



Contribution ID: 238

Type: **not specified**

A (Simple) Hazard Pointer Update

Saturday 13 December 2025 15:45 (45 minutes)

Following up on the initial hazard pointer demonstration 1, the in-kernel hazard pointers have been significantly improved based on extensive feedback and discussions. A status update is therefore warranted.

Besides some improvement/fixes of the normal hazard pointers implementation, we will demonstrate a variant of hazard pointers – simple hazard pointers (or shazptr) – that offers an intuitive API and can outperform RCU (including expedited RCU) in scenarios prioritizing updater wait times or memory footprint.

Shazptr achieves simplicity through a shared per-CPU hazard pointer slot and a “wildcard” mechanism, eliminating the need to allocate hazard pointer slots prior to use.

Applying simple hazard pointers to lockdep accelerates dynamic key unregistration (~20x speed-up 2), while avoiding additional IPIs. Micro-benchmarks further indicate that shazptr surpasses RCU in updater wait times while maintaining reasonable overhead on the reader side.

We will also cover the future potential of merging shazptr and normal hazard pointers.

Primary authors: FENG, Boqun; MCKENNEY, Paul (Facebook)

Presenters: FENG, Boqun; MCKENNEY, Paul (Facebook)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track