How Container Networking Affects Database Performance

Vadim Tkachenko and Tyler Duzan

KubeCon & CloudNativeCon 2019
Wednesday, November 20th, 2019

PERCONA

Who is Percona

- **Open Source Databases Experts**
  - “Unbiased Champions of Open Source Databases”
- **Open Source Software and Services for**
  - MySQL
  - PostgreSQL
  - MongoDB

Who Are We?

- Vadim Tkachenko
- Chief Technology Office and Co-Founder of Percona
- Widely regarded expert on database engine internals and performance
- Co-Author of “High Performance MySQL”

Who Are We?

- Tyler Duzan
- Product Manager for MySQL Software and Cloud at Percona
- Formerly spent 12 years as an operations/cloud focused engineer

Percona and Kubernetes

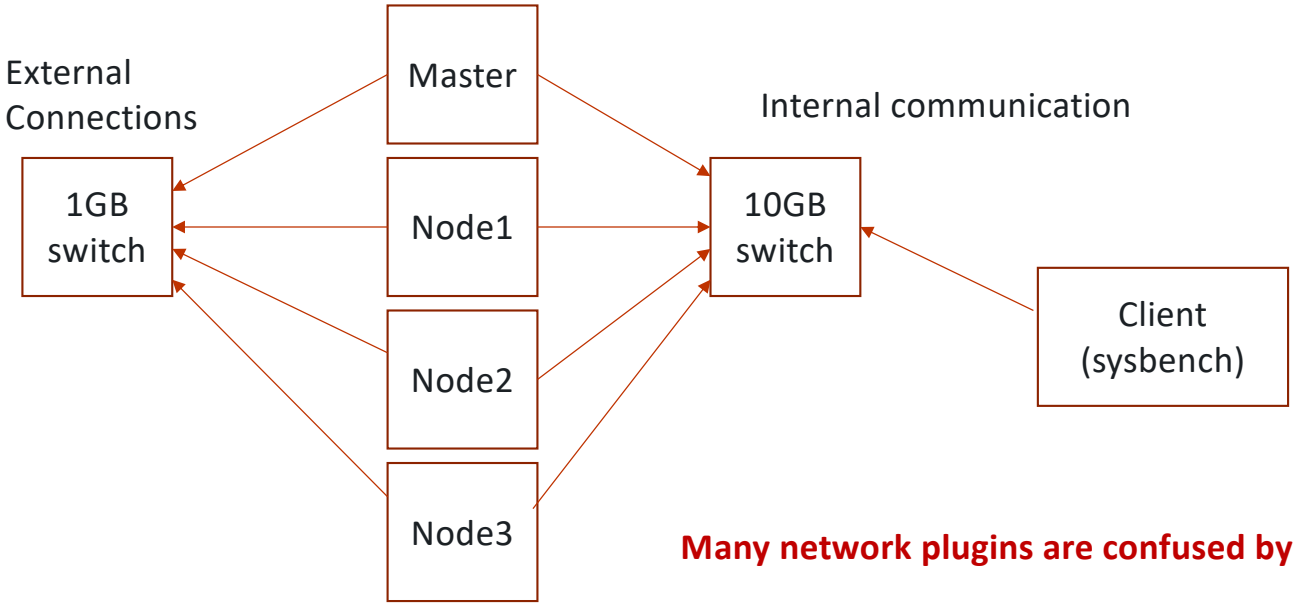
- **Making a bridge for Databases**
  - From “Don’t do this” to “Let’s do this”
- **This talk is not to blame or do finger pointing, but to share our adventure into databases and networking in Kubernetes**

Talk Background

- Percona launched a Kubernetes Operator for Percona XtraDB Cluster (MySQL clustering) and for Percona Server for MongoDB as GA in May 2019 at Percona Live
- We've noticed significant performance differentials while working with customers who are migrating workloads into Kubernetes
- Many of our customers currently run apps in Kubernetes and databases on bare metal outside their Kubernetes cluster
- We performed benchmarking to understand the relationship between networking performance in Kubernetes and database performance.

Benchmark Methodology

Hardware layout

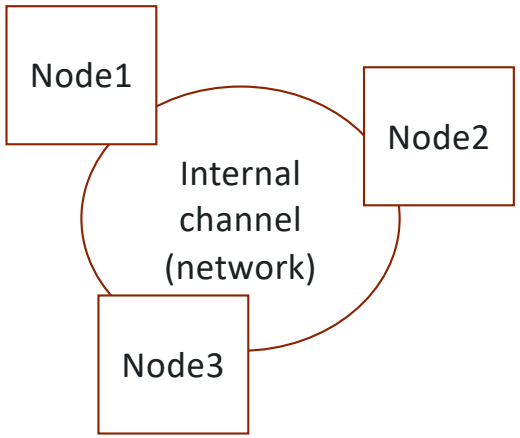


Many network plugins are confused by 2 network cards

Database – Percona XtraDB Cluster

What is Percona XtraDB Cluster?

- MySQL nodes working together synchronously
- 3 nodes recommended minimal setup
- Allows to achieve High Availability and Data Consistency

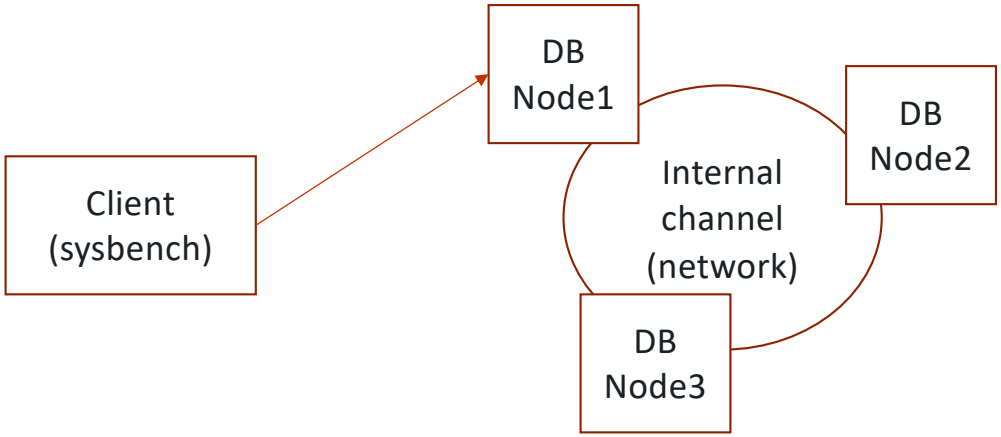


Synchronous commit requires high performance network

- Latency – transaction latency depends on network latency
- Throughput – transaction throughput depends on network throughput

Storage is another important factor for the performance.
Shared storage increases demand on network

Database layout



Benchmark Methodology

- Benchmarks:
  - Database: sysbench with the oltp-read-write workload
  - Network bandwidth: iPerf3 to capture network throughput
- Baseline benchmark was taken against a 3-node PXC cluster running on bare metal
- Network cards:
  - Intel X540-AT2 and Intel I350 NICs (2x 10GigE, 2x Gbit)
- OS: Ubuntu 16.04 LTS, uname below
- Linux nodesm 4.15.0-66-generic #75~16.04.1-Ubuntu SMP Tue Oct 1 14:01:08 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux

Benchmark Methodology

- Kubernetes was deployed using kubeadm with 3 Nodes and 1 Master
- CNI plugins were installed following their provided installation instruction, such that the 10GigE interface was used for traffic
- Database / Kubernetes Servers contain 2x Intel Xeon E5-2680v3 (12C/24T), 128GB of DDR4 2133 RAM, and 2x 256GB Samsung 850 Pros in RAID 1 (md)
- Sysbench Servers contain 2x Intel Xeon E5-2683v3 (14C/28T), 128GB of DDR4 2133 RAM, and 2x 256GB Samsung 850 Pros in RAID 1 (md)

Benchmark Methodology

- For bare metal tests, we installed PXC 5.7.27-31.39 using system packages from the Percona public repos
- For Kubernetes tests, we installed using our 1.2.0 release of the Percona Kubernetes Operator for PXC which ships 5.7.27-31.39
- In both cases, a tuned my.conf was utilized, which was provided as a ConfigMap in Kubernetes
- In both cases we utilized local storage on the SSD RAID1, which was mapped via HostPath method in Kubernetes

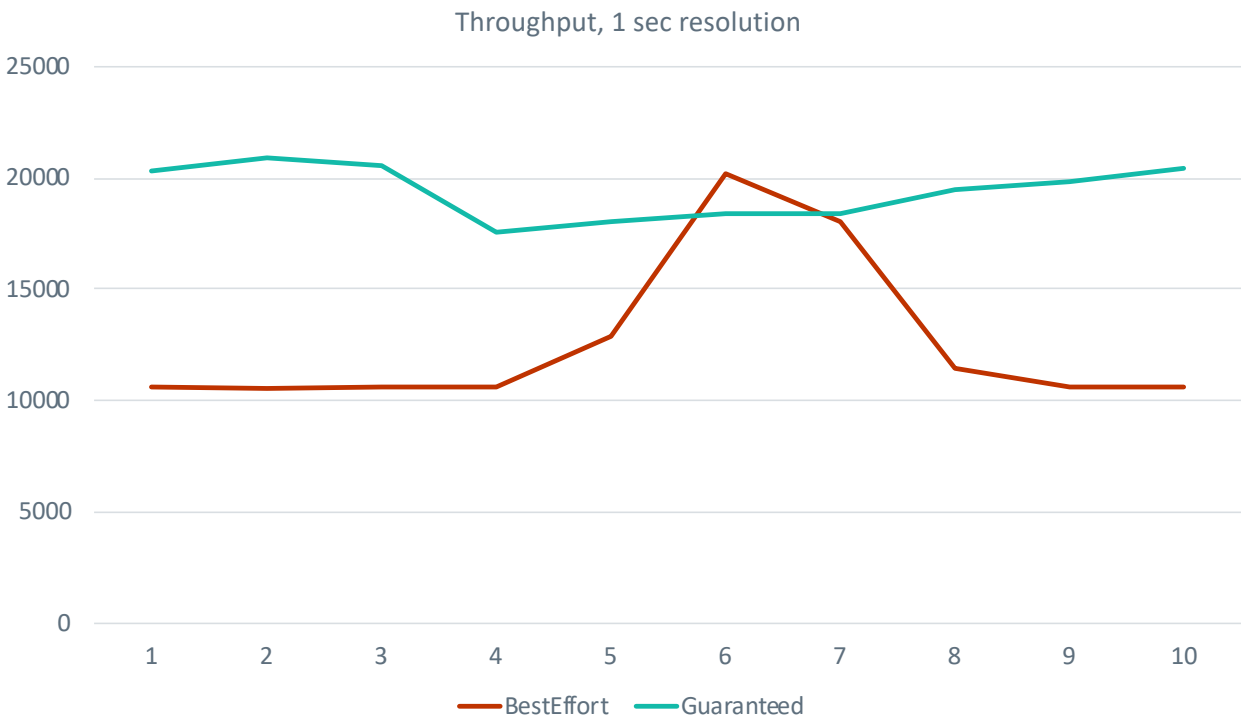
Benchmark Methodology

```
[mysqld]
table_open_cache = 200000
table_open_cache_instances=64
back_log=3500
max_connections=4000
innodb_file_per_table
innodb_log_file_size=10G
innodb_log_files_in_group=2
innodb_open_files=4000
innodb_buffer_pool_size=100G
innodb_buffer_pool_instances=8
innodb_log_buffer_size=64M
innodb_flush_log_at_trx_commit  = 1
innodb_doublewrite=1
innodb_flush_method          = O_DIRECT
innodb_file_per_table        = 1
innodb_autoinc_lock_mode=2
innodb_io_capacity=2000
innodb_io_capacity_max=4000
wsrep_slave_threads=16
wsrep_provider_options="gcs.fc_limit=16;evs.send_window=4;evs.user_send_window=2"
```

```
resources:
    requests:
      memory: 150G
      cpu: "55"
    limits:
      memory: 150G
      cpu: "55"
```

Guaranteed QoS

Guaranteed QoS vs BestEffort QoS



PERCONA

Benchmark Methodology

- Sybench command line:
  - ./sysbench --test=tests/db/oltp.lua --oltp_tables_count=10 --oltp_table_size=10000000 --num-threads=64 --mysql-host=172.16.0.4 --mysql-port=30444 --mysql-user=root --mysql-password=root_password --mysql-db=sbtest10t --oltp-read-only=off --max-time=1800 --max-requests=0 --report-interval=1 --rand-type=pareto --rand-init=on run

CNI Network Plugins Tested

Calico

- “Project Calico is designed to simplify, scale, and secure cloud networks.”
- Calico supports IP in IP, testing was conducted with IPIP and without. It is enabled by default

PROJECT CALICO

Flannel

- ”Flannel is a network fabric for containers, designed for Kubernetes”
- Flannel supports several “backends”, the default is UDP, and we also tested vxlan and attempted to test host-gw
- We were unable to successfully setup host-gw in our environment in the time we had allotted.
- vxlan performance was good on average, but had many many stalls which we suspect may be caused by a Linux kernel issue

Cilium

- Cilium is “API aware networking and security using BPF and XDP”
- Installation was very straightforward, however we were unable to get Cilium to route traffic across the 10GigE network.
- We are reporting results, but these are obviously going to be heavily impacted by the above issue.

Weave (weave-net)

- Weave is “Simple, resilient, multi-host container networking and more”
- Tightly integrates into Kubernetes to provide “invisible networking”
- Despite showing that Pods were assigned IPs in the netblock associated to our 10GigE interface, traffic was routed across Gigabit. Significant troubleshooting was unable to resolve.
- We are reporting results, but these are obviously impacted by this issue.

SR-IOV w/ Multus

- Multus ”enables attaching multiple network interfaces to Pods in Kubernetes”
- intel/sriov-cni provides the CNI driver for SR-IOV
- intel/sriov-network-device-plugin dynamically creates configuration for sriov-cni based on VFs allocated on the host
- intel/multus-cni adds Pod interfaces

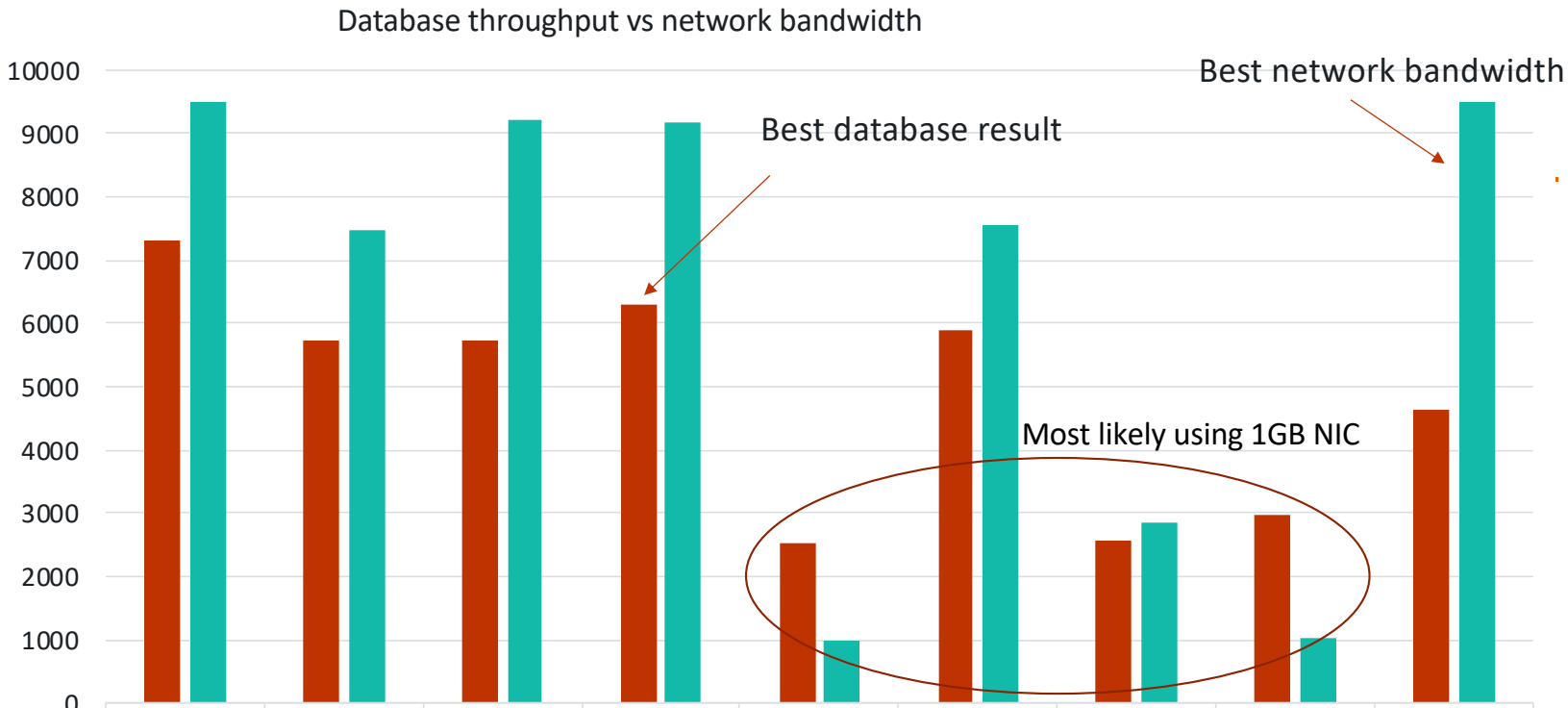
Kube-Router

- Kube-Router is “a turnkey solution for Kubernetes networking”
- Kube-router is very simple to deploy
- Generally worked flawlessly

Benchmark Results

Database throughput on bare metal

- Over 1Gb network: 2700 tps
- Over 10Gb network: 7302 tps
- Proof that Database throughput is limited by 1Gb network



	Bare metal	Calico, default	Calico, no IPIP	Kube-router	Flannel default	Flannel, vxnet	Cilium	Weave	SR-IOV
database throughput, tps	7302	5747	5738	6281	2538	5885	2571	2971	4646
network bandwidth, Mbits/sec	9490	7470	9200	9160	978	7560	2860	1030	9480

Results - Conclusions

- Even the best result with kube-router is ~13% performance penalty compared to bare metal
- Some network plugins may require extra efforts to make them work as expected
- There is a strong correlation (~0.89) between network performance and database transaction throughput

Questions?

Champions of Unbiased
Open Source Database Solutions


KubeCon
CloudNativeCon
North America 2019