Networking and Multicore/NUMA – The Last Mile

Herbert Xu
Red Hat Inc.

Background

- CPU population explosion:
  - Multicore
  - NUMA
- CPUs don't like working together.
- Linux needs to keep the peace.
- Global state is bad!

Multicore and Multiqueue

- Multicore is similar to SMP, needs locking.
- High lock contention reduces CPU efficiency.
- Solution: multiqueue NICs
  - Each core has its own queue and interrupt.
  - Transmit: CPU chooses queue.
  - Receive: NIC chooses queue.

Support for Multiqueue

- Full support in driver API.
- Receive Packet Steering.
- Support for hardware flow steering (manual).
- Default packet scheduler only.
- TX queue selection by hash.

Network Stack

- Driver
- Packet scheduler
- Layer 2 (e.g., bridge or VLAN)
- IP (routing, netfilter)
- TCP/UDP
- Application

Packet Scheduler

- Hardware assistance:
  - Priority-based queueing.
- Change in semantics:
  - RED, GRED, etc.
- Go back to a single queue!
  - Separate data from metadata.

Layer 2

- VLAN is trivial.
- Bridge:
  - Unicast MAC lookup
  - Multicast MAC lookup
  - netfilter

IP

- Routing:
  - Kill route cache?
  - Local route cache?
  - Need global metrics.
  - Hybrid of local route cache + global metrics.
- netfilter:
  - Connection tracking.

TCP/UDP

- Socket lookup:
  - Fast path uses local lookup table.
  - Fallback global lookup table.
- TX/RX CPU/queue assignment must match.
- UDP needs to become lockless again.

Application

- Multi-threaded.
- NUMA-aware.
- Thread/socket/CPU/queue alignment.

To Do

- Audit all counters.
- Local route cache.
- Thread/socket/CPU/queue alignment.
- Bridge unicast address lookup.
- Packet schedulers.
- Virtualisation.

Questions