

Docker Usage

Baohua Yang

2015-11-05

Outline

- Data
- Networking
- Security
- Configuration
- Monitoring
- And?

Data

- `/var/lib/docker`
 - *aufs*. UFS storage
 - *containers*. Information of each container
 - *execdriver/native*. Running container information
 - *graph*. Images information
 - *init*. docker init binary versions
 - *linkgraph.db*. SQLite db file keeping the links between containers
 - *repositories-aufs*. Info for images
 - *trust*. signatures
 - *volumes*. Randomly created volumes on host

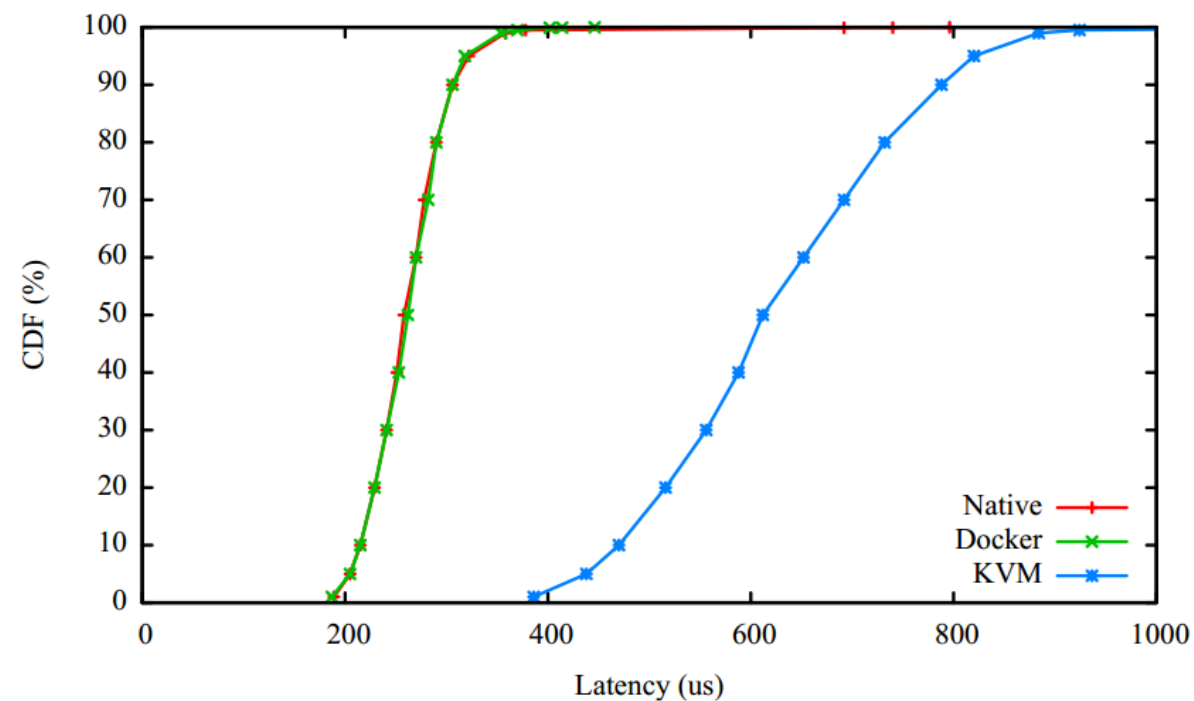
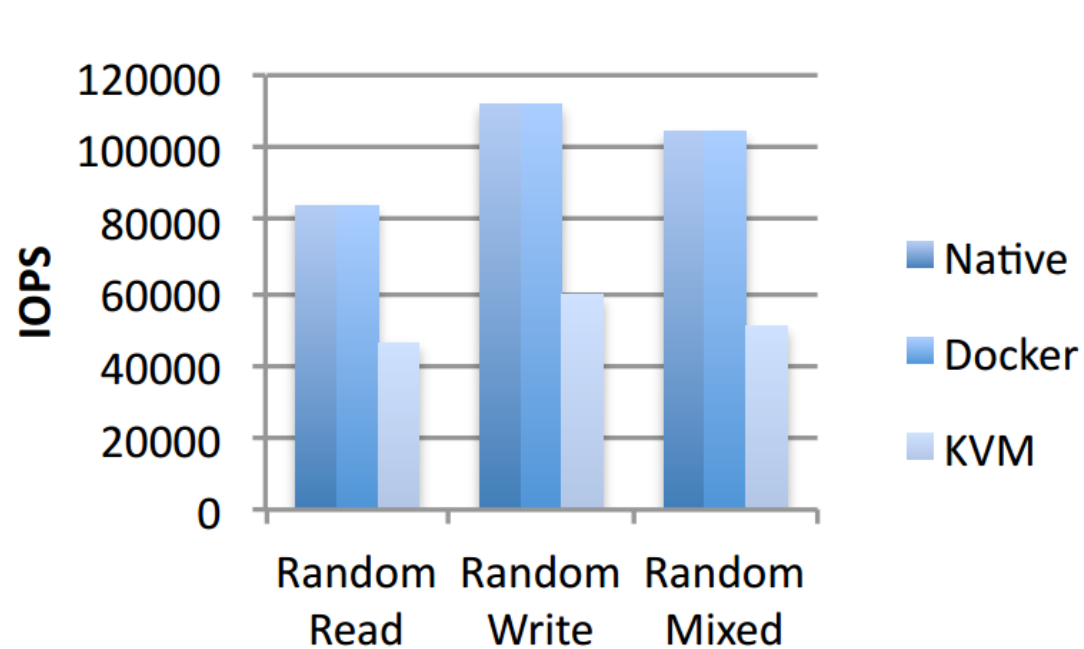
Data/Backends

- No Reliability guarantee!
- -s to select your favorite
- AUFS
 - Not upstreamed
- Device Mapper
 - Thin provisioning
 - Loopback mounted sparse file
- Btrfs
 - Docker upstream
 - No selinux
 - No page cache sharing
- OverlayFS
 - Supported by Linux upstream
 - Potential one

```
// Slice of drivers that should be used in an order
priority = []string{
    "aufs",
    "btrfs",
    "devicemapper",
    "overlay",
    "vfs",
}
```

daemon/graphdriver/driver.go

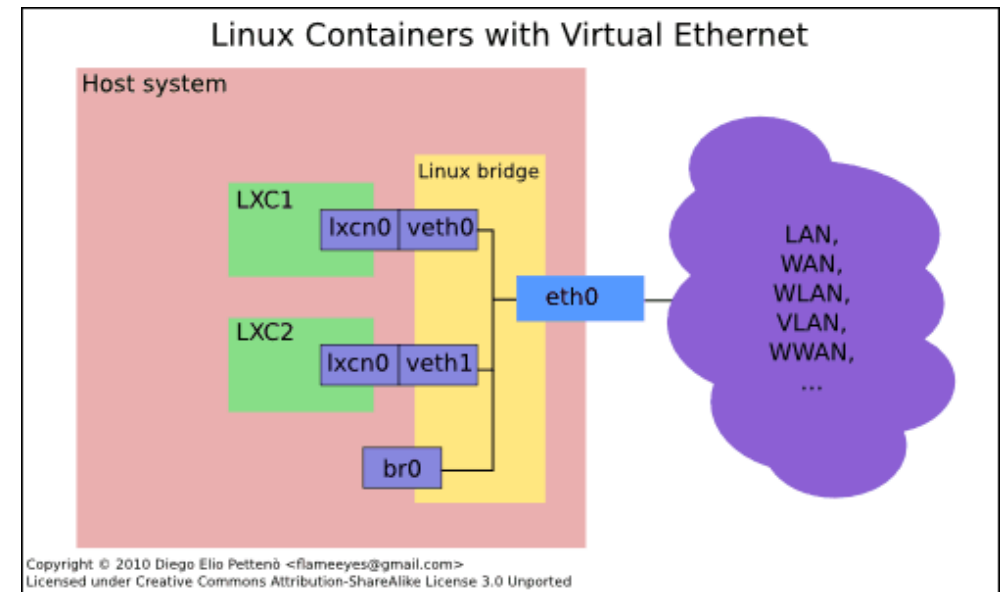
Data/Performance



An Updated Performance Comparison of Virtual Machines and Linux Containers, IBM Research, 2014

Networking

- Think container as virtual machine! But
 - Quicker booting/exiting
 - More instances
 - Shorter life (depends...)
- Based on Linux Networking
 - veth
 - macvlan
 - namespace
 - iptables



Networking/DHCP

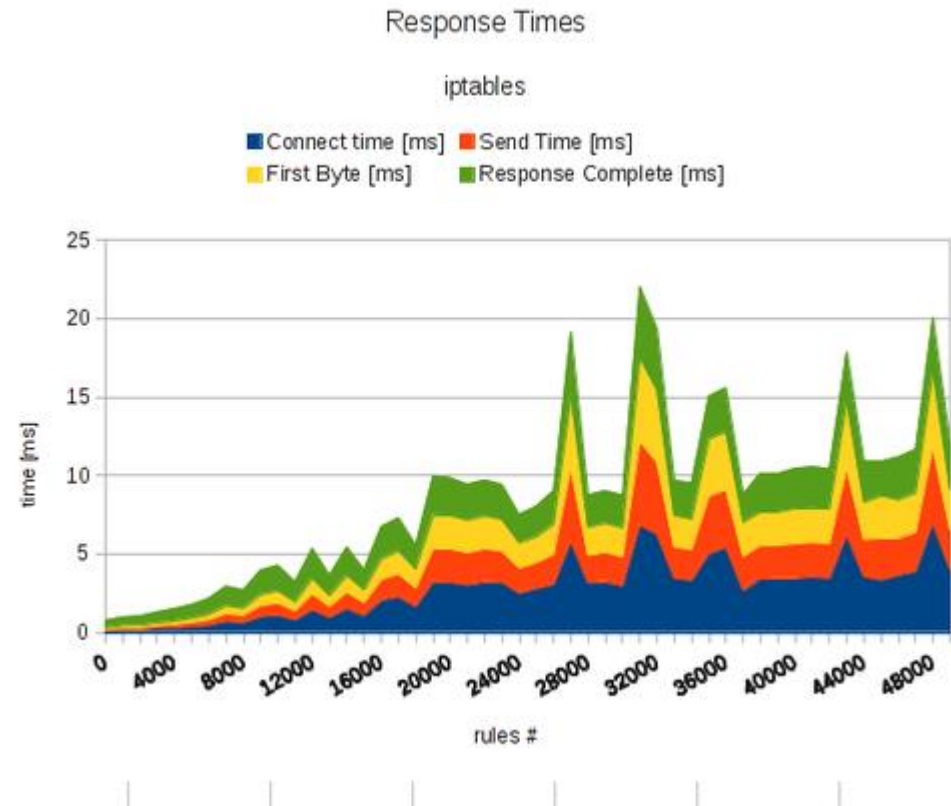
- DHCP problems
 - Docker internally assigns IP for each container and change on restart even in 1.6 (#issue 2801).
 - Check bridge, routing table and /etc/resolv.conf to obtain local network information
 - DHCP service: fast!

```
addr = []string{
    // Here we don't follow the convention of using the 1st IP of the range for the gateway.
    // This is to use the same gateway IPs as the /24 ranges, which predate the /16 ranges.
    // In theory this shouldn't matter - in practice there's bound to be a few scripts relying
    // on the internal addressing or other things like that.
    // They shouldn't, but hey, let's not break them unless we really have to.
    "172.17.42.1/16", // Don't use 172.16.0.0/16, it conflicts with EC2 DNS 172.16.0.23
    "10.0.42.1/16",  // Don't even try using the entire /8, that's too intrusive
    "10.1.42.1/16",
    "10.42.42.1/16",
    "172.16.42.1/24",
    "172.16.43.1/24",
    "172.16.44.1/24",
    "10.0.42.1/24",
    "10.0.43.1/24",
    "192.168.42.1/24",
    "192.168.43.1/24",
    "192.168.44.1/24",
}
```

daemon/networkdriver/bridge/driver.go

Networking/iptables

- Docker uses static iptables NAT rules to do port mapping
 - Existing rules confliction
 - Performance problems
 - Slow in update/config
 - Complex in grammar/ more rule
 - Dynamic rule generation?
- nftables since 3.13
 - Faster in update/config
 - Less kernel work



Networking/ideal solution?

- Kubernetes
 - Flannel
- Weave
- Socket
- OpenStack Neutron
- pipework
- tenus
- docknet
- ...

SDN!

Security

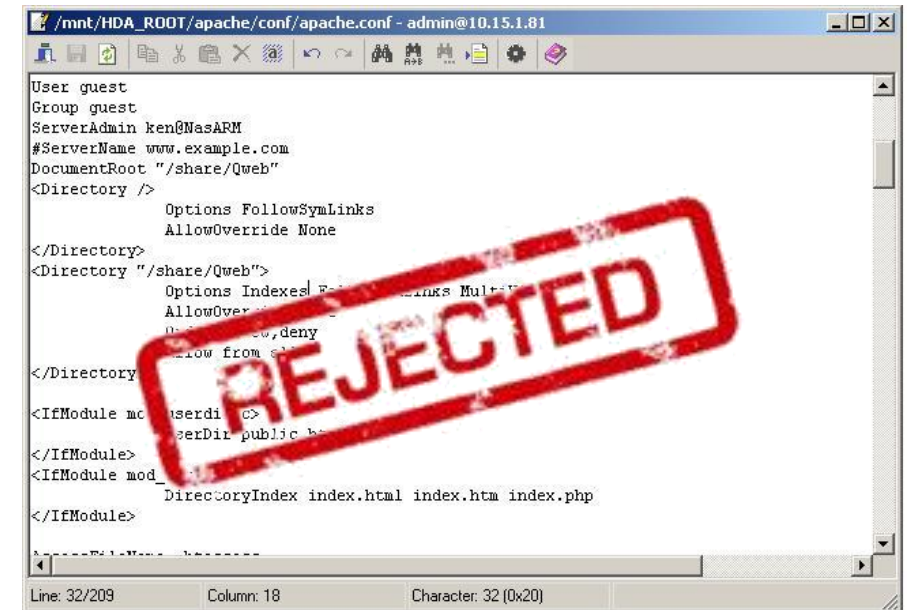
- Think container as application!
- Validate external images
- Only put mutually-trusted containers on the same host
 - `--icc=false` + manual link
- Use AppArmor/SELinux
- Minimize privileges enforced inside the container
 - `docker run -it --rm -u user1 --cap-drop SETUID --cap-drop SETGID ...`
- Limit resource usage using cgroups
 - `docker run -it --rm --cpuset=0,1 -c 2 -m 128m ...`
 - `docker -d --storage-opt dm.basesize=5G`
- Mount volume with proper permission
- Check the [Docker Secure Deployment Guidelines](#)
- Some one chooses running containers inside VM

Configuration

- Config file is the legacy way
 - Formats vary by xml, json, manual-defined, etc.
 - Where to store for application in container?
 - How to update config values flexibly?
 - Management is too hard

- Decouple from application itself

- Configuration = Key+Value
 - Store in centric db
 - ENV variables/running options?



```
/mnt/HDA_ROOT/apache/conf/apache.conf - admin@10.15.1.81
User guest
Group guest
ServerAdmin ken@NasARM
#ServerName www.example.com
DocumentRoot "/share/Qweb"
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory "/share/Qweb">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order deny,allow
    Allow from all
</Directory>
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.php
</IfModule>
-----File Name-----
Line: 32/209      Column: 18      Character: 32 (0x20)
```

Monitoring

- Easier to monitor container than virtual machine
 - CPU
 - Memory
 - IO
 - Network
 - FD
- Root Tracing
 - docker logs
 - Tools like ELK
- Inject limit information into container proactively!

Miscellaneous

- Supervisor
- Discovery
- Boot order
- Fat container
- Zombie-reaping, syslog-ng, ssh, cron, runit, setuser problems
 - phusion/baseimage

No Use Docker Manually

- Docker Inc
 - Compose
 - Machine
 - Swarm
- IBM
 - Bluemix
- Google
 - Kubernetes
 - Borg/Omega
- Mesos
- OpenStack

4 PRINCIPLES

4 Basic Principles

- 0. No silver bullet!
- 1. DONOT use container before understanding enough
- 2. Try using container transient, stateless, and fault-tolerant
- 3. Do you care IO or security heavily?

Q&A

