Typesafe

Bringing Reactive Applications
to the Java Virtual Machine

Code Size is down, Performance is up: Why Datazenit went Reactive

Datazenit swapped out Python/Django back end for Scala and Play Framework for a modern, Reactive architecture

About Datazenit

Datazenit is a web-based software for database management and administration. Its abilities go beyond the standard tools for database management, and include data visualizations, an intelligent schema builder and many other features that ease and enhance database management. Datazenit is designed to make databases accessible, safe and easy to use. It can be set up on a server or any desktop that has Java runtime. For more information, please visit: http://datazenit.com/

The Challenge

When Datazenit was first conceptualized, it was only supposed to run in the cloud, and a self-hosted, local option was out of picture. It was written in Python and used several Python packages -- most notably, Django framework and SQLAlchemy -- for its database inspection capabilities. The project also used SQLParse, a Python library for parsing SQL queries.

While Django was great for building a quick prototype, handling user authentication and basic functionality, performance became an early challenge. Lauris Dziļums, Founder and CTO at Datazenit, was frustrated by the overhead of Django framework and always memory-hungry SQLAlchemy. The backend of Datazenit had to be as performant as possible because it must run on low resource servers and desktop computers and be able to handle any dataset the user threw at it.

"We couldn't demand high performance servers from our customers or consume all available RAM without a real reason," said Lauris. " Luckily Scala had no problem with that."

After many internal discussions and feedback from potential customers stakeholders, Lauris and his team determined that a self-hosted option had become a must-have to remain competitive, and they needed a framework solution that could get them there.

Lauris was looking for a solution that enabled:

- **Portability (Write once, run anywhere):** Datazenit needed to be a cross-platform software that could run in any environment.
- **Functional language:** Lauris' team was looking for a language that let them write side effect free, type-safe and concise code with an easy to achieve concurrency.
- **Reactive performance:** Datazenit needed a platform that was resilient, responsive, scalable and event-driven.

The Solution

This is where Scala came along. It compiles down to Java bytecode, which is a compile once, run anywhere type of thing. Also Scala's functional programming features really helped with transforming and processing data structures, which is the core behind Datazenit backend.

Lauris chose Play framework, because it gave an impression of a mature, stable and lean framework. Lauris had long reviewed it for personal projects, and this was the opportunity to use it in a real world application. It was easy to set it up and get running. There were some hiccups and learning curve in edge cases, but the solution in the end was always elegant and performant Scala code. "The code made me happy as a developer and it clicked right away," said Lauris. "I was able to develop faster and be more productive with features that matters the most." Lauris found Play was just as quick as Django for a rapid quick start, and it was pleasant to build the core of Datazenit on top of it.

Because there were many excellent open source libraries, tools and frameworks in Scala, that enabled Datazenit to rapidly catch up, the backend was completely rewritten in just a few weeks. The second biggest contribution to the core of Datazenit came from a library called ScalikeJDBC. It is an abstraction over JDBC, which they use for database connections, meta data processing and query inspection.

"Great software takes time and the right tools for the job. We've found the Typesafe Reactive Platform to be the best tool for our architecture needs."
Lauris Dzilums, System Architect, Founder and CTO, Datazenit

The Results

The end result of switching to the Typesafe Reactive Platform was impressive; code size was down, the performance was up and Datazenit already has a binary which works on almost every platform out there. The backend is more modular than before and it is definitely less prone to bugs, because we mostly catch the bugs right away, not when running in production.

*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.***

info@typesafe.com