



Contribution ID: 449

Type: **not specified**

Enforcing PI locks by default

Performance Inversion (more generalized form of Priority Inversion) is a common problem in the wild. RT tasks are not the only ones susceptible to it; SCHED_NORMAL are prone to it too. Whether it is due to usage of nice value, running on big.LITTLE system or DVFS, the lock holders can cause a delay to an important waiter leading to performance problems.

Proxy execution effort led by John Stultz is generalizing a solution to the problem at scheduler level. While it is not connected to futex_pi yet, but once it does it won't be effective unless userspace opts in.

Unfortunately userspace coding is too complex and layered today for any single user to identify which locks can be contended badly and cause inversion problems. The library dependency is wide and out of a single owner's control to enforce an opt-in to use PI locks. On going QoS discussions for the scheduler will mean more reasons to cause inversion problems that can affect performance.

For these reasons making PI the default behavior at the lowest possible level is necessary to address this class of problems hopefully once and for all. To my knowledge non Linux OS did make choices like this.

What would it take to flip PTHREAD_PRIO_INHERIT to be the default on libc?

How can we deal with other forms locks that are non pthread related? Some language like Java implement their own locks, how can we help them to be PI by default?

Non lock based inheritance is an important topic too. What semantics are required to be introduced to enable pthread_condition_variables be PI aware?

Primary author: Mr YOUSEF, Qais (Google)

Presenter: Mr YOUSEF, Qais (Google)

Session Classification: Toolchains MC

Track Classification: Toolchains MC