Twitter Thread by Dave Anderson



Dave Anderson
@dave_universetf

Hooray, my IPv6 parser is mostly working, and only missing the final cursed case.

It's done. By god, it's done.

I mean, parseIPv6 is also awful and embarassing and much worse than the stdlib implementation right now. But is it at least faster than the stdlib?

Ope never mind, other tests are failing still. Bah.

lol, fuck. Not only is it 50% slower than the stdlib's parser, it's also not alloc-free.

oh, it's because it's benchmarking parsing a zone, and we have to intern the zone... I think.

Yess. For IPs without a zone, my parser is faster and alloc-free. Not hugely faster mind you.

I guess if you squint, it's about as good as the IPv4 parser. It has to touch 4x more bytes, so just under 4x slower isn't that bad.

That said, I'm pretty sure my code is a branch predictor's nightmare right now, not to mention it's entirely unreadable even by my low standards. Needs another round of improving, now that it passes tests.

I think it's time to checkpoint and play Elite Dangerous tonight, but the plan for tomorrow is to see if I can't turn this into a more explicit state machine, so that the code is considering fewer possibilities at any one time.

In general, I think all IPv6s parse in 4 stages:
- Hex before ellipsis
- Hex after ellipsis
- Embedded IPv4
- Zone suffix

All but the first are optional, and the conditionals to decide which new state to transition to is a bit ugly...

... But I think within each state the "happy loop" condition is simple and should keep the CPU and branch predictors pretty happy.

Current WIP, as I pack in for tonight to go play Elite: https://t.co/acwEFavxQi

WARNING: side-effects may include eye bleeding and stomach ulcers.

It also still has a bunch of debugging code in it, the comments may not match the code any more, and I haven't yet iterated on the benchmarks and SSA to see what hand-inlining was actually productive vs. harmful. So it's all hand-inlined, in the worst possible way.

The code's definitely not getting submitted like this, but that's my hacking process I guess :)