http://highscalability.com/blog/2009/4/5/at-some-point-the-cost-of-servers-outweighs-the-cost-of-prog.html

At Some Point the Cost of Servers Outweighs the Cost of Programmers

Sunday, April 5, 2009 at 7:12AM

Todd Hoff in RoR, Twitter, scala

This is the intriguing quote by Bill Venners in an interview with Twitter's Alex Payne on Twitter's heretical switch from a pure Ruby stack to a Ruby on Rails stack on the front-end and JVM/Scala on the back-end:

> *So performance was also one of the problems with JRuby, which I [Bill Venners] think helps explain better why they'd [Twitter] prefer Scala over Ruby or JRuby for some things.*
>
> *I have often heard Rubyists say that although Ruby is slower than Java, for many things it is plenty fast enough, and they are right. The logic goes further, saying that servers are cheap, and programmers expensive, so it makes sense to tradeoff some runtime*

> *performance for programmer productivity. And I think that's very often true too, but not always.* ***If you have enough traffic, at some point the cost of servers outweighs the cost of programmers****. I'm not sure whether Twitter is past that point, but they get a lot of traffic. And frankly this isn't an intrinsic tradeoff. Other dynamic languages are faster than Ruby, and Scala is too. And people can be quite productive in these other languages too, including Scala.*

I feel Alex's *Max Payne*. You might wonder why the geekosphere cares so passionately which technology stack Twitter uses? Well, it's Twitter and it's Ruby on Rails. That's like the Lindsay Lohan and Samantha Ronson of tech buzz. It creates it's own self-sustaining posting reaction. Boom!

It took some giant cajones to switch from a well defended platform like Ruby on Rails to an obscure language like Scala. Few people would have been brave enough to pull the trigger on that decision.

Twitter didn't take this large leap out of ignorance or incompetence. Twitter's Steve Jenson said they *spent several weeks going over our options, running extensive load tests, and presented our findings to the team at each stage. We did our due diligence.*

They did the work and came to conclusions valid for their situation. They have to follow their own bliss. They aren't telling you to use Scala. They aren't telling you not to use Ruby. Have at it. But they have chosen the path less traveled and seem happy with the direction they are heading. If you aren't happy with their decision then that's a you problem, not a them problem.

This points out for me the evolving nature of the two tier web: a client tier and a back-end tier accessed only via an API. That back-end tier can be implemented in anything at all. Twitter likes the JVM for it's undeniable performance, reliability, and scalability. They may like Scala because it has a lot of cool features, is concise, has static typing, performs well, and is a pleasure to develop in.

When people jumped with a happy little grin on their face to Ruby, they loved a lot of things about Ruby that helped them make that decision to switch. Maybe Scala is cool, effective, and fun to use too?

So instead of worrying how the homeboys have ever so indirectly dissed Ruby, maybe it's worth taking a look at Scala to see if it's actually any good?

As a start try Bill Venners on the rise of Scala. It's a great overview of Scala and why it's a worthy next evolution. Then maybe take a peek at First Steps to Scala. And if you want to take a look at some real-life code try Kestrel - tiny queue system based on starling.

Knowing only the vaguest bits about Scala before my own little exploration, I come out impressed and interested. I like a lot of things Bill had to say: Scala means scalable language as in it scales to different sized tasks and domains; Scala's static typing is valuable, especially in a team project; Scala has the feel of a scripting language, but can also work as a systems language; Scala is an artful blend of a fully Object and fully Functional language; Scala supports imperative programming, but with a functional bent; Scala can express more in types than Java so you get more rigorous type checking; Scala is concise so can say more with less; Scala treats libraries like creating internal DSLs; Scala helps programmer productivity like Ruby and Python.

Sounds like Scala is worth a deeper look to me. Can't we all just get along?

Well, that's not how this post started at least. It started with Bill's refreshingly anti-establishment statement: *If you have enough traffic, at some point the cost of servers outweighs the cost of programmers*. This is an idea we don't hear much of anymore in this era of abundance. Servers cost real cash though, especially for bootstrappers and the truly humongous. So if you can cut your cash requirements by putting in a little programmer elbow grease, that makes a lot of sense to me. Performance still matters.

Related Articles

- Twitter on Scala by Bill Venners
- My Reasoned Response about Scala at Twitter by Obie Fernandez.
- Twitter Blaming Ruby for their Mistakes by Tony Arcieri
- Scala's home page
- Hacker News Thread on Message Queue Options
- The Secret Behind Twitter's Growth by Kate Greene
- Why Scala for Web 2.0? by Alex Payne
- Mending The Bitter Absence of Reasoned Technical Discussion by Alex Payne

---

Article originally appeared on High Scalability (http://highscalability.com/).