RPS, RFS, and SKB lists

David S. Miller

Red Hat Inc.

Seville, Spain, 2010

Before

RPS, RFS, and SKB lists

David S. Miller

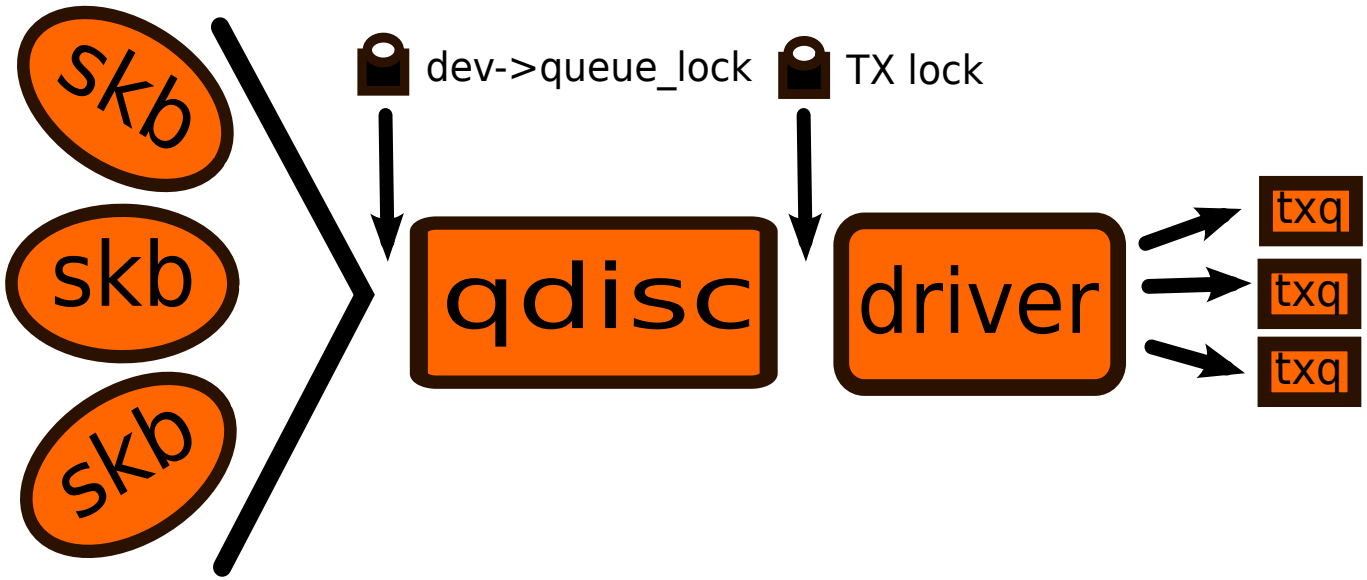
Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head



After

RPS, RFS, and SKB lists

David S. Miller

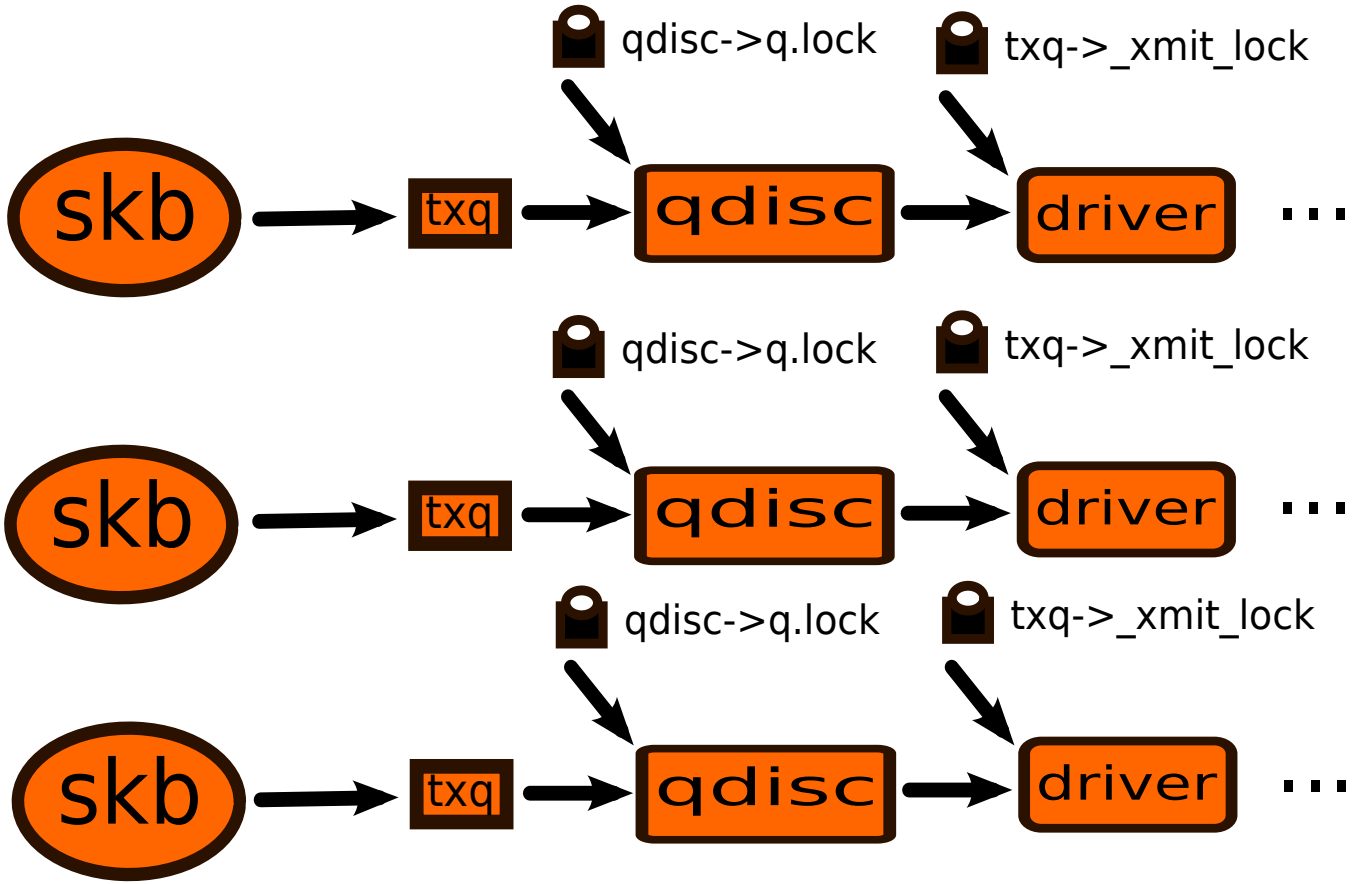
Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head



RPS: Overview

- Dumb software packet steering
- Preserves flow packet ordering
- Legacy hardware
- Sysadmin flexibility
- Met with resistence at first

RPS: Legacy hardware

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Old systems without multiqueue NICs
- Some have ”lots” (f.e. google)
- Older systems benefit ”more”
- But it’s not just about dusty HW

RPS: Flexibility

- Per-queue cpu mask
- Partition traffic arbitrarily
- Even beyond HW capabilities

RPS: Resistence

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- We don't need this
- Everyone will have HW multiqueue
- In particular before SW code is deployed
- This proved to be wrong
- Especially considering RFS

RFS: Overview

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Stateful flow targetting
- Watches I/O calls
- Remembers cpu
- Directs flows to I/O cpu
- Uses hash table

RPS/RFS: What to improve

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- No config automation
- Too much by-hand tweaking
- RPS/RFS even helps loopback
- Need it on by default
- Need reasonable default
- Keeping policy out of kernel, etc.

SKB: Why do this

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- list_head is good
- Understood by many developers
- Makes code more hackable
- Free debugging facilities
- Better integration with other subsys
- Smaller data-structures

SKB: Barriers to Change

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Not just doubly linked lists
- Frag lists, etc.
- Not all use skb_* interfaces
- Last holdout, PPP/ISDN-PPP fragments

SKB: Task One

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Frag lists
- Consistent semantics of next/prev
- All accesses behind skb_* interfaces
- Change sk_buff head members
- Difficult cases (IXGBE RSC, etc.)

SKB: Task Two

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Resolve all remaining by-hand references
- PPP fragmenter code work posted to netdev
- ISDN/PPP fragment code remains
- Analyze the (few) remaining cases

SKB: Task Three

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Convert sk_buff to list_head
- Update all skb_* interfaces to match
- Do the happy dance

SKB: Future

RPS, RFS, and SKB lists

David S. Miller

Multiqueue Impetus

RPS: Receive Packet Steering

RFS: Receive Flow Steering

RPS/RFS: Downsides

SKB: Conversion to list head

- Change sk_buff_head to list_head where possible
- Evaluate remaining sk_buff_head cases
- Investigate new debugging possibilities