10GbE on Linux

Herbert Xu
Red Hat Inc.

Background

- Ethernet bandwidth from 10Mb/s to 10Gb/s.
- MTU (Maximum Transmit Unit) 1500 bytes.
- Packet rate from 833pps to 833000pps.
- CPU moving to multicore and NUMA.
- Challenges:
  - Packetisation.
  - Synchronisation.

Raising MTU

- Larger MTUs through Jumbo Frames.
- Standard jumbo frames are only 9000 bytes.
- Lowest MTU over whole path applies.
- PMTU discovery determines MTU.
- PMTU discovery works poorly over Internet.

TCP Segmentation Offload

- Content providers biased towards sending.
- TSO raises MTU within host to 64KB.
- Resegmentation before wire.
- Sufficient for 10GbE.
- IPv6 jumbograms for MTUs above 64KB.
- Linux has Generic Segmentation Offload.

Large Receive Offload

- NIC still receives ~1 million packets.
- LRO merges packets upon entry into stack.
- Larger internal MTU as TSO.
- Widely supported by 10GbE drivers.

LRO Problems

- Packet merging doesn't preserve all state.
- Incompatible with packet forwarding.
- Incompatible with virtualisation.
- LRO limited to TCP over IPv4.

Generic Receive Offload

- GRO restricts what can be merged:
  - Identical MAC header.
  - Only certain IP header fields can differ.
  - Only certain TCP header fields can differ.
- Merged packet can be resegmented losslessly.
- GRO reuses GSO infrastructure.

GRO Status Quo

- Supported by many (but not all) 10GbE drivers.
- Complete conversion of remaining drivers.
- Address any performance regressions.
- Eventual removal of LRO.

Future Work

- Generic flow-based merging:
  - TCP ACK merging.
  - Linked list of skb's per IP flow.
  - Merging UDP and other protocols.
- Reuse hardware receive hash in GRO.
- Emulate multiqueue receive in software.

Multi-core and Multiqueue

- Multi-core is similar to SMP, needs locking.
- High lock contention reduces CPU efficiency.
- 10Gb/s cannot afford reduced efficiency.
- Solution: multiqueue NICs
  - Each core has its own queue and interrupt.
  - Transmit: CPU chooses queue.
  - Receive: NIC chooses queue.

Support for Multiqueue Receive

- NIC decides which queue to use.
- Usually done by hashing the packet header.
- Only needs to modify driver to support this.
- Multiqueue NAPI requires stack modifications.
- Oct 07: Multiqueue NAPI support added.

Support for Multiqueue Transmit

- July 08 (David S. Miller):
  - Default qdisc (pfifo_fast):
    - Each netdev corresponds to many qdiscs.
    - Each qdisc corresponds to a hardware queue.
  - All other qdiscs remain as before.
- Resolves qdisc lock contention for default qdisc.

Future Work

- Queue selection for local traffic.
- Eliminate remaining shared state.
- NUMA scalability.
- Support for qdiscs other than default:
  - Need to add multiple queues within each qdisc.
- Virtualisation and user-space networking.

Questions