

Using DL servers for FIFO tasks starvation avoidance

Thadeu Lima de Souza Cascardo

September 19, 2024



On the author

- Kernel generalist developer
- Specialized in finding and fixing bugs



Problem Background

- SCHED_FIFO tasks run until they sleep or a higher priority thread becomes runnable
- Non SCHED_FIFO tasks may run by using a throttling mechanism
- Same-priority SCHED_FIFO tasks will run in FIFO list order **if** the running task sleeps



Problem statement

- FIFO tasks may be starved by other FIFO tasks
- Those starved tasks may be waiting for mutexes or even hold them
- That may starve other tasks as well
- RCU stalls and task hangs warnings may trigger



Proposed solutions

- In LPC 2021, Sharan Turlapati and Srivatsa Bhat presented
“Linux Kernel Support for Kernel Thread Starvation Avoidance”
- stald userspace solution
- A Kernel stall monitor was proposed
- Mailing list patches were not found



Deadline servers

- Introduced as a mechanism to allow fair tasks to run, that is, avoid starvation
- Still work in progress, only partially merged
- Allow a `SCHED_DEADLINE` entity to service lower priority tasks



Proposal

- Use DL servers to avoid starving FIFO tasks
- Introduce an `rt_server` that would service starving tasks under `SCHED_DEADLINE`
- Challenges: how to pick up the next task



Alternatives

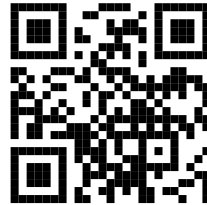
- This is working as designed even if it allows system hangs.
- `stalld` should be used until misbehaving programs or kernel threads are fixed
- Corner cases should be fixed when identified



Open discussion

Join us!

<https://www.igalia.com/jobs>





Backup

- Case in point:
 - vhost kthread consuming from an eventfd may lead to an infinite loop
 - it inherits scheduling policies from task starting the kthread

