

# 6 Ways Not to Scale that Will Make You Hip, Popular and Loved By VCs

Monday, April 18, 2011 at 8:49AM

Todd Hoff in MySQL, Strategy, funny

This is a hilarious presentation by [Josh Berkus](#), called [Scale Fail](#), given at O'Reilly MySQL CE 2011. Josh is entertaining, well spoken, and cleverly hides insight inside chaos. And he makes some dang good points along the way.

Josh has a problem, you see Josh has learned how to make sites that are both scalable and reliable. So he's puzzled why companies "whose downtime interfaces (Twitter) are more well known than their uptime interfaces" get all the attention, respect, and money for being failures. Just doing your job doesn't make you a hero. You need these self-inflicted wounds in-order to have the war stories to share at conferences. They get the attention. Just doing your job is boring. This is so unfair in that way life can be.

So if you want to turn the tables and take the low road to fame and fortune, here's Josh's program for learning how not to scale:

**Be trendy.** Use the tool that has the most buzz: NoSQL, Cloud, MapReduce, Rails, RabbitMQ. It helps you not scale and the VCs like. Use Reddit to decide what to tool use. Whatever is getting the most points this week is what you should use.

**Troubleshoot after the barn door has closed.** Math is not sexy. Statistics are not sexy. Forget resource monitoring, performance testing, traffic monitoring, load testing, tuning analysis. They are all boring. Be more intuitive. Let history be your guide. Whatever

problems you had on your last job are the ones you'll have on this job.

**Don't worry about it.** Parallel programming is not sexy. Erlang can parallel program a 1000 node cluster, but it's not sexy. Be hot and from the hip. Ignore details about memory and management. Don't worry about it. Use single-thread programming, lots of locks, ignore scope and memory contexts, have frequently-updated single-row tables, have a single master queue that controls everything, and blocking threads are your friend.

**Hit the database with every operation.** Caching is not your friend. Every single query should go directly to the database. Ignore caches completely.

**Scale the impossible things.** Scaling easy things is for wimps. There's no hotness there. There's no speaking engagements in scaling web servers, caches, shared-nothing hosts and simple app servers. Scaling the impossible things is where the hotness is: transactions, queues, shared file systems, web frameworks. This is how you can have the long nights and weekend and the war stories that will get you up on stage.

**Create single points of failure.** No matter how large your software is, you must have a couple of places where a single point of failure will bring down your entire infrastructure. Like a single load balancer, or a single queue, or load balancers that run a 100% capacity.

Josh says that after following this program you'll have learned how not to scale and become the big macho guy on stage.

If you've worked as a programmer for any period of time this analysis strikes home. What we know about human nature is that to be a hero you must overcome great odds. If your code always works and never stops a release with a sev one bug, then paradoxically, you will not be an

organizational hero. You will just be reliable Good Old Joe. The hero is the programmer who stayed up 7 days straight, mainlining red-bull, looking all bleary eyed, to get that last check-in just before the dead-line. All this to fix a problem they had in fact created from the start.

I may add that here at HighScalability.com we would love your story on how you keep your site up. Tell us your Hero story.

## Related Articles

[Hilarious Video: Relational Database Vs NoSQL Fanbois](#)  
[NSFW: Hilarious Fault-Tolerance Cartoon](#)

[Brian Aker's Hilarious NoSQL Stand Up Routine](#)

---

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

See website for complete article licensing information.