# *Linux Networking Futures 2010*

David S. Miller

Red Hat Inc.

Boston, USA, 2010

# RPS



- SMP stateless flow seperation in software
- Hash computed in software
- OR taken from hardware
- Cross-calls used to doorbell remote cpu queues

# RFS



- More sophisticated
- I/O calls monitored
- CPU of I/O call recorded in hash table
- Recorded CPU directs future packet steering
- Out-of-order issues avoided during hash changes

# XFS



- I will leave this to Tom.

# State



- RPS/RFS fully merged and functional
- XFS is in review state, 2.6.38 likely
- Unfortunately RPS/RFS not enabled by default
- Hey, it helps “loopback”
- Lack of infrastructure

# SKB list_head: How close?



- We're almost there.
- Abstractions exist for all uses
- Even for frag_list
- PPP, ISDN-PPP, LRO driver code, and GRO

# SKB list_head: Next steps



- Test and merge PPP/ISDN-PPP conversion
- Move forward with ixgbe et al. frag_list conversions
- Pull the trigger
- Fix the bugs
- Pop the champagne

# Iteration Elimintation: netif_rx() sucks



- Transmit path of loopback and tunnels
- Queue packet, signal interrupt
- Expensive way to avoid stack overflow

# Iteration Elimintation: Quick hacks



- netif_receive_skb() with recursion limit
- Eric Dumazet
- lock_sock sets mark
- netif_rx() queues when mark set
- release_sock runs packets in queue
- Issue: What is we sleep with socket lock?

# Iteration Elimintation: More Formal



- net_start_transmit()
- Acts like lock_sock idea above
- But never done before sleep'able section
- Finish with net_end_transmit()
- Runs queue etc.
- State is cpu-local, like current backlog

# RTCACHE ELIMINATION: WHY NOW?



- Eric Dumazet's optimizations
- fib_trie faster than routing cache
- rtcache non-determination
- rtcache complexity

# RTCACHE ELIMINATION: MAIN PROBLEM



- Metrics
- Flow based metrics exist only in routing cache
- As do path based metrics (PMTU, redirects, etc.)
- Supplement with inetpeer cache?
- New metric cache by flow?
- Issue of IPSEC based metrics?