% application availability:** Because Kifi plans to become a critical part of people's lives, the company needed to achieve near zero downtime and 100% resilience if newly deployed code caused a server fault.
- **A strong SOA orientation** in the underlying software languages and frameworks: Kifi needed to be built as a true Service Oriented Architecture, so a software language that handled reactive, non-blocking service communication would be critical.

“We need the ability to run Web applications that change very frequently and need to be very flexible in a true Service-Oriented-Architecture. Typesafe’s Play framework and its interaction with Akka handles this beautifully and gives us a very Reactive system. Since we are data heavy application and we have a lot of different services and data types, we also rely heavily on multi-threading that is very well integrated into Scala. Support for things like WebSockets makes it much easier for us to maintain performance and a broad set of features and services in a small number of virtual machines.”

Eishay Smith, CTO and co-founder - Kifi

The Solution

The original engineers building Kifi had previously worked with Scala-based application architectures, and knew the Typesafe stack would meet these requirements. For the core application, Scala and Play are used for all web services. “Our AngularJS website, browser extensions, and mobile clients call our Play API, speaking JSON,” says Stephen Kemmerling, a Kifi engineer who has worked extensively with Play building the messaging service. Kifi also uses the native WebSockets capability in Play to enable low-latency, two-way client communication such as for real-time messaging and updates to clients in browser extensions, iOS, and Android.

On the backend Kifi is built using several MySQL databases with a multi-caching layer in between the primary database and the actual application using Memcached and a custom in-memory cache. The cache was built to improve response times and run more of the application and tasks in memory. Yasuhiro Matsuda, based on experience as the Chief Architect of LinkedIn distributed search, led development of Kifi Search using Lucene components to adaptively return highly personal results. Additionally, Kifi has a custom Graph engine, built completely in Scala, to generate user recommendations.

For distributed work and task scheduling, Kifi uses Typesafe’s Akka. “Our scraping service downloads every page that someone keeps so we can index and search with it real-time,” explains Raymond Ng, an engineer working on Kifi. “We also use Akka’s actors for work such as syncing social network updates, system checks, and link imports.” Coordination of running services and service topology is managed using Zookeeper, a service discovery and management program that is commonly used in service oriented architecture applications.

Additionally, engineers at Kifi have fostered a strong Scala and Play community. The Kifi Engineering blog frequently posts about machine learning, Scala, Play, and Slick. Andrew Conner, who has built applications using Play for over two years, started the SF Bay Area Play Framework Meetup group last year. It has attracted high interest from local Scala/Play enthusiasts and people interested in building modern web applications. “We’ve been pleasantly surprised with the community response to the meetup group. More and more companies are embracing quick development and reactive architectures, and Play meets the needs of small startups to large enterprises exceptionally,” Andrew added.

The Results

Kifi launched its app and has thus far experienced excellent performance and continued rapid development. The team's design for a Reactive infrastructure leveraging Scala, Play and Akka has allowed them to perform dozens of deployments per day in a highly aggressive continuous deployment environment. With the Typesafe Scala components Kifi has:

- Hit their internal targets for application response times and delivered a 99%-ile search response of less than 40ms
- Leveraged multi-threading on servers to handle heavy computation more efficiently and across fewer servers at a lower cost
- Enacted a rapid continuous deployment iteration process with as many as five or six new code deployments per hour, even with a small team
- Experienced no unplanned service interruptions even as the application user base has scaled rapidly
- Expanded its team size rapidly, bringing new engineers to the Typesafe ecosystem

“We built Kifi from the ground up as a Reactive application and the Typesafe tools gave us the best combination of scalability, agility and reliability. We also got the added benefit of a great recruiting tool because great engineers want to learn Scala.”

Eishay Smith, CTO and co-founder - Kifi

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**.