

Netflix Heads into the Clouds

Interview with Adrian Cockcroft

RIK FARROW



Adrian Cockcroft is the director of architecture for the Cloud Systems team at Netflix.

He is focused on availability, resilience, performance, and measurement of the Netflix cloud platform. Adrian is also well known as the author of several books while a Distinguished Engineer at Sun Microsystems: *Sun Performance and Tuning*, *Resource Management*, and *Capacity Planning for Web Services*. From 2004 to 2007 he was a founding member of eBay Research Labs. He graduated with a BSc in Applied Physics from The City University, London.

acockcroft@netflix.com

I first heard about Netflix's remarkable journey into the cloud in Adrian Cockcroft's presentation at HPTS (see the reports in this issue). Adrian explained why Netflix was moving to the cloud after having had their own datacenter: the cloud was both easier to work with and more reliable. Adrian's presentation slides [1] as well as very detailed blog entry [2] do a great job of explaining where Netflix is today, and where they plan to go. I suggest that you read both of these resources; the slides present an overview, and the blog gets into details. What is missing from these resources is a bit of history, which Adrian provides here.

Rik: I am guessing that Netflix started out using a big server in a datacenter, pretty much the way other companies had always done, and you gradually learned about the advantages of not having your own servers. Could you tell us a little about the past that led you to this point?

Adrian: The history of how we got to the cloud is that our original datacenter systems were based on a few large Oracle servers with a Java front end. The load at that time was dominated by the DVD business. We had a storage data corruption bug in the summer of 2008 which took Netflix down for several days, and we also saw that in the future we would need to rebuild our site for higher availability to support the demands of streaming. It was clear that we needed large-scale redundant datacenters, but we couldn't easily predict how much datacenter capacity we would need for the rapidly growing streaming business and where it should be located. We had also had a fairly painful experience moving from a single datacenter to a pair of small ones (leased cages), and needed to decide whether to invest heavily in the staff and skills needed to run a large and high-growth-rate datacenter infrastructure, or outsource and leverage an external cloud supplier. In parallel we ran a large upgrade of datacenter capacity and an investigation into the feasibility of using cloud. We quickly settled on AWS as the largest cloud and established an executive-level relationship as a foundation for the business and technical relationship.

Through 2009, we explored the cloud platform with several pathfinder projects and non-customer-facing workloads such as encoding and Hadoop-based log analysis. In early 2010 we brought up the first customer-facing workloads, starting with the simplest ones with fewest dependencies, and gradually filling in the data sources until almost everything is running in the cloud, but with the data resident in both cloud and datacenter. In 2011 we gradually moved the "source of truth" systems into the cloud, with copies in the datacenter as needed. The final stages of that are currently being completed.

Rik: I understand that you have built many tools for managing and monitoring your servers and software in the cloud, and that you have made them open source. Can you briefly tell us about them?

Adrian: We have built a platform that runs on the AWS cloud but provides abstractions and patterns that are more portable and convenient for the developers at Netflix. I describe this PaaS in a presentation given at QConSF in October; slides are at <http://www.slideshare.net/adrianco/global-Netflix-platform>.

The platform is primarily based on Java running in Tomcat on Linux. We also support the Groovy/Grails environment, primarily for building internal tools. We use open source components combined with custom code. The Netflix Global PaaS has the following features:

- ◆ Global distribution of traffic, processing, and data
- ◆ Localized support for multiple languages and jurisdictions
- ◆ Support for dynamic and ephemeral cloud resources
- ◆ Data migration mechanisms from datacenter to cloud and between regions
- ◆ Continuous backup and secure distributed archive of cloud-based data
- ◆ Dynamic security key management with multi-level key protection
- ◆ Fine-grained least privilege security based on AWS security groups and IAM
- ◆ Scalable to many thousands of instances, autoscaled with load

The components that we are open sourcing at github.com include:

- ◆ Curator: a distributed coordination framework based on Apache Zookeeper
- ◆ Priam: Tomcat-based automation for simple management of Apache Cassandra on AWS
- ◆ Astyanax: a Java client library for Cassandra that improves on an earlier client called Hector
- ◆ Honu: a high throughput streaming data logging system based on Apache Chukwa

Rik: In your presentations you mention two tools your developers use: Jenkins and Perforce. Could you tell us more about those tools? For example, why did you choose Perforce over an open source solution?

Adrian: Netflix has been using Perforce as its in-house source code control system for many years, and when we moved to the cloud we didn't change it. We use Jenkins to run our build system for the cloud. Carl Quinn presented on this at DevOxx [4]. Carl's team runs Perforce, Jenkins, Ivy, Artifactory, and related tools for our cloud developments.

Rik: You've made it clear that your move to AWS is as much about reliability as it is about flexibility. Yet even AWS can fail, as seen in the April 2011 partitioning event that occurred in Amazon's East Region datacenter [3]. That event was related to a specific network configuration that exacerbated the initial problem, a problem that occurred because the failover hadn't been tested under full load. What do you recommend that other organizations that plan on, or have moved to, cloud operations do to prepare for such events?

Adrian: We have published a Tech Blog that summarizes what we learned from that outage [5]. There is a lot of detail there that answers your question.

Rik: What's coming next?

Adrian: We are currently working on the backend infrastructure to support our UK and Ireland launch, which leverages the AWS Europe region located in Ireland. Netflix has no employees in Ireland, and the flexibility this gives us in contrast to owning our own datacenters is extremely valuable. From a technical perspective, the Netflix Global Cloud Platform is being polished, hardened, and tuned to run more efficiently, and we have several additional components we are planning to open source during 2012.

Resources

- [1] Slides for Cockcroft's presentations: <http://www.slideshare.net/adrianco>.
- [2] Cockcroft's blog entry: <http://techblog.Netflix.com/2011/11/benchmarking-cassandra-scalability-on.html>.
- [3] Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region: <https://aws.amazon.com/message/65648/>.
- [4] Carl Quinn at Devvix: <http://www.devvix.com/display/DV11/Carl+Quinn>.
- [5] Netflix blog regarding lessons learned from AWS outage: <http://techblog.netflix.com/2011/04/lessons-netflix-learned-from-aws-outage.html>.