CONGA: Distributed Congestion-Aware Load Balancing for Datacenters

By Alizadeh, M et al.

Presented by Andrew and Jack

Motivation

Distributed datacenter applications require large bisection bandwidth

ECMP: hash-based hop selection without reordering

\[ F(sIIP, sPort, dIIP, dPort, prot) = 0 \]
ECMP Problem: hash collisions lead to imbalance

\[ F(0, 0, 3, 0, \text{TCP}) = 0 \]
\[ F(0, 1, 4, 0, \text{TCP}) = 0 \]

ECMP Problem: local decisions oblivious to downstream asymmetry

MPTCP: split flows into sub-flows

MPTCP Problem: higher congestion at edge
**MPTCP Problem: transport layer-specific**

**CONGA: Congestion Aware Balancing**

Network load-balancing without transport layer interference

Make globally optimal load-balancing decisions

Use common datacenter network features (e.g., overlay networks)

**CONGA Overview**

Track end-to-end congestion along path

Feedback loop between leaf switches: relay congestion information

Leaf switches send traffic on least congested path

**CONGA Design Goals**

1. Responsive
2. Transport independent
3. Robust to asymmetry
4. Incrementally deployable
5. Optimized for Leaf-Spine
Distributed load-balancing is highly responsive, near optimal for regular topologies.
Global congestion awareness is necessary to handle network asymmetry.

(a) Static (ECMP)
(b) Congestion-Aware: Local Only
(c) Congestion-Aware: Global (CONGA)
Overlay networks allow leaf switches to know destination leaf and carry congestion metrics.

Overlay networks allow leaf switches to know destination leaf and carry congestion metrics.

Overlay networks allow leaf switches to know destination leaf and carry congestion metrics.

Overlay networks allow leaf switches to know destination leaf and carry congestion metrics.
Overlay networks allow leaf switches to know destination leaf and carry congestion metrics.

Packet-granularity scheduling can result in reordering → modifications to end-host TCP.
Packet-granularity scheduling can result in reordering → modifications to end-host TCP
Flowlet: break apart flow based on delayed bursts

Route separately if \( \text{gap} > |d_1 - d_2| \)

Overlay packet header contains CONGA metadata

- **LBTTag**: src port
- **CE**: path congestion
- **FB_LBTTag**: fb dst port
- **FB_Metric**: reported cong.
Discounting Rate Estimator (DRE)

$X$: register quantifying load
- Additive increase by bytes sent for each packet
- Multiplicative decrease every $T_{dre}$ by $\alpha$

$$X \leftarrow X \times (1 - \alpha)$$

More responsive to traffic bursts than EWMA
Flowlet Detection

$T_{fl}$: flowlet inactivity gap

Hash flowlets based on 5-tuple
  Collision is not a correctness issue

Round-based aging

LB decisions made based on first packet

New flowlet: choose uplink minimizing the max local metric

Implementation: custom ASICs rather than software to reduce overreaction, oscillations
Evaluation

1. How does CONGA impact flow completion times (FCT) vs. state of the art?
2. How does CONGA perform under the impact of failed links?
3. Does CONGA perform well on real-world traffic?

Experimental Setup

Compared CONGA, CONGA-FLOW, ECMP and MPTCP

Baseline Performance

Two Workloads: Enterprise and Data-mining

Breakdown: Short Flows and Long Flows
Link Failure

HDFS Benchmark

Incast

Analytical Evaluation

Worst-Case performance: The ratio between the most congested link in CONGA and the best possible assignment of flows is 2.
Analytical Evaluation

What is the expected traffic imbalance?
How does it depend on workload?

\[
E(\chi(t)) \leq \frac{1}{\sqrt{\lambda_e t}} + \Theta\left(\frac{1}{t}\right),
\]

where:
\[
\lambda_e = \frac{\lambda}{8n \log n \left(1 + \frac{\sigma^2}{E(S)}\right)^2}.
\]

Conclusion

CONGA: globally aware datacenter load balancing
- No transport layer intervention

Implemented in custom ASICs

Better flow completion times than ECMP, Incast MPTCP

Discussion

Leaf-Spine topology has each leaf only two hops apart
- Significant performance degrade if implemented in software?
- Extensible to larger, multi-layered topologies?

Less imbalance with many small flows, more imbalance with fewer large flows