Typesafe

Bringing Reactive Applications to the Java Virtual Machine

Typesafe aids CCAD in managing cable operations

Scaling developer productivity, improving code quality and removing locking and contention were reasons that influenced CCAD's decision to move away from the traditional Java-based stack they were using to look for more sustainable alternatives. The Typesafe Reactive Platform assisted CCAD in addressing these issues, and enabled process changes elsewhere in the organization that improved overall product quality too.

About CCAD

CCAD focuses on the development of current and future conditional access (CA) technologies and supporting control system infrastructure, which is deployed among major operators worldwide. Conditional access technologies enable operators to distribute and safeguard video content supplied by the numerous video program providers.

The Business Problem

Ensuring that subscribers have access to only the audio/video content they have purchased or subscribed to is an ongoing challenge faced by cable companies worldwide. The problem is further compounded as cable companies offer various promotional bundling packages from time-to-time that changes consumer entitlements on a regular basis.

CCAD has solved this problem in an interesting way with a product suite of several different products, the largest one named CASMR. This product set - a combination of both hardware and software - is utilized by cable companies worldwide to solve the problems with managing entitlements, conditional access and integration with billing systems. It's important to point out that these are predominantly "lights-out" systems that actually run the cable company and are used by employees who are trying to effect some change in the system.

Subscribers today have more choices therefore forcing providers to constantly adjust content offerings and prices. In addition, technology and consolidation trends have enabled service providers to significantly grow their subscriber base. The subscriber experience must be flawless from both a timing and accuracy perspective regardless of subscriber population.

info@typesafe.com

Revision 1.2

Changing the Channel

CCAD's initial video control system dealt with both individual transactions to set-top boxes, as well as various transactions to back-end servers. The system was highly concurrent, and it was done entirely in Java. As a result there was a lot of boilerplate code, developers would spend time creating their own immutable domain models, and generally "swimming upstream" against the Java ecosystem.

CCAD also had their own internal asynchronous state machine library, which was prone to dead-locking or live-locking. The library itself was fine (it had been built on java.util.concurrent), but its callback based design led to deadlock situations - a common problem seen when attempting to build out massively concurrent systems in Java.

The deadlocking issues were certainly impeding productivity and innovation and as a result in 2009, CCAD started using the Scala programming language. 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. Specifically, CCAD utilized Scala's standard library Actors for concurrency. Since the application in question - CASMR - ran in a single JVM instance, it wasn't possible to do a wholesale migration to Scala, so the CCAD team started implementing individual, highly concurrent modules in Scala Actors, with a simple Java façade around them for interoperability. CCAD utilizes OSGi extensively within their organization, which allows them to run their Scala-based applications on a micro-service architecture.

[Scala] makes development noticeably faster and makes me feel more confident that the end result will be correct.

Michael Smith
Senior Software Engineer, CCAD

This led to the CCAD team's increased productivity, and reduced the deadlocking and livelocking issues significantly, and as time progressed, Scala's use became broader. Until this point, Scala had been used with some restrictions, however CCAD started lifting some of those restrictions (i.e. only in small modules) so that engineers could use it anywhere, with the one caveat that the APIs of bundles still needed to remain in Java. That caveat was short lived as the performance gains and the productivity gains were just too hard to ignore. Today, all engineering work is done with Scala.

Within CCAD, Akka and Spray, also parts of the Typesafe Reactive Platform, complement Scala. Akka is a toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications. Spray is a toolkit for building REST/HTTP-based integration layers on top of Scala and Akka. Being asynchronous, actor-based, fast, lightweight and modular Spray is a great way to connect Scala applications to the outside world.

The move to Akka came about when the team felt that the technology was ready for prime time; indeed the Actor model utilized in Scala has been replaced by Akka's implementation due to its technical superiority, inherent supervisor model, and substantial performance gains.

Spray has been chosen recently and is replacing Spring MVC components over time. The front end of the existing applications is written in Adobe Flex, while new applications are written in HTML5 with AngularJS.

The quality level of what we produce with Scala, Akka, and functional programming techniques is so high. Our product marketing team and our product owner have made numerous comments about the technology stack's contribution towards both this quality level and reduced qualification and regression testing.

Michael Pilquist
Chief Software Architect, CCAD

Leveraging Typesafe and the ecosystem

Direct support from Typesafe has been essential; the most tangible instance of this that comes to mind is a strange startup issue seen recently after the team refactored some code. By having a direct line to the core engineering teams via Typesafe's support offering, "Typesafe Together", Scala Compiler Engineer Jason Zaugg was able to rapidly find a bug in the Scala Standard Library and turn around a fix in no time at all.

With some companies, we get support engineers that were hired specifically to answer questions and are not able to make changes in the products. It's been anything but that with Typesafe, something that has pleased us immensely.

Michael Pilquist
Chief Software Architect, CCAD

CCAD are proponents of test-driven development with Scala, and leverage ScalaTest extensively for automated acceptance testing. A huge deliverable and clear benefit of this approach is that the CCAD team has been able to generate PDF-based traceability reports that show downstream test teams the number of test scenarios run, what steps were performed and what the resulting output was. Indeed, this ended up being a major process change for all of the teams' downstream consumers. Pilquist adds that the team also uses shapeless (a type class and dependent type based generic programming library for Scala) and Scalaz (a Scala library for functional programming) to aid in the development effort.

*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