Characterizing network performance of single-node large-scale container deployments

Conrado Boeira, Miguel Neves, Tiago Ferreto, Israat Haque

DALHOUSIE UNIVERSITY

PUCRS

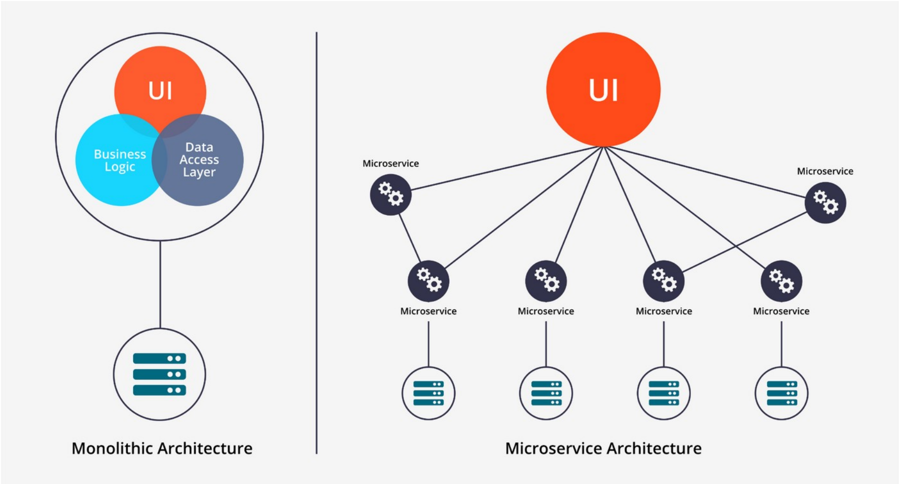
PINet Programmable and Intelligent Networking

Outline

- ❑ Motivation
- ❑ Related Work
- ❑ Methodology
- ❑ Results
- ❑ Final Considerations

Motivation

- ❑ Cloud services are shifting from monolithic architectures to microservices
- ❑ Containers widely used in microservice deployment
- ❑ Large number of containers running on a single server[1]
- ❑ Need for minimal performance degradation on intra-host communication



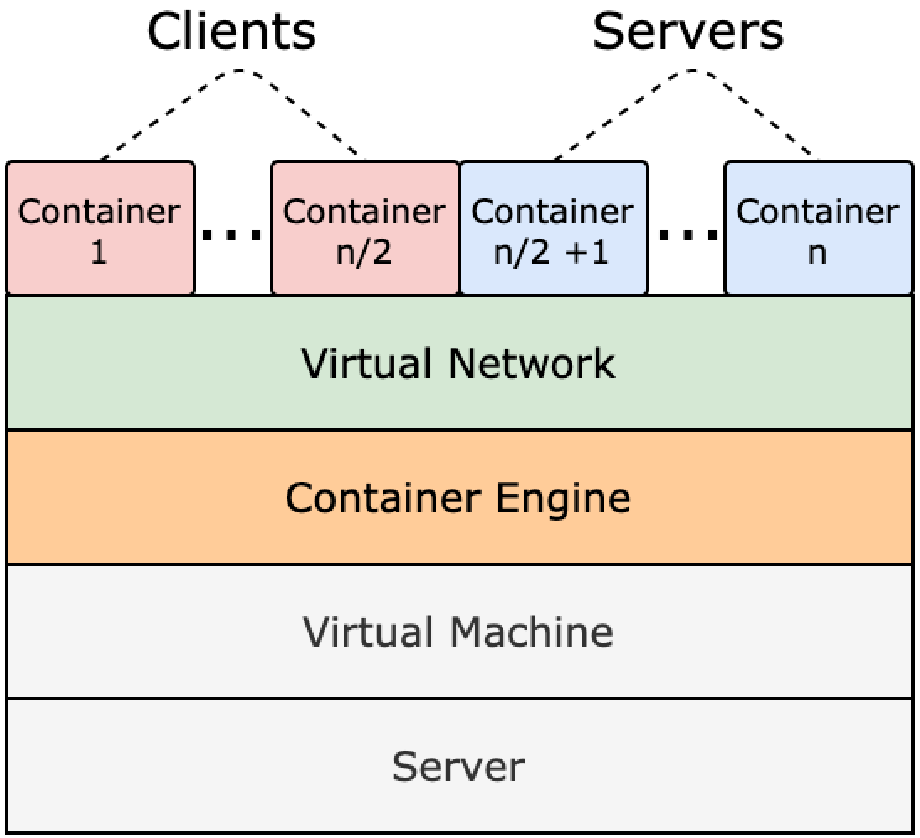
[1] Gan, Yu, et al. "Seer: Leveraging big data to navigate the complexity of performance debugging in cloud microservices." Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems. 2019.

Related Work

- **Zhao et al. (HotConNet 2018)**
  - Multiple allocation scenarios (e.g., on the same VM, same host or different hosts)
  - Single pair of containers
- **Suo et al. (INFOCOM 2018)**
  - Evaluate the influence of packet size, network and CPU interference on container performance
  - Up to 8 container pairs
- **Mentz et al. (CloudNet 2020)**
  - Multiple allocation, workloads and congestion scenarios
  - Two pairs of containers
- **Our work**
  - **Single-node** allocation, multiple workloads and congestion scenarios
  - Up to **50** container pairs

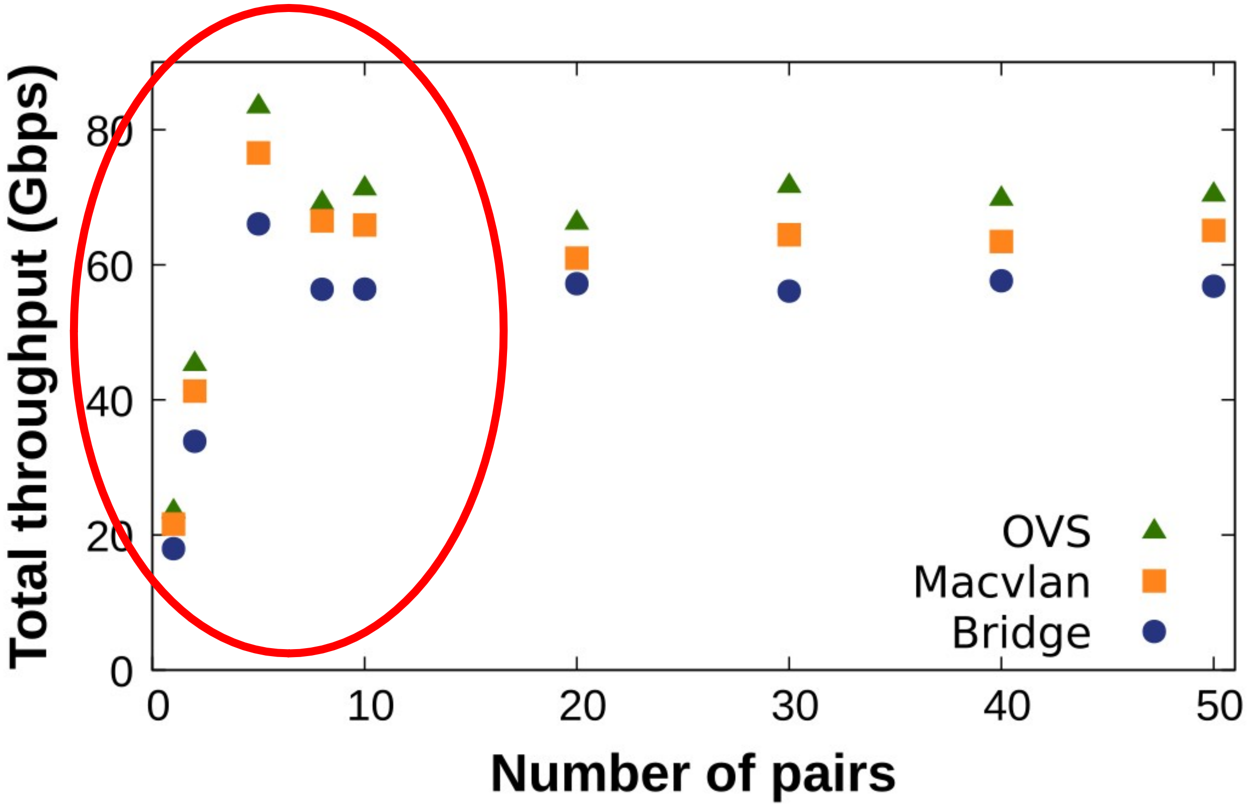
Methodology

- **Testbed**
  - Single server running an Ubuntu VM
  - Docker as the container engine
- **Workload**
  - Throughput stress tests (Iperf, sockperf)
  - Fixed-size (5 MB) file transmissions (Netcat)
- **Metrics**
  - Throughput
  - Flow-Completion Time (FCT)
- **Network drivers**
  - Linux Bridge
  - Macvlan
  - Open vSwitch (OVS)



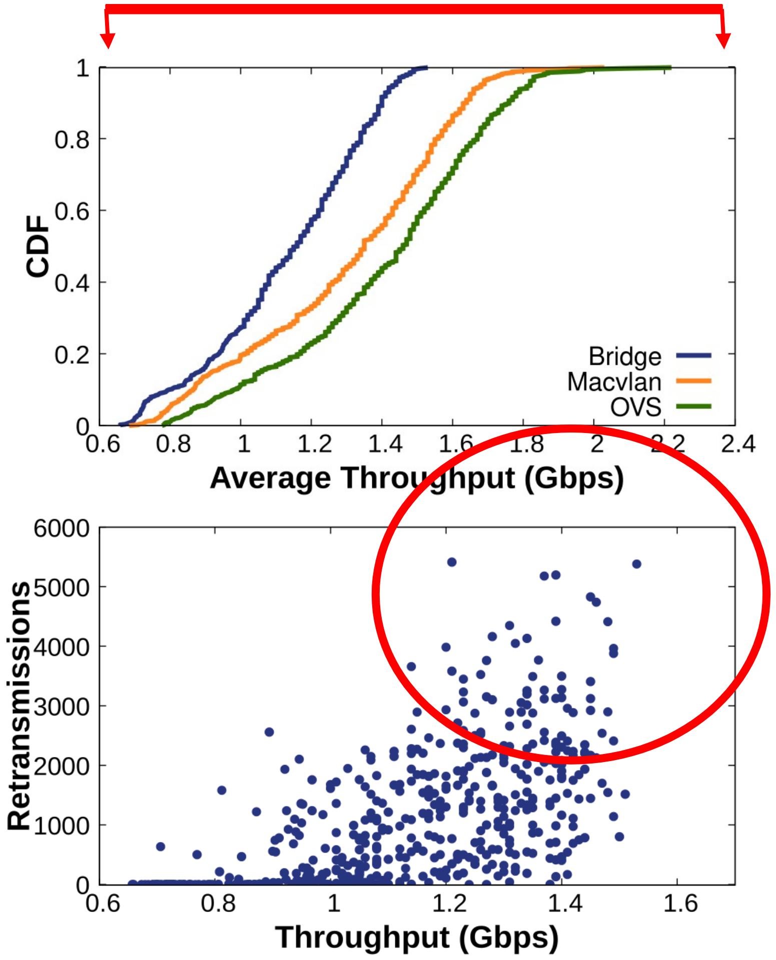
Overall Performance

- ❑ OVS presents the best performance due to the *fast-path* mechanism
- ❑ Throughput increases up to 5 pairs, then stabilizes
    - ❑ Performance restricted by the CPU capacity



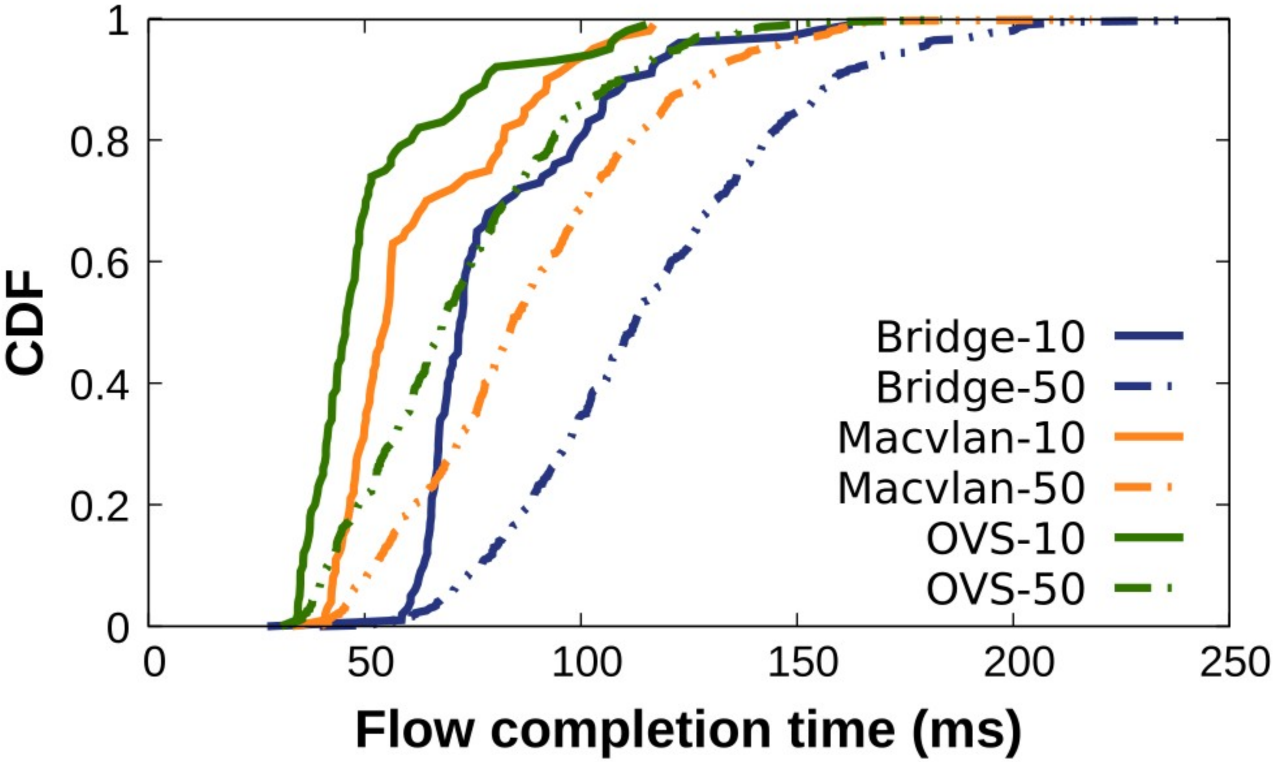
Network Interference among containers

- Throughput can vary up to 3x for all drivers
  - Unfair resource sharing
- Clients with more retransmissions achieve higher throughput values
- Scarce CPU resources lead to unequal opportunities to transmit packets



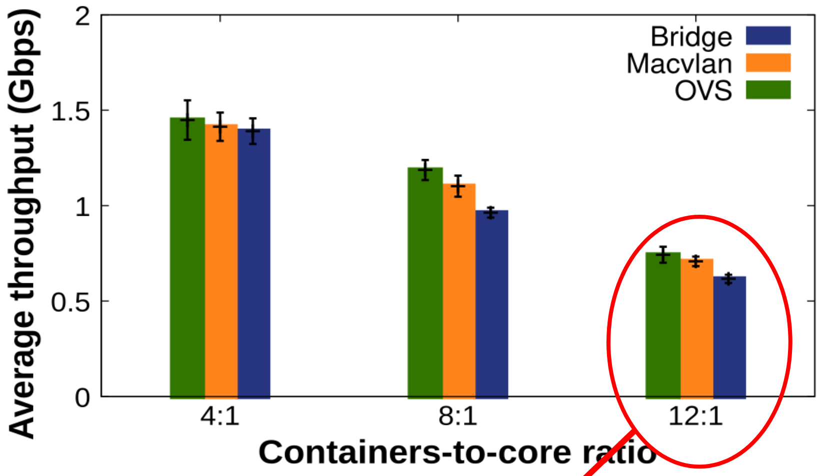
Network Interference among containers

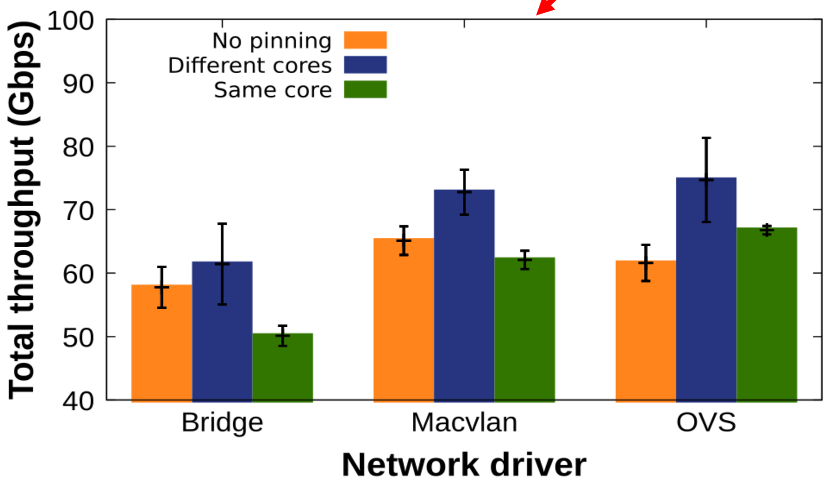
- ❑ Large difference in performance for concurrent flows
  - ❑ up to 5x depending on the networking driver
  - ❑ Result of intense CPU contention
- ❑ High increase in the FCT as the number of pairs increase
  - ❑ >30% for OVS (99th-percentile)



CPU Contention

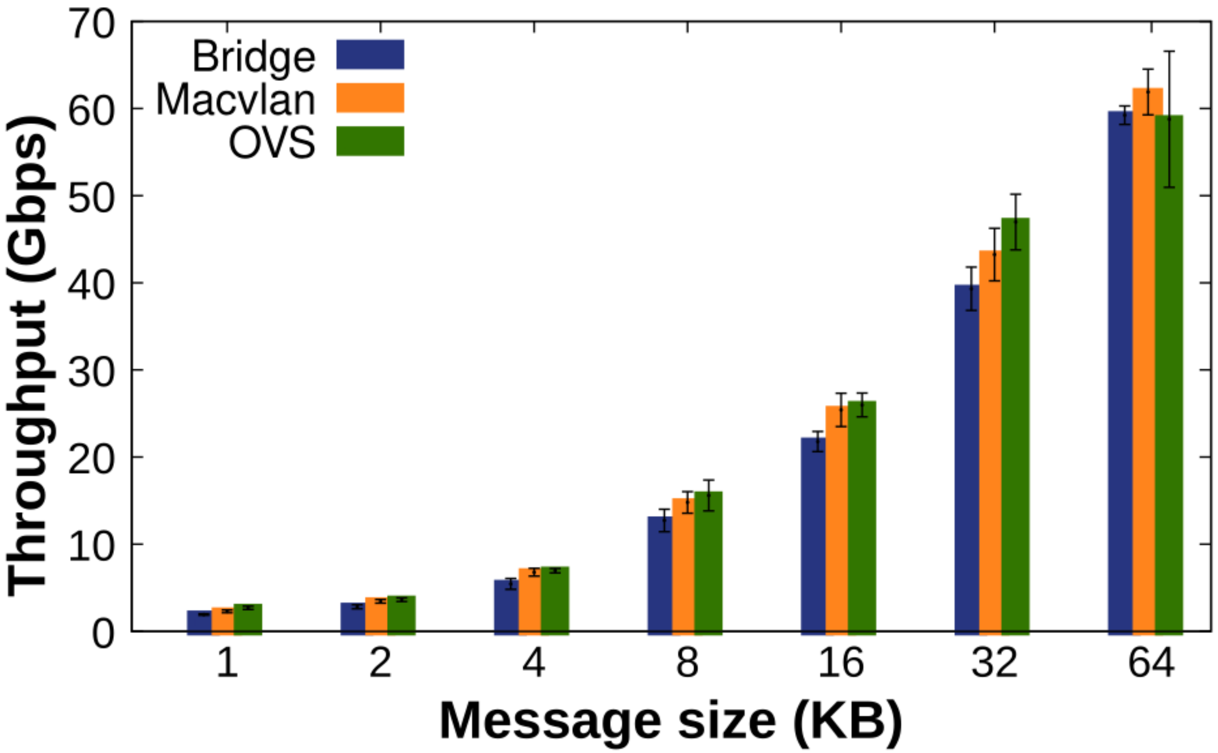
- Throughput greatly decreases with higher container-to-core ratios
  - Smaller CPU slices per container
  - More interrupts and context switches
- Pining containers to cores can help
  - Reduce context switches
- Pining client and server in the same core was detrimental
  - Limits concurrency





Packet Size

- ❑ Larger message sizes result in higher throughput
  - ❑ Less packets to send
  - ❑ Less forwarding overhead
- ❑ Significant performance degradation despite the intra-VM communication

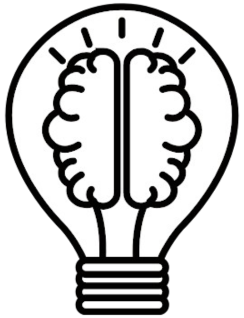


Key Takeaways

- **Drivers' Performance**:
  - OVS consistently presents the best results
- **Importance of CPU Resources**:
  - Number of CPU cores and/or usage of binding techniques can greatly influence performance
- **Resource Sharing Fairness**:
  - Some containers can achieve much higher throughput than others due to the unfair sharing of the CPU

Research Directions

- ❑ Different allocation scenarios
  - ❑ Services spread throughout multiple VMs
  - ❑ Services in different physical hosts (virtualized or not)
- ❑ Different transport protocols
  - ❑ UDP
  - ❑ QUIC
- ❑ Flow prioritization
  - ❑ Priority based mechanisms to enforce fair CPU sharing



Thank you!

Contact: Conrado.Boeira@dal.ca