

TWEETERS AND FOLLOWERS: MAINTAINING THE HEALTH OF THE TWITTERSPHERE

A conversation with Engineer, Security Researcher, and Photographer Twitter John Adams

Open a newspaper. Turn on the radio or TV. Go online. You can't turn around without running into Twitter. Everyone is writing about it, talking about it, and millions and millions of people are actually tweeting. The first news of the Mumbai attacks and the ditching of a jetliner in the Hudson River was pushed out via Twitter. Cutting-edge marketers are using it to sell shoes and computers, and to find unhappy customers and help fix their problems. But what's behind this phenomenon that pumps out millions of 140-character-or-less messages every day — or is it every hour? Benchmark reached out to Twitter Operations Engineer John Adams to get a look at the technology behind the tweets, and how the world's most popular micro-blogging platform is staying ahead of its exponential growth.

Benchmark: Let's get the must-ask question out right away. What's the official count of Twitter users as of today?

John Adams: Twitter doesn't currently release information as far as total number of users or number of active users. What I can tell you is we've had very, very massive growth. I think February '08 to '09 was almost 1800%.

Benchmark: Eighteen hundred percent? Wow.

John Adams: yeah — it was 87% from Jan 09 to Feb 09.

Benchmark: OK, so whatever numbers we find in the media, it's safe to say that you're adding millions of users very quickly. What kind of infrastructure do you have to support them, and how do you keep up?

John Adams: Let's see, what can I tell you is — we have a large number of servers. We use a combination of Ruby on Rails, Memcached, MySQL and Scala server to run the site. The day-to-day is a lot of trying to understand how many machines we need to order, and trying to keep ahead of the capacity curve. The one thing about Twitter — which is not really unique to Twitter but that many people need to understand in capacity planning, is the use of a metrics-driven architecture. We don't make changes until we have numbers to back up what we're about to do. I think there are many sites out there where the administrators are very reactionary and they'll say, 'maybe changing these random things will make things better.' Or maybe by changing something else, buying more

servers, or changing the version of this software performance will somehow improve.

We tend to not work that way. We don't work in a guesswork fashion, ever. We track thousands of metrics on the site using many homegrown tools and some external services.

For the most part we're interested in ensuring that we have a very, very accurate picture of the error rate and delays on the site, and an overall view of a user's experience.

Benchmark: Even as you're adding new capacity every day just to keep up with your user base?

John Adams: Yes. We are always attempting to find proactive ways of handling our load, as opposed to firefighting. I really believe that over the last nine months it's made a real difference.

Benchmark: What sort of external monitoring are you doing? Are you using agents out in the field using real browsers, or are you simulating the end user experience in some way?

John Adams: At present we don't do much as far as simulations. We do run some synthetic load tests. We use SmokePing, MRTG, and Ganglia. And those are all open source tools. External monitoring is on different sites outside of our infrastructure using many of the same tools.

Benchmark: But even as you're getting readings on performance, it's changing, because you're racing to keep up with demand. I can't imagine that you have a ton of excess capacity sitting around just waiting to be used.

John Adams: Every week is an exercise in capacity planning to scale the service. And there are multiple aspects of that. We're constantly changing code, changing the way that we handle the database and always trying to reduce the load that the servers are experiencing. Even though you buy more and more servers, you still have to write efficient code to really utilize that hardware. That's what our engineering team does, and they're amazing.

We are a source of metrics for the engineering team as well as a customer of the engineering team. We are taking input from them

and we are evaluating what the performance of that software is, and working in this continuous feedback loop, to improve the site.

Benchmark: What about the stresses caused by event-related surges in traffic? Did you have any difficulties on Inauguration Day, for example?

John Adams: Yes, somewhat, but we know that these events are coming in advance and we plan for them. And we also have one thing that we have sort of as our ace in the hole — we have hundreds of levers that we can pull on the site that enable or disable specific functionality. So that allows us to increase site performance at a very small user impact. It's interesting, if you look at a lot of large sites, they all do this — it's not a novel concept or any sort of proprietary Twitter feature. I remember reading once that CNN has a feature that they could put the entire site into a static mode, where it was just static HTML to eliminate interactions with the database. You have to have a backup plan.

Benchmark: OK, so take us through the journey of a typical tweet. Once I send it out, from my desktop or my phone, what happens to it?

John Adams: About twice as many requests come from API as come from the web. In the case of either request, they both get processed by the Web front-end, which then goes to Rails. Rails would then fire off the particular application controller. In most cases it's the status controller, which is what people use for posting. We accept input in many different ways, XML, JSON, and HTTP, to name a few.

From that point it enters the message queue. The message queue holds the message and that message then gets processed by any number of back-end processes that deliver the message.

There are so many issues involved in there as far as, one request comes in, many requests go out. Let me back up. We call this process the fan-out, and we identify problems in message delivery by looking at the age and size of those queues. People that work in financial institutions or any institution involved with large volume message processing would be very familiar with this paradigm.

Benchmark: The idea that one message comes in, and that triggers many messages going out? And those messages can be going out in many ways — to the Web site, to an application, to a mobile device.

John Adams: Yes, one message comes in and you have to notify a number of people. You have email as a device. An SMS destination, what SMS providers call 'mobile terminated,' is a device. And each of those various devices has a separate queue.

So it's very important to us to ensure that we work in real time — they enter the queue and leave the queue in real time or as close to it as possible.

Benchmark: That's a lot of complicated traffic to keep flowing smoothly and quickly.

John Adams: It is. We have a number of internal service agreements with ourselves that 'these things will happen within this time,' otherwise alarms go off and people get paged. I think if you remember last year, the general failure mode of Twitter was to whale and not be available. Now our auxiliary mode tends to be, we would rather have the site delay, rather than be unavailable. We have caching in place in case databases fall behind; or if a message queue becomes unresponsive or becomes too backed up, we have dynamic ways of changing how that stuff is processed. So that message will get written to our cache and it goes right into the database and then you see it on your screen. In the case of an SMS message we then generate an external request that goes to somebody else to deliver that.

Benchmark: I imagine a lot of people think, it's just 140 characters, how hard could it be?

John Adams: It's not super complicated, but I think a lot of people believe that Twitter is so simple and so easy that anybody could build it. And I think that that's entirely wrong. I mean it is, on the surface, a very simple service, in the same way that Google Search is a very simple search service from the outside.

You type in something and a response comes back. But dealing with a load of millions and millions of users and being able to process those requests in a very limited time is very difficult.

Benchmark: So I send out my Tweet, 'having a chat with John Adams at Twitter,' and it goes through the queues and shows up on the Twitter Web site, and then gets broadcast out as an SMS?

John Adams: Everything goes through the Web, right, no matter what. Everything goes through the search engine no matter what, because everyone is part of this standing public timeline. And then, user by user, preferences determine whether or not we will send an SMS.

Benchmark: The follower's preference?

John Adams: The follower's preference. The beauty of Twitter is in asymmetric relationships between people. If you look at Facebook, you have to have a one-to-one following — you have to follow me and I have to follow you before there's communication going on between us. But on Twitter, it's asymmetric. So you can follow someone like Shaquille O'Neal and he doesn't have to follow you.

Benchmark: The other thing about Twitter that seems to be growing very quickly is the number of applications that can be used to access the service. There are a whole host of ways to connect.

John Adams: Yes, Our service works over multiple devices including mobile texting (SMS), web browsers, and API clients. Twitter has an open API and you can get information about the API at apiwiki.twitter.com. There are many open source libraries that are available, with prebuilt libraries that communicate with the API, which anyone can use. For applications that use the service very intensely, they may have to talk to us first to avoid our rate limiting and the controls that we've put in to fight spam and abuse. There's a really wonderful developer ecosystem that has built up around the site. I think that's been a big advantage for us.

Benchmark: What has surprised you the most about how people use Twitter or the impact of the service?

John Adams: I think the biggest surprise that I've had is in the way the site transcends class and social boundaries between

people. If you woke up ten years ago and wanted to send mail to a famous celebrity or communicate with them, you would never get a response, right? You'd never be able to communicate with them. Now you can.

Since the communication is so frictionless on Twitter, you can talk to people that are much more experienced, let's say, or that are working in a completely different field than you, and get immediate feedback from them.

So that was just such a big shock. The fact that you could send a message to some famous musician who immediately responds, or you could sit there and talk to MC Hammer and see how his day went.

It's very kind of crazy. Most recently we were on Oprah, and we had the whole Larry King/Ashton Kutcher war to get a million followers which created it's own set of stresses on the site.

Benchmark: And the famous get more famous. Well, we'll be following Twitter to find out how this amazing phenomenon plays out. Good luck!



John Adams

Engineer, Security Researcher, and
Photographer
Twitter

John Adams (@netik) is a Engineer, Security Researcher, and Photographer. He currently works in Operations at Twitter and is responsible for scaling, securing, and improving the micro-blogging service. He has worked in the computer industry for over eighteen years, focusing on media distribution, security, scalability, and (more recently) infrastructures to drive high-volume social networking. From his original work on some of the first e-commerce sites, to the communications juggernaut that is Twitter, he has worked on the public Internet from its start. He was the Director of Engineering at IFILM, and worked in the security groups at both Apple and Inktomi.

© 2009 Keynote Systems, Inc. All rights reserved. The trademarks of Keynote Systems, Inc. include Keynote®, DataPulse®, CustomerScope®, Keynote CE Rankings®, Keynote Customer Experience Rankings®, Perspective®, Keynote Red Alert®, Keynote Traffic Perspective®, Keynote WebEffective®, The Internet Performance Authority®, MyKeynote®, SIGOS®, SITE®, Keynote™, The Mobile & Internet Performance Authority™ and all related trademarks, trade names, logos, characters, design and trade dress are trademarks or registered trademarks of Keynote Systems, Inc. in the United States and other countries and may not be used without written permission.