Stateless Datacenter
Load-balancing with **Beamer**

Vladimir Olteanu, Alexandru Agache,
Andrei Voinescu and Costin Raiciu
*University Polytehnica of Bucharest*

Presented by Pascal Wiesmann
20.4.2018
Presentation overview

• What is the problem?
• How is it solved?
  i.e. Contributions of the paper
• Evaluation
Goal: Better load balancing

• “[…] steer traffic to a given service across a dynamic set of backend machines.”

• Fast
• Robust (against attacks)
• Stable (no big fluctuations)
Unhappy Customers

== less money
Cheap hardware == more money
Load Balancing: Existing Approaches
keeping per-flow state on the muxes
Problems of existing approaches

SYN floods

Scale-out
Beamer: stateless load-balancing

• Contributions:
  • Muxes without per-flow state
  • Stable hashing
  • Daisy chaining
  • Handling Multipath TCP
Stable Hashing

Trivial: $\text{hash}(5\text{tuple}) \% \#\text{servers}$
New: $\text{hash}(5\text{tuple}) \% \#\text{buckets}$ requires synchronization of bucket-to-server mapping between muxes!

- Fixed number of buckets

a) Stable hashing with three servers

b) Server B fails
Daisy Chaining

a) Buckets belong to server A

b) Buckets moved to B, Inconsistent mappings.
Multipath TCP (MCTCP)
Mux behavior and state

```c
packet* mux(packet* p) {
    if (p->dst_port<1024) {
        gen = buckets.version;
        b = hash(5-tuple) % B;
        dip = buckets[b][0];
        pdip = buckets[b][1];
        ts = buckets[b][2];

        return encapsulate(mux, dip, pdip, ts, gen, p);
    }
    else {
        dip = id[p->dst_port];
        return encapsulate(mux, dip, p);
    }
}
```
Benchmarks

“[...] our software implementation of Beamer is twice faster than Google’s Maglev, the state of the art software load balancer, and can process 40Gbps of HTTP uplink traffic on 7 cores.”

Figure 14: Forwarding performance (ten-core Xeon, 4 x 10Gbps). Beamer forwards 40Gbps with 128B packets.
More Benchmarks

• Faster than (own) stateful Baseline
• Mux latency: < 0.3ms
• Handles Mux churn smoothly
Benchmarks: MCTCP

Figure 19: Flow completion times for MPTCP clients using Beamer

Figure 20: Flow completion times for MPTCP clients using Stateful.
Benchmarks: Controller update

Figure 24: Time to propagate a controller update from ZooKeeper to all the muxes.
Beamer: Conclusion

• Pros:
  • Good connection affinity for TCP
  • Robust against SYN floods
  • Fast
  • Supports MPTCP
  • Open Source

• Possible Weaknesses
  • P4 not really tested
  • How important is the problem? How often does mux and server churn happen in practice?
  • Is the 40% speed improvement also observable in real world?
Paper Review

• Pro:
  • Well structured
  • Good explanations of the algorithms they introduce

• Con:
  • No “future work” section
  • Evaluation section unclear
  • Benchmark plots hard to understand
  • Only test against their own baseline implementation
That's all Folks!
Stateless Datacenter
Load-balancing with Beamer

Questions?