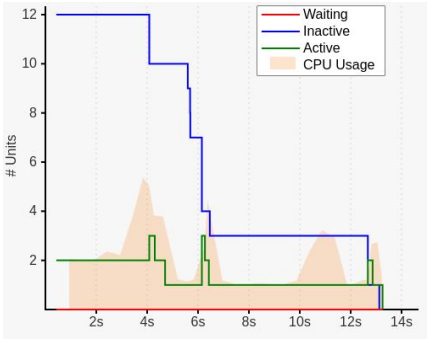
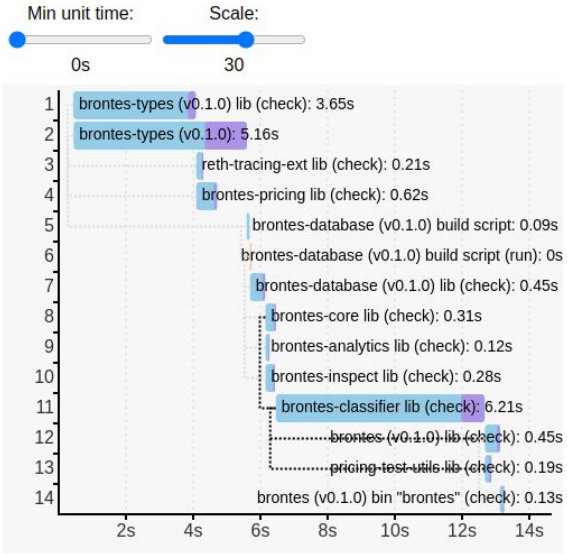
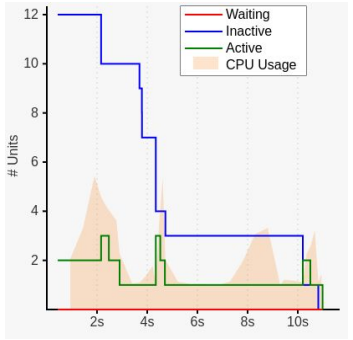



## Original Rust compiler

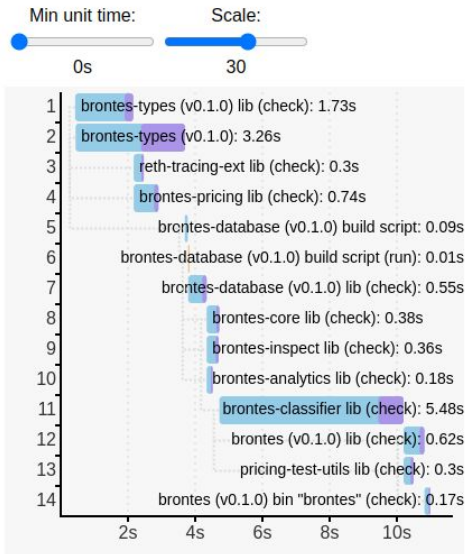
**13.3 sec**
for an incremental build




## Modded Rust compiler

**11.0 sec**
for an incremental build



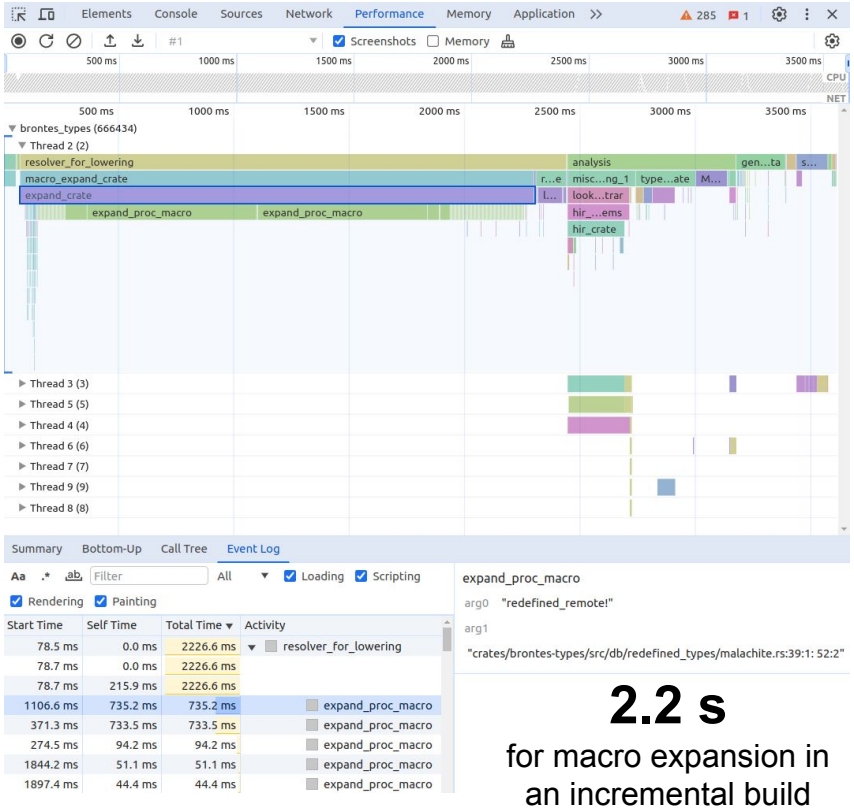

Our compiler is **17% faster than the default compiler** for incremental builds. Note: it is *not* faster for clean builds.

How? We implemented macro expansion caching.

The next page is a case study examining the macro-heavy crate (brontes-types) using the Rust self-profiler

# Original Rust compiler:

*Proc macros are always re-expanded*



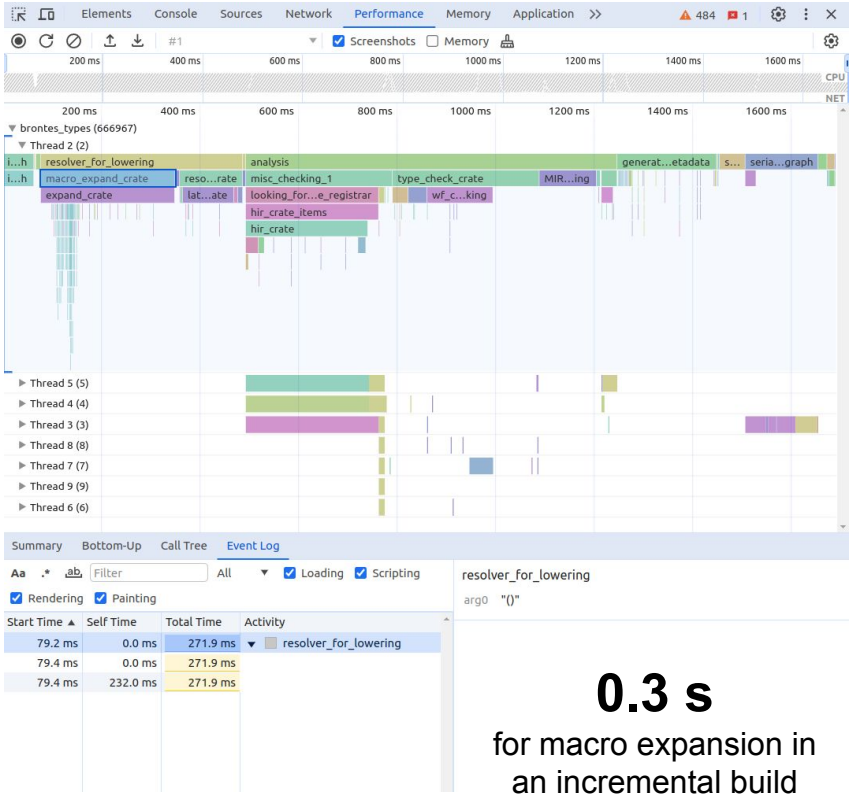

**2.2 s**

for macro expansion in an incremental build

Your old compiler: On **any** code change, **all** of the 350+ macros in this crate are expanded (and 1000s across crates)

# Modded Rust compiler:

*Cache proc macro expansion*




**0.3 s**

for macro expansion in an incremental build

Our modded compiler: Here we cache all macro expansions! This effectively solves IDE lags when expanding macros