



Contribution ID: 115

Type: **not specified**

## Modularization for Lockdep

*Monday, September 12, 2022 3:45 PM (45 minutes)*

Lockdep is a powerful tool for developers to uncover lock issues. However there are things that still need to improve:

- The error messages are sometimes confusing and difficult to understand, and require experts to decode them. This not only makes read deadlock scenarios challenging to understand, but also makes internal bugs hard to debug.
- Once one lock issue is reported, all the lockdep functionalities are turned off. Although this is reasonable, because once a lock issue is detected the whole system is subject to lock bugs and it's pointless to continue running the system until the bugs are fixed. However this is frustrating for developers when they hit some lock issues that happen in other subsystems, they cannot test their code for lock issues until the existing ones are fixed.
- Detection takes time to run and creates extra synchronization points than production environments. It's not surprising that lockdep uses an internal lock to protect the data structures for lock issue detections. However, this lock creates synchronization points and may make some issues difficult to detect (because the issues may only happen for a particular even sequence, and the extra synchronization points may prevent such a sequence from happening).

This session will show some modularization effort for lockdep. The modularization use a frontend-backend design: the frontend tracks the current held locks for every task/contexts and reports lock dependencies to the backend, and the backend maintains the lock dependency graph and detect lock issues based on what the frontend reports.

Along with the design, a draft implementation will be shown in the session too, providing something concrete to discuss about the design and the future work.

### I agree to abide by the anti-harassment policy

Yes

**Primary author:** FENG, Boqun

**Presenter:** FENG, Boqun

**Session Classification:** LPC Refereed Track

**Track Classification:** LPC Refereed Track