

# Discussing **Softirq** revert and side-effects

Jesper Dangaard Brouer

Netconf  
Paris, Sep 2023

# Background: “Let ksoftirqd do its job”

Eric’s 2016 (LWN) change “softirq: Let ksoftirqd do its job” – Reverted May 2023

- **Issue:** Aggressive **softirqs** often handled by **current process**
  - **Innocent threads cannot make progress**
- **Solution:** if ksoftirq is running, then **skip** `do_softirq_own_stack()`

Patch **solved UDP overload** case

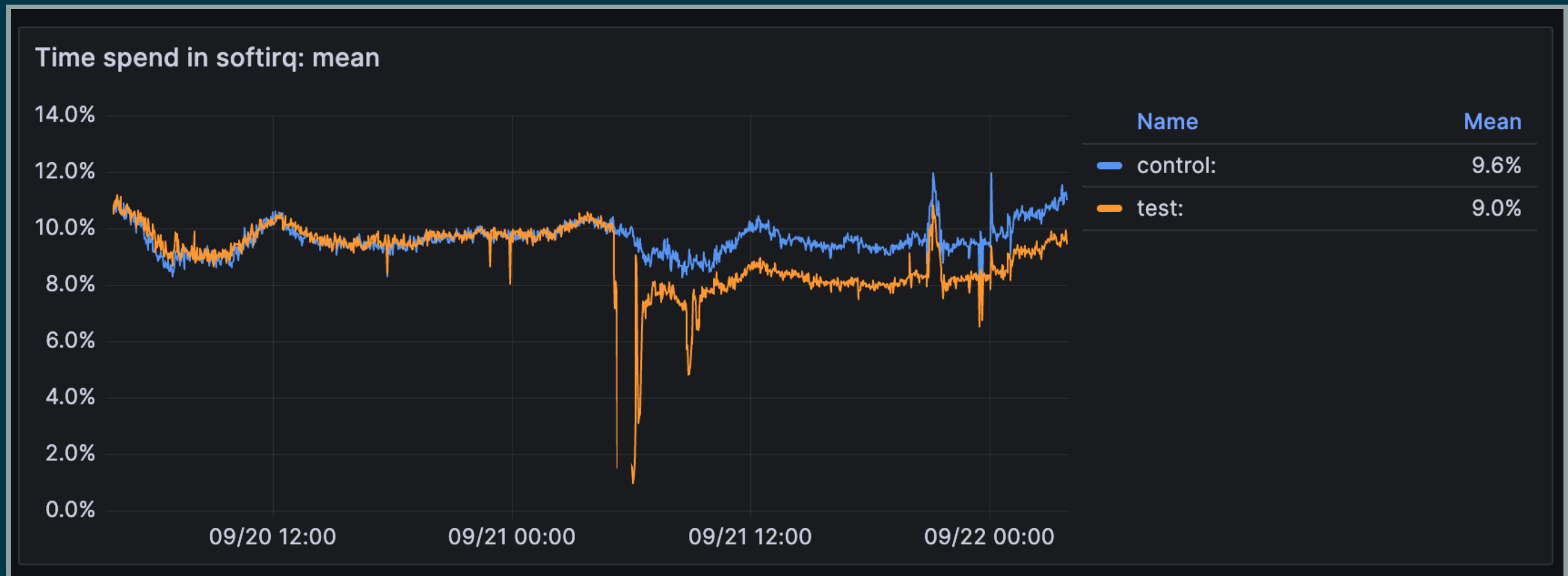
- that resulted in **falling-of-edge** when UDP-app and RX-NAPI **share CPU**

Over the years: several cases **indicate** approach have **caused issues**

- Thus **likely right choice to revert**
- The question is how to address (UDP) overload case in a **new way ?**

# Production benefits? (1/2)

Reduced CPU time spend in **softirq** (?) (test == Kernel 6.1.53 with reverted softirq change)



# Production benefits? (2/2)

CPU usage shifted: **Innocent threads** are likely **running softirq** (?)



# Solve via: Ask users to use other APIs?

The question is how to address UDP overload case in a new way?

The **solutions** could be to **ask users to use other APIs**

- Tried **multi** UDP message **recv** (recvmsg) but does NOT work
- Tried **UDP GRO** but didn't manage to configure it correctly
- Simple **io\_uring** UDP (echo) program didn't solve issue

# End and Thanks

Thanks to Cloudflare Engineers

- Yan Zhai - for providing production data and test
- Marek Majkowski - for providing IO uring test program