Container Networking Solutions
Research Project 2

Joris Claassen

Supervised by Paola Grosso
define: "container"

and what's with the buzz?

- != Virtual machine
  - Low overhead
  - Scaling
  - Use cases
    - Google
    - Spotify
- Isolation
  - Namespaces
    - NET
    - PID
  - cgroups
docker 101
containers for everyone

- De-facto standard
- LXC wrapper
  - User "friendly"
  - Extra functionality
    - Docker Hub (Repository)
    - Dockerfiles
    - Basic networking built in
- Used in my experiments
core@node1 ~ $ docker run -ti ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from ubuntu
428b411c28f0: Pull complete
435050075b3f: Pull complete
9fd3c8c9af32: Pull complete
6d4946999d4f: Already exists
ubuntu:latest: The image you are pulling has been verified.
Important: image verification is a tech preview feature and
should not be relied on to provide security.
Digest: sha256:45e42b43f2ff4850dcf52960ee89c21cda79ec657302d36faa
aa07d88a215dd9
Status: Downloaded newer image for ubuntu:latest
root@5f10fb1c1f6c:/# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 14.04.2 LTS
Release: 14.04
Codename: trusty
research
container networking solutions

- Overlay networks
  - Literature

- Kernel modules
  - Literature
  - Performance evaluation
overlay networks

- Weave
  - Router on each node
  - ~30-70% performance hit(!)
- VXLAN
  - VLANs spanning other networks
  - 6-10% overhead on MTU 1500
Kernel modules

veth

- Connected pair of ethernet devices
- Bridged mode
  - docker0
  - Can perform NAT
- Limitations
  - Efficiency
kernel modules

macvlan

• Modes
  - Private
  - VEPA
  - Bridge

• Limitations
  - MAC access control
  - Promiscuous mode
kernel modules

ipvlan

- Modes
  - L2
  - L3
- Limitations
  - Not completely stable
  - Multi- and broadcast support
performance evaluation

How do virtual ethernet bridges perform compared to macvlan and ipvlan?

- In terms of
  - throughput
  - scalability
testing
hardware

- Nodes
  - CPU: 2x Intel(R) Xeon(R) CPU E5620 @ 2.40GHz
  - RAM: 24GB DDR3 1600Mhz
  - NIC: 10Gb SFP+
- Switch
  - Dell PowerConnect 6248
  - 10Gbit SFP+ modules
local testing

- iperf3
  - $N$ of containers exponentially increasing
  - Exponentially decreasing throughput expected

TCP - $N=1, 2, 4, 8, 16, 32, 64, 128$
UDP - $N=1, 2, 4, 8, 16$

$\text{networking_solution} = \text{veth bridge, macvlan (bridged), ipvlan (L3)}$
$\text{mtu} = 1500, 9000$
• ipvlan (L3) performs best on all number of containers pairs
  - L3; no broad- or multicast
• macvlan very close second
  - Uses of RAM as pseudo-bridge buffer
• Bridged veth pipes show their speed limitations
• performance difference with TCP
  - hypothesis: udp segment offloading in driver
• macvlan performs best
• veth performs better than ipvlan using 1 to 2 containers
switched testing

TCP - $N=1,2,4,8,16,32,64,128
UDP - $N=1,2,4,8,16

$networking_solution = veth bridge, macvlan (bridged)
$mtu = 1500, 9000
- No clear winner on 10Gbit
- Use what suits the environment
- Low bridged veth performance on MTU 1500
  - CPU bound
- Less frames sent on MTU 9000
conclusion

- Local environment:
  - `ipvlan (L3)` performs a bit better than `macvlan (bridge mode)`
    - But is still unstable when networked
  - `macvlan (bridge mode)` performs ~2.5 times as good as bridged veth
    - And stable as well
    - Namespace has its own MAC address

- Switched environment:
  - veth bridges and `macvlan (bridge mode)` on par in TCP
  - `macvlan` uses less CPU than `veth bridges`
    - Performs better using UDP
future work

- Further look into kernel modules
  - Re-evaluate ipvlan performance
  - Further UDP testing

- Test performance of new overlay networks
  - Socketplane - now Docker
    - Based on Open vSwitch and VXLAN
  - Weave with Open vSwitch backend
    - Still in development
Thank you and enjoy your lunch!