



Overclocking a Fusion Reactor

I don't know how many of you have written to me whining about performance because I bin those letters as soon as I realize what they're about. Sometimes that takes a few minutes, because so few are capable of composing a letter that complies even minimally with English language standards, distressingly low as they are. The *FreeBSD Journal* is a classy establishment. We have not, nor will we at any time in the future, accept letters via text message, as most carriers have banned the emojis most suitable for response as unspeakably obscene.

I do, however, feel obliged to throw a few of your queries into the blender, and I see some readers want to know how to optimize performance.

I have suggestions. Many suggestions.

They all come down to: don't.

The natural impulse to compare your server against others, to whup your neighbor, is exactly that: natural. It's like flipping rocks to find tasty grubs. Sleeping in trees coveting the upper-class caves inhabited by the snooty grizzly bears. Perishing within 20 years of What Is Vitamin C Anyway? We invented civilization to escape dysentery, cleaning ourselves with mud—and benchmarking.

This is so natural, it's even in the most sacred of geek cultural touchstones: Star Trek. In Deep Space Nine, O'Brien is constantly fiddling with the Cardassian space station to get its performance to a level he finds acceptable. Not the station commander's standards. Not the station's inhabitants' desires. A level he finds acceptable. Truth is, the space station works just fine. The space station resists his tampering because its software knows how the systems are *supposed* to behave, and his obsessive tinkering threatens its integrity. Someone combining my Trek gene with an Obsessive, Irrelevant, Detail-Measurement gene has certainly counted how many times O'Brien almost killed hundreds of people by insisting on overclocking an alien fusion reactor. (The United Federation of Hold My Beer simultaneously explains all Star Trek plot holes *and* all Star Trek plots. Look it up.)

Maybe it's your job to nurture the server and you want to know that you're doing your job well. The machine doesn't get a vote on how you're doing your job, though. IPC with your organizational parent process instead. It's your manager's job to tell you how you're doing.

But no. Some of you will insist you can eke a few extra percent of performance out of the system. The fact that everyone might wind up sucking vacuum is a risk you're willing for them to take.

The only performance that matters is what the user experiences.

So, stir your lazy carcass and communicate with the users.

Preferably by voice. Writing is a horrid communications mechanism at the best of times.

If you can manage it, communicate in person. Barring plagues, of course. *Two-way* communication. Converse with your users.

This means listening. You can't use the time people are babbling ignorance at you to figure out what you're going to say next. You need to process their babble and extract meaning from it. Preferably, the meaning they intended. Present your extracted meaning to the user and

request verification. I guarantee that your users' suffering has no bearing on how many disk writes per second the database groans out. If by some chance such things are relevant, I guarantee you then an extra one or two percent performance is not going to ease their woes.

You won't enjoy this process. Sysadmin Rule #2 is very clear: "People were a mistake."

But the system exists to serve the users.

Many of the things they disclose will appear to be not your problem. If they're trying to run your fancy client-side JavaScript application on a prototype Atom processor from the previous millennium, it will be slow. Is the problem that their desktop should have been sent to the Giant Heap of Eternal Toxicity a decade ago? Is your application wrong for this organization? Were both desktop and application approved by a computer-illiterate whose primary qualification for the Purchasing gig is an ability to keep the Chair of the Board of Directors' most troublesome child from constantly calling Mommy with their brand-new Distressing Drama of the Day?

Will benchmarking help you with the issue? No. Discover the real problem. Solve it.

If the lure of the natural urge to benchmark proves too strong, though, sublimate it in useful directions.

Sure, set up monitoring to be sure that your application responds within reasonable time limits. That's more of an operations issue. Graph the results of that monitoring so that you can see how performance changes over time. When someone says, "the application seems slow after the upgrade," that graph will either let you declare that you're already aware of the issue or will justify your scorn.

Beyond that, target a user. You know the one. The Problem User.

Benchmark *them*.

Fire up your favorite packet sniffer. Watch the interaction between the server and Problem User's machine. Capture packets. Analyze them. See what happens in the real world. Take notes.

Dtrace, ktrace, truss. They're all your friends. Watch the server do its job. Where does your application spend its time? Take more notes and samples.

Perhaps you can find a real problem that affects Problem User. If not, you'll learn an unwholesome amount about your environment and your applications.

Eventually, you'll get called into a meeting with your organizational parent process. Maybe it's an annual review. Perhaps Problem User has finally made enough noise that Management has no choice but to stir themselves enough to make interested noises. When they ask what you've been doing, you have a stack of data to show them. Perhaps it's not useful data, but that's okay. You've delved deep into the innards of your environment. You can demonstrate that you have spent time investigating Problem User's issues. Packet traces and Dtrace results, heavily annotated with your comments and arrows and swooping lines connecting different sections, never fail to impress.

Yes, benchmarking is a natural human thing. And there's nothing more *natural* than a business.

You, personally, are constantly benchmarked. Make your benchmarks solid.

It's almost as much fun as overclocking a fusion reactor.

Have a question for Michael?
Send it to letters@freebsdjournal.org



MICHAEL W LUCAS (<https://mwl.io>)'s newest books are *Sudo Mastery, 2nd Edition* and *Terrapin Sky Tango*.