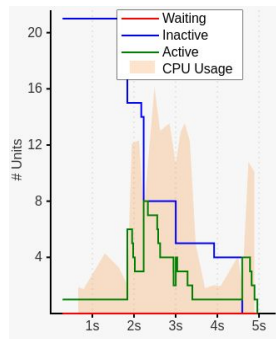
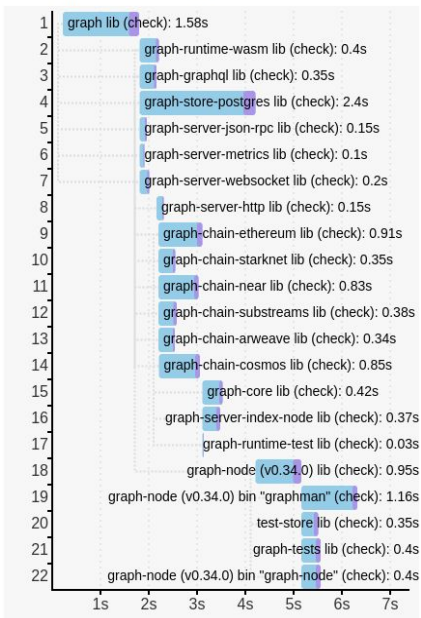


Original Rust
compiler

6.3 sec

for an incremental build

Min unit time: Scale:

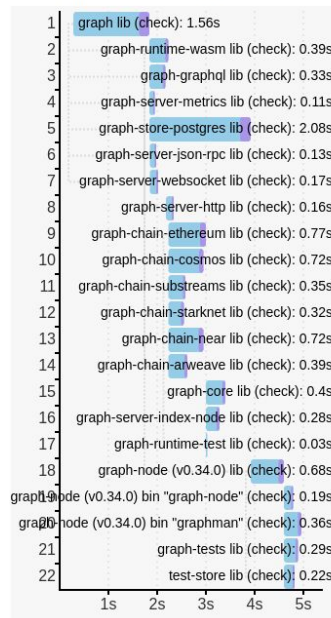


Modded Rust
compiler

5.0 sec

for an incremental build

Min unit time: Scale:



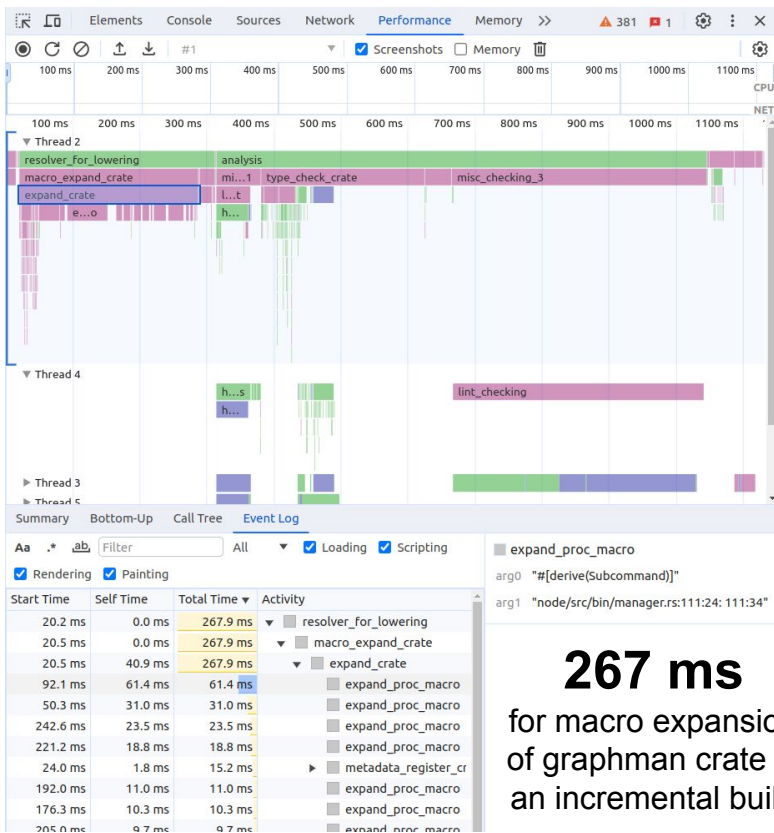
Our compiler is
**21% faster than the
default compiler**
for incremental builds.
Note: it is *not* faster for
clean builds.

How? We implemented
macro expansion caching.

The next page is a case
study examining the slowest
crate (frame-support) using
the [Rust self-profiler](#)

Original Rust compiler:

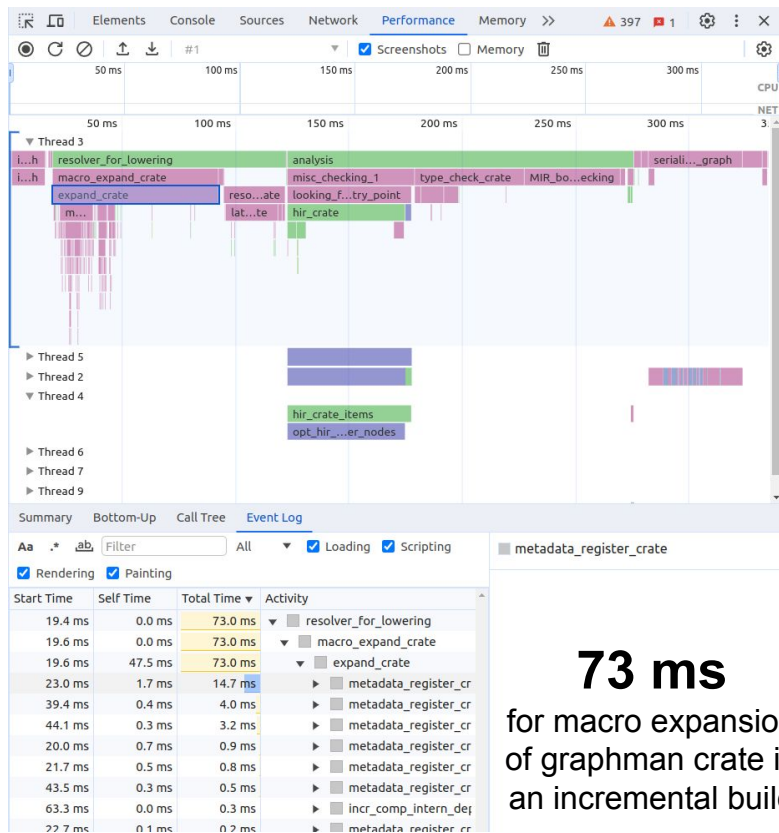
Proc macros are always re-expanded



Your old compiler: On **any** code change, **all** macros are expanded
... also yielding slower type-checking (244 ms) and linting (375 ms)

Modded Rust compiler:

Cache proc macro expansion



Our modded compiler: Here we cache all macro expansions!
This also yields faster type-checking (48 ms) and linting (2 ms)