



I was wrong about Rust build times

And I have learned the errors of my ways

Why is my Rust build so slow?

Dec 30, 2021 30 minute read [rust](#) · [release-engineering](#)  [fasterthanlime](#)

Contents

[What is cargo even doing](#)

[How much time are we spending on these steps?](#)

[Linker, is it you?](#)

[Debug symbols, perhaps?](#)

[Incremental builds](#)

[Link-time optimization \(LTO\)](#)

[Rustc self-profiling](#)

[Warp, I trusted you](#)

[Revisiting all the other changes](#)

[Splitting into more crates!](#)

[Conclusion](#)

I've recently come back to an older project of mine (that powers this website), and as I did some maintenance work: upgrade to newer crates, upgrade to [a newer rustc](#), I noticed that my build was taking too damn long!



The still-relevant

- ``cargo build --timings`` is still useful



The still-relevant

- Apple made Id64 twice as fast (WWDC 2022)
<https://developer.apple.com/videos/play/wwdc2022/110362>



The still-relevant

- Incremental builds are still better but:
 - The idea wasn't allowed to go all the way
 - It really only works for local caching
 - Some CI providers have started "bringing the builds to cache" rather than the other way around



What was wrong

- Splitting crates. Why?



My new approach

- Dynamic linking with stabby