



Contribution ID: 173

Type: **not specified**

## Improve scheduler debuggability

Wednesday, 18 September 2024 15:44 (22 minutes)

One of the bottlenecks to making progress in scheduler is understanding ‘what is the problem?’

Reporters who don’t understand the scheduler can’t provide useful info to help root cause why they see a problem.

Developers, seasoned or new, can trip over many details and corner cases that might make what appears to be a bug to be actually a feature that is just not well understood by the developer.

Using bare tracepoints we can start adding more probing points to understand why wakeup path and load balancer (the most complex parts) have made a decision at a specific point of time. But we don’t want to stop there, but solving these two should pave the path to do more.

The difficulty is then on how to we extract this info and present it in a way that is easy to visualize and debug?

sched-analyzer [1] aims to solve this by hooking into mature existing technologies

1. ftrace
2. perfetto
3. bpf

It is a glue logic that uses bpf and ftrace to connect to the tracepoints and extract whatever info deemed useful and emit them out as perfetto event - which has a mature visualization and sql based query to help understand what a scheduler is doing at any point of time.

It also has a python pandas interface that is combined with its sql query can enable creating strong post processing analysis tools to identify patterns and problems from a captured trace.

sched-analyzer has a TUI based interface that should make sharing output easy on the list.

The main goal of the discussion is to explore ways to introduce better debugging. With sched-analyzer as a potential tool to build on top.

[1] <https://github.com/qais-yousef/sched-analyzer>

**Primary author:** Mr YOUSEF, Qais (Google)

**Presenter:** Mr YOUSEF, Qais (Google)

**Session Classification:** Sched MC

**Track Classification:** Sched MC