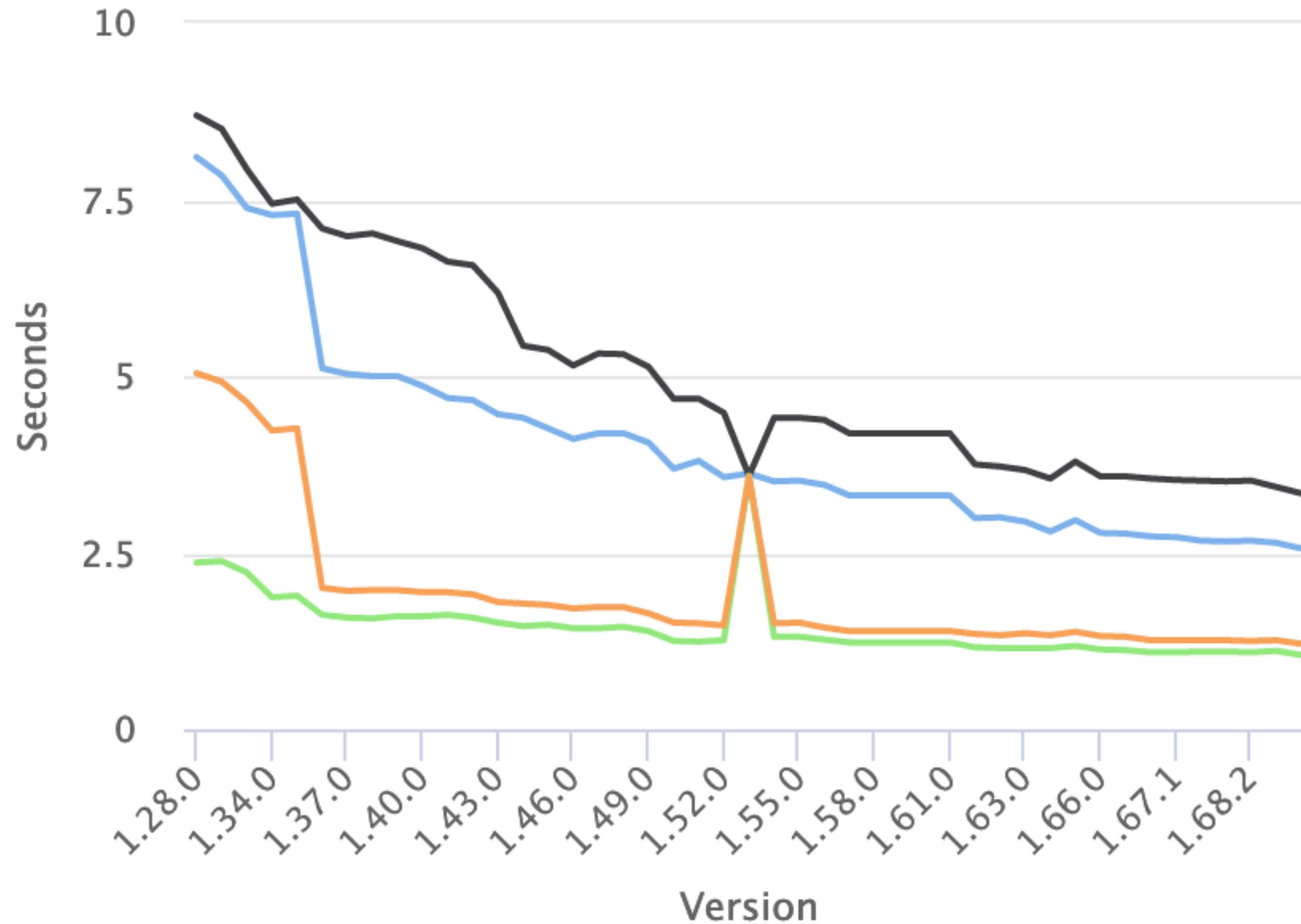


The status of parallel rustc

Nicholas Nethercote
RustNL 2023

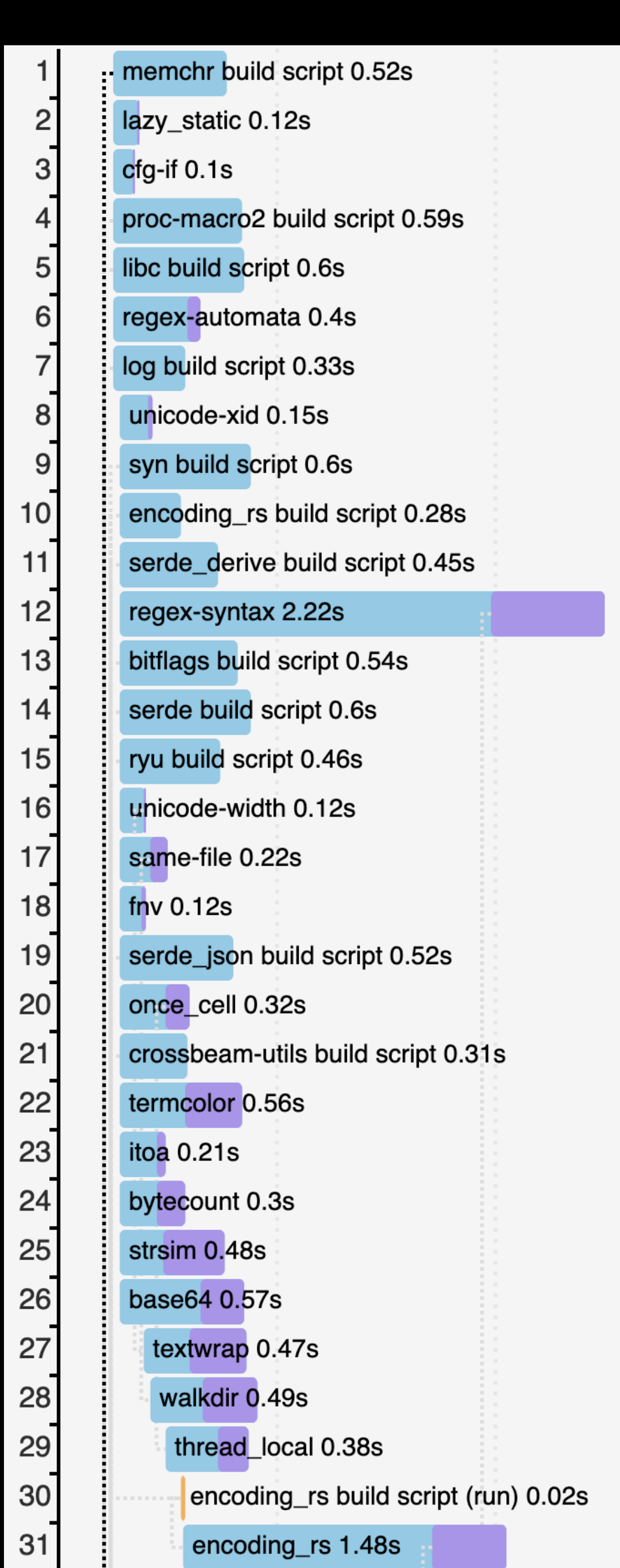
rustc is slow

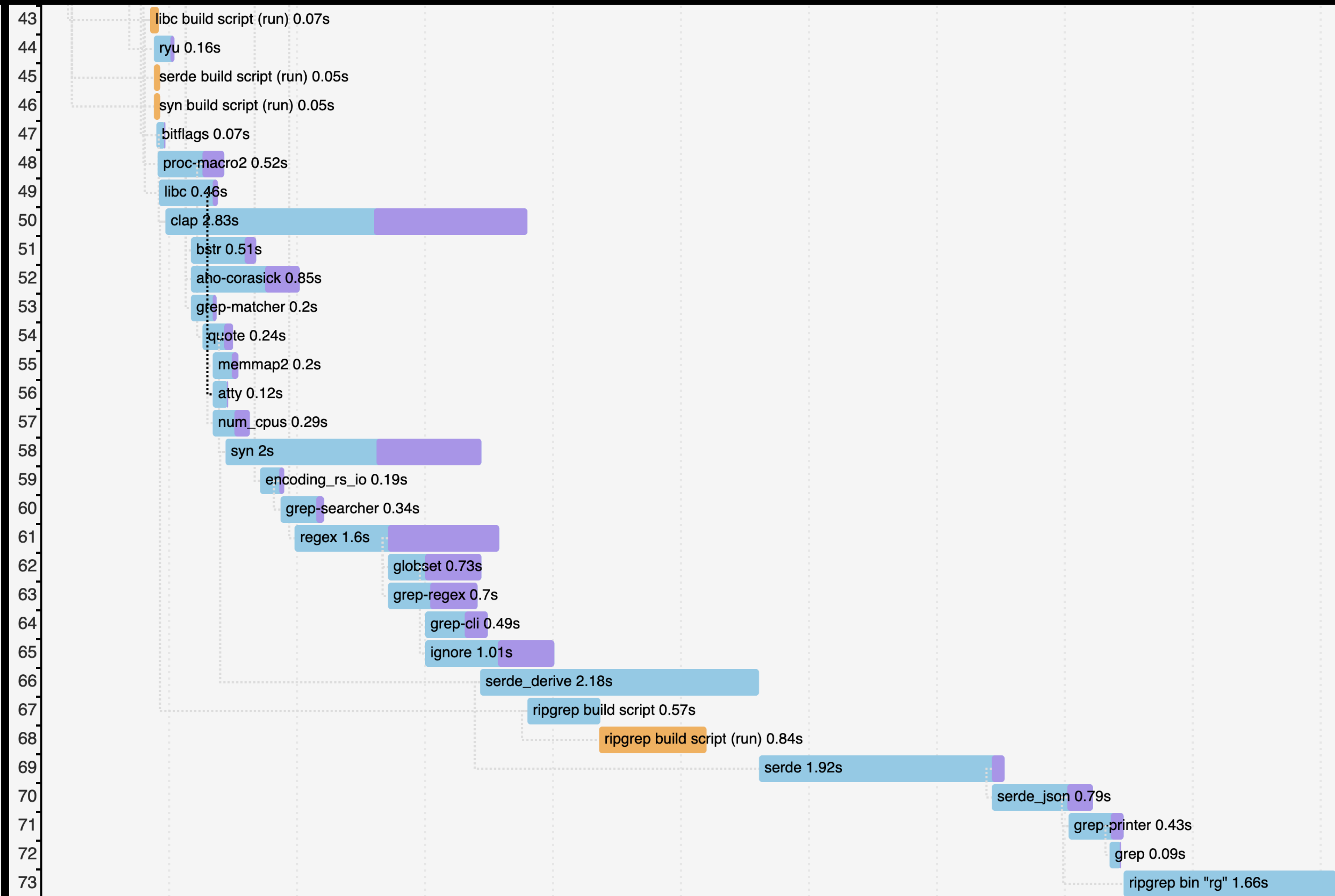
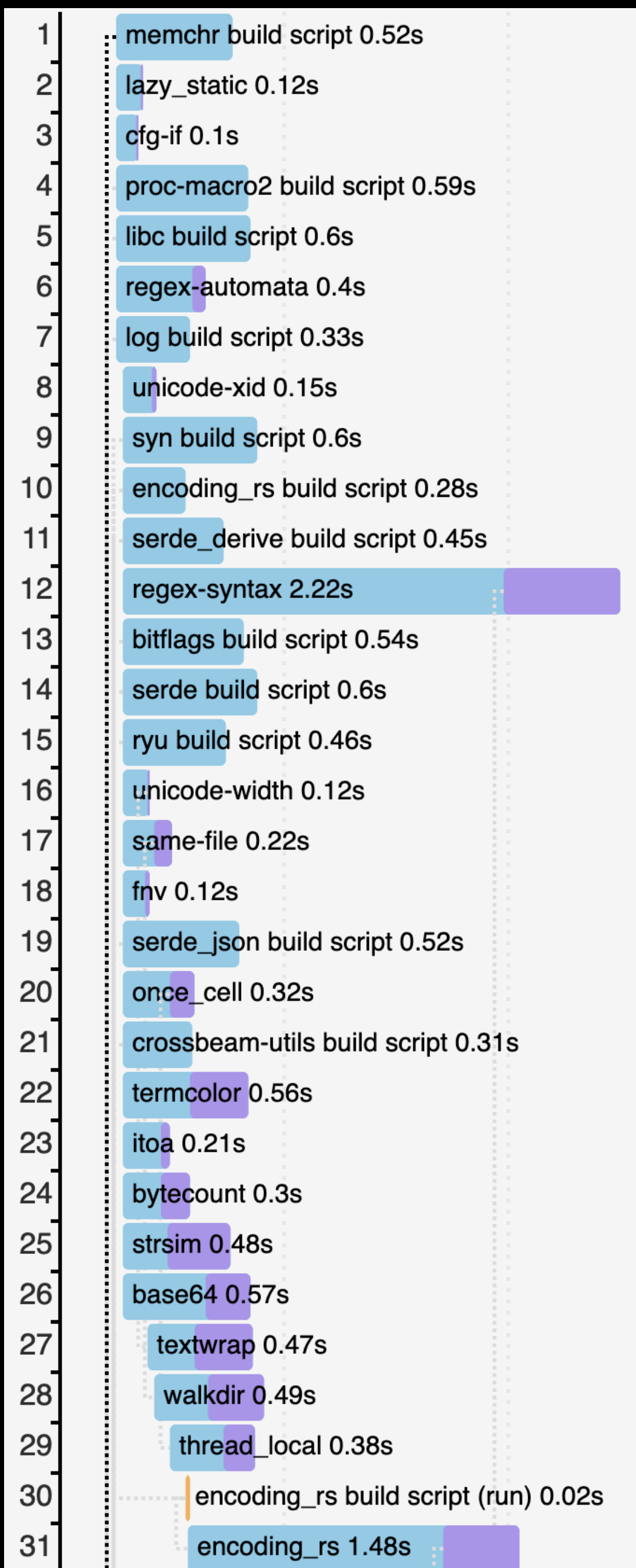
Average time for a debug build



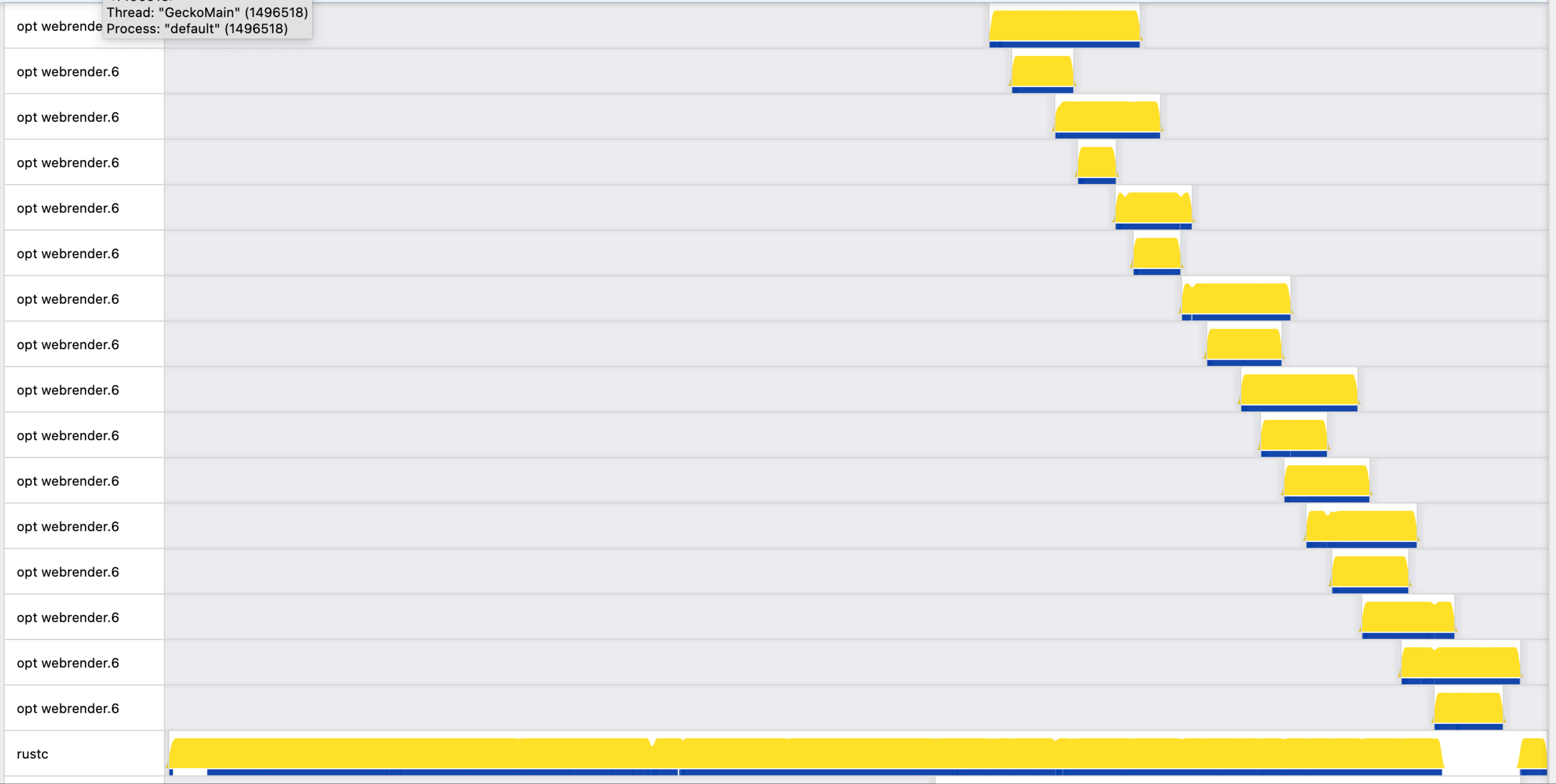
Parallel execution

Parallel execution 1: cargo





Parallel execution 2: back-end



Parallel execution 3: front-end

```
$ cat config.toml  
[rust]  
parallel-compiler = true
```

```
$ rustc +nightly -Zthreads=4
```

Operations

join
parallel
par_iter
par_for_each

Types

Type	Parallel	Serial
-----	-----	-----
Lrc<T>	sync::Arc<T>	rc::Rc<T>
Lock<T>	parking_lot::Mutex<T>	RefCell<T>
RwLock<T>	parking_lot::RwLock<T>	RefCell<T>
AtomicBool	sync::atomic::AtomicBool	Cell<bool>
AtomicU32	sync::atomic::AtomicU32	Cell<u32>
ParallelIterator	rayon::iter::ParallelIterator	Iterator

Shipping strategy

0. Serial front-end

1. Parallel front-end,
single-threaded by default

2. Parallel front-end,
multi-threaded by default

Performance tracking

	Benchmark	Profile	Scenario	% Change
ripgrep	ripgrep-13.0.0	doc	full	-5.10%
	ripgrep-13.0.0	opt	full	-4.14%
	ripgrep-13.0.0	opt	incr-unchanged	-2.56%
	ripgrep-13.0.0	opt	incr-full	-2.39%
	ripgrep-13.0.0	debug	incr-patched: println	-1.91%
	ripgrep-13.0.0	debug	incr-unchanged	-1.67%
	ripgrep-13.0.0	opt	incr-patched: println	-1.04%
	ripgrep-13.0.0	check	full	0.55%
	ripgrep-13.0.0	debug	incr-full	-0.34%
	ripgrep-13.0.0	check	incr-patched: println	0.25%
	ripgrep-13.0.0	debug	full	-0.22%
	ripgrep-13.0.0	check	incr-full	-0.21%
	ripgrep-13.0.0	check	incr-unchanged	-0.07%
	ripgrep-13.0.0	check	incr-patched: println	-0.07%

Profiles
X
Scenarios
X

-Zthreads (1, 2, 4, ...?)

walltime

vs

instructions executed

Conclusion