PostMan: Rapidly Mitigating Bursty Traffic by Offloading Packet Processing

Panpan Jin\textsuperscript{1}, Jian Guo\textsuperscript{1}, Yikai Xiao\textsuperscript{1}, Rong Shi\textsuperscript{2}, Yipei Niu\textsuperscript{1}, Fangming Liu*\textsuperscript{1}, Chen Qian\textsuperscript{3}, Yang Wang\textsuperscript{2}

\textsuperscript{1}OpenCloudNeXt Group, Huazhong University of Science and Technology
\textsuperscript{2}The Ohio State University
\textsuperscript{3}University of California Santa Cruz

OpenCloudNeXt Group: http://grid.hust.edu.cn/fmliu
Bursty traffic is a headache!

Increase of conversation rate in 24h

- Cyber Monday: 35%
- Black Friday: 42%

- 1,480,000,000 deals in total
- 42,000,000 queries/s at peak
- 256,000 deals/s at peak
Bursty traffic is a headache!

- Large volume
- Short duration
- Small packets
- Severe overhead

Payload size breakdown:
- [0, 31 bytes]
- (31 bytes, 41 bytes]

Packet processing throughput (Gbps):
- 10 Gb Linux: 10 Gbps
- 10 Gb IX: 5 Gbps
- 64 bytes: 1 Gbps
- 64 KB: 10 Gbps
Traditional remedy: migrating hot data for load balancing

Clients -> Memcached servers

- Time-consuming
- Exacerbate overload
- Packet processing overhead
PostMan: batching and offloading on demand

Clients → Helper servers → Memcached servers

- Server with normal load
- Server experiencing bursty traffic

Helper assembles small packets into large ones

- Large packets
- No data migration
- Rapid mitigation
PostMan: are helpers efficient?

Clients — Helper servers — Memcached servers

- mTCP
- DPDK & RSS
- Server experiencing bursty traffic
PostMan: are helpers efficient?

DPDK & mTCP based stack
Efficient packet processing

Server experiencing bursty traffic

Memcached servers

Clients

Helper servers

mTCP

TCP/IP/MAC header

Payload
PostMan: are helpers efficient?

Clients → Helper servers

DPDK & RSS → mTCP

PayLoad of assembled packet

PostMan header

mTCP

Memcached servers

Server experiencing bursty traffic

☑️ DPDK & mTCP based stack
Efficient packet processing

☑️ Remove duplicated headers
Reduce bandwidth consumption
PostMan: are helpers fault-tolerant and scalable?

Clients -> Helper servers -> Memcached servers

Free connection migration ✅ Stateless ✅ No scalability bottleneck ✅

Server with normal load
Server experiencing bursty traffic
PostMan vs. Data migration: rapid and efficient

Mitigation time: **550 ms** vs. 13s

Throughput: **2.8×**
Thank you!

Track II: Networking

9:35 AM, Friday July 12, 2019