



KVM
FORUM

Comparing Performance of NVMe Hard Drives in KVM, Baremetal, and Docker Using Fio and SPDK for Virtual Testbed Applications

Mauricio Tavares



KVM
FORUM

Comparing Performance of NVMe Hard Drives in KVM, Baremetal, and Docker Using Fio and SPDK for Virtual Testbed Applications

(Testing the lazy way)

raubvogel@gmail.com

Outline

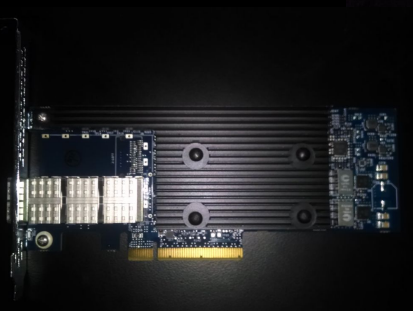
- Why
- Build test environment
- Results
- Conclusion?
- Questions

Warning

- This presentation does not reflect the views and opinions of my employee
- Less experiment results, more how to get them without going insane
- Pretty pictures

Virtualization and physical devices

- Why?
 - Research needs moah speed!

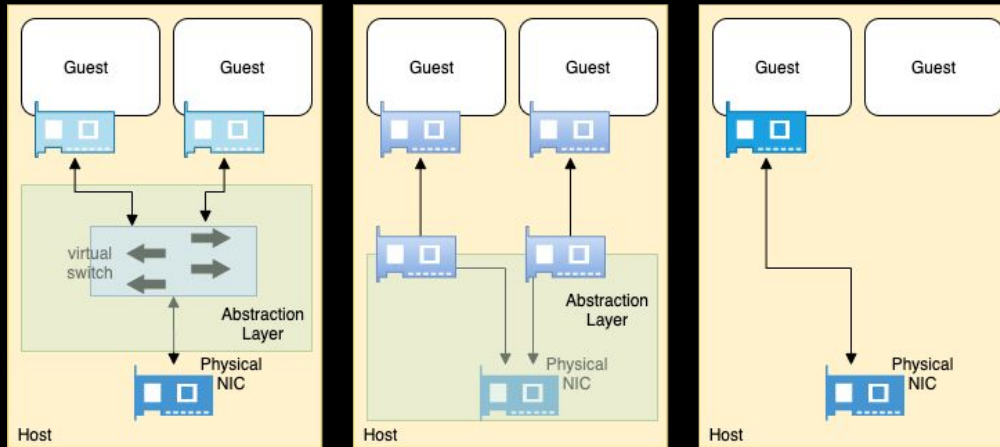


- But have limited space/energy resources



Does it really work?

- Good question!
- Abstraction costs. Absolute abstraction costs absolutely.



You are not telling me everything...

Right you are

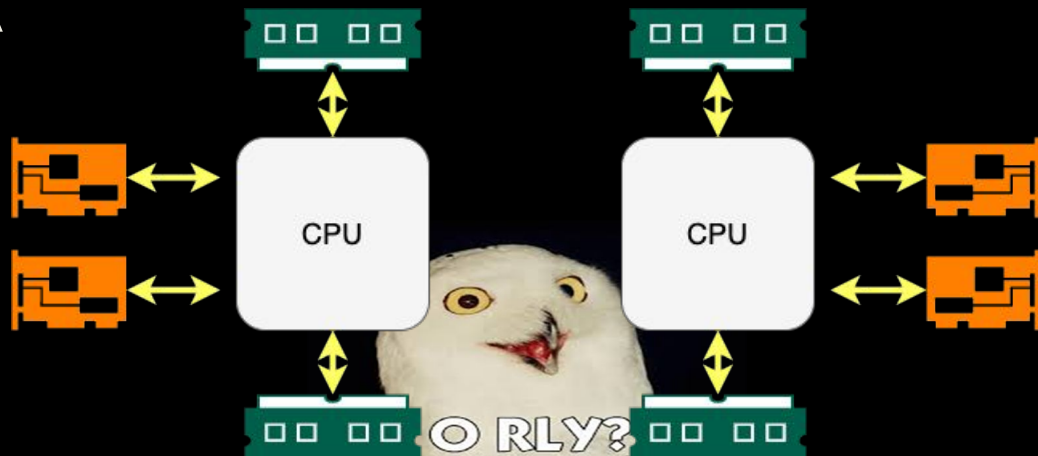
- PCI
 - versions, bandwidth and lanes
- NUMA



You are not telling me everything...

Right you are

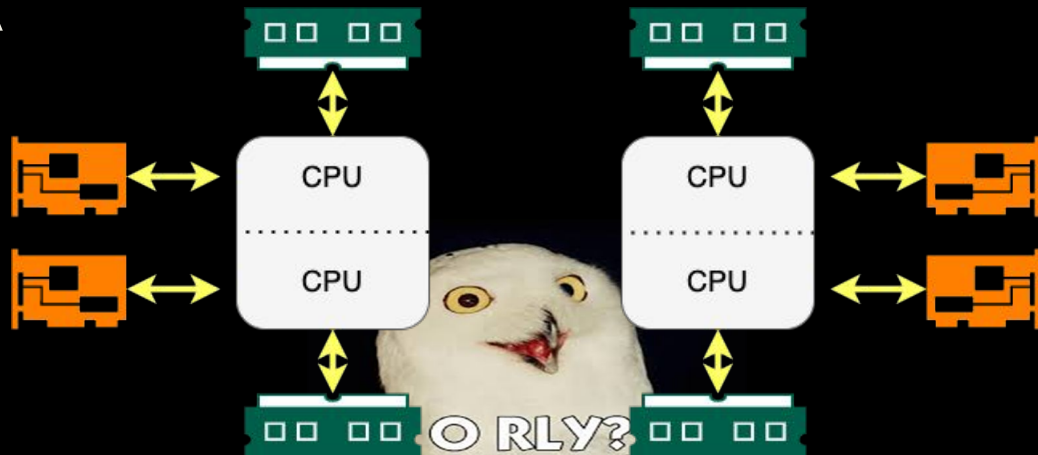
- PCI
 - versions, bandwidth and lanes
- NUMA



You are not telling me everything...

Right you are

- PCI
 - versions, bandwidth and lanes
- NUMA



Testing

- Standard test procedures
- Hard drives (NVMe)
 - SNIA SSS performance tests v2.0.1
 - How to prepare/condition the drive
 - How to test for BW, IOPS, latency

Toolkit

- FIO
- SPDK
- Custom code
 - findrive
 - torturedisk

Experiment

- Hardware
 - Supermicro 2029U-TN24R4T
 - 48 PCIe v3 lanes :(
 - 2 NUMA Zones :~(
 - 10x Intel Intel DC P4500 4TB NVMe
 - Attached to one controller :(
 - Other cards
- CentOS 8



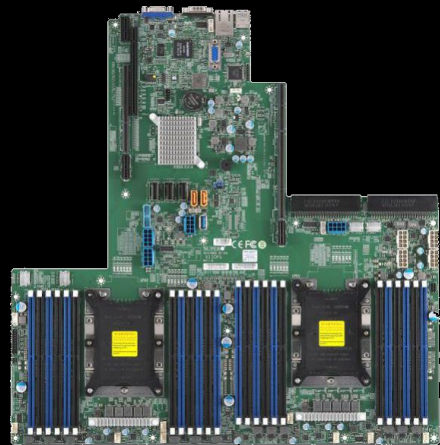
Experiment

```
[root@testbox ~]# virsh nodeinfo
CPU model:          x86_64
CPU(s):             36
CPU frequency:     2582 MHz
CPU socket(s):     1
Core(s) per socket: 18
Thread(s) per core: 1
NUMA cell(s):     2
Memory size:       791312404 KiB

[root@testbox ~]#
```

Experiment

```
[root@testcentos ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):               36
On-line CPU(s) list: 0-35
Thread(s) per core:  1
Core(s) per socket:  18
Socket(s):           2
NUMA node(s):       2
Vendor ID:            GenuineIntel
CPU family:           6
Model:                85
Model name:           Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz
[...]
NUMA node0 CPU(s):   0-17
NUMA node1 CPU(s):   18-35
```



Experiment

- FIO vs SPDK vs whatever
- 10 Drives
- 8 blocksizes
- 7 different read/write ratios
- Baremetal vs KVM vs Docker
 - Tweaking

Automation

- Hands-off
- Minimize downtime between tests
- Reproducibility
- Documentation

Automation

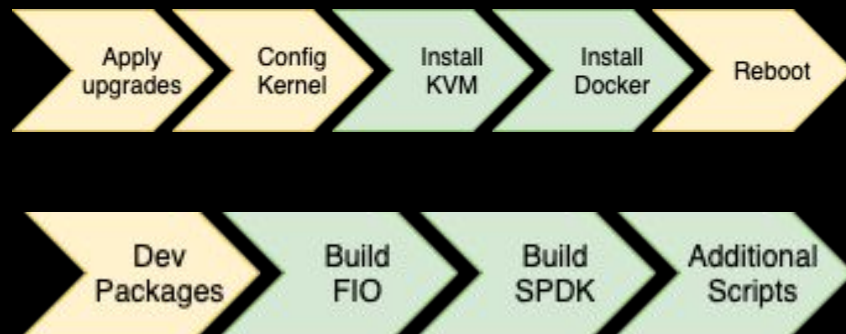
- Reproducibility

```
[root@testcentos ~]# cat
dev/test/results/nvme2n1-fiotest_2020-09-16-1207/torture.write.1024K.1
[global]
bs=1024K
direct=1
rw=write
ioengine=libaio
iodepth=8
runtime=24h
steadystate_duration=1800
steadystate=iops_slope:0.3%

[test]
filename=/dev/nvme2n1
[root@testcentos ~]#
```

Automation

- Ansible



- Basic Docker/KVM images

Automation

- Running the experiment

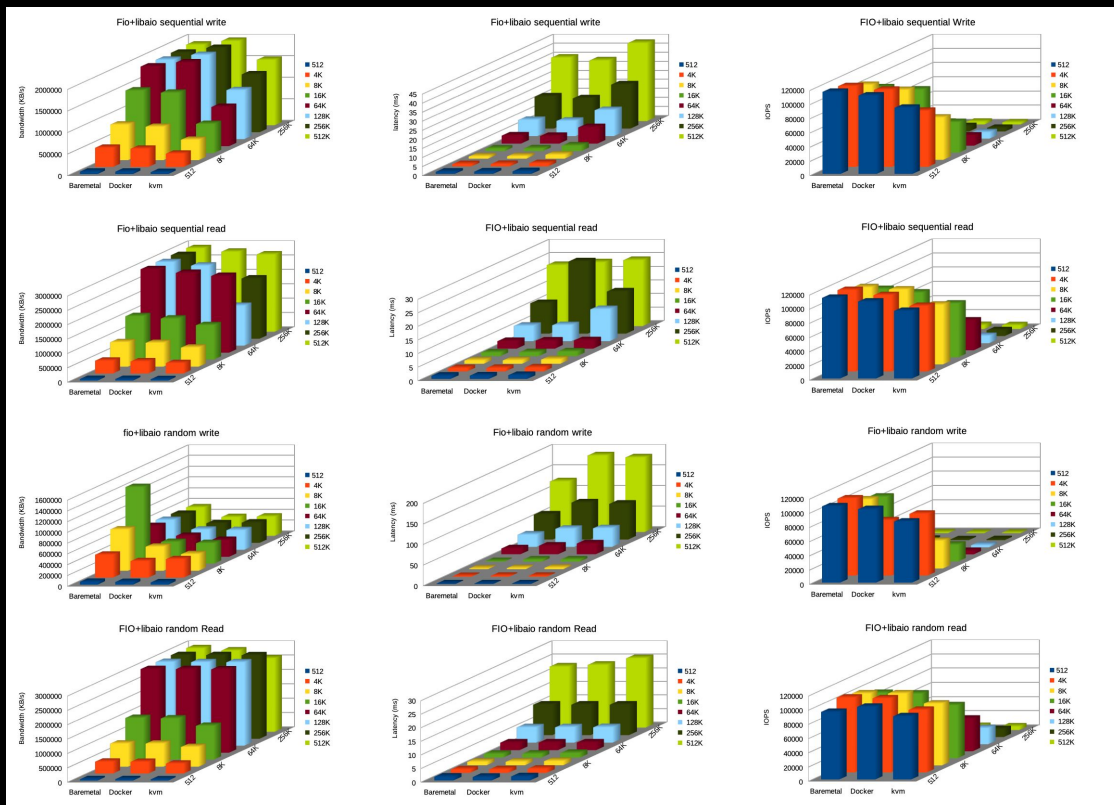
```
torturedisk --blocksize "1024K 128K 64K 32K 16K 8K 4K  
512" --mixreads "100 95 65 50 35 5 0" --device  
"/dev/nvme2n1 /dev/nvme5n1 /dev/nvme9n1" --ioengine  
libaio --outdir libaio --time 24h --steadystate
```

Automation

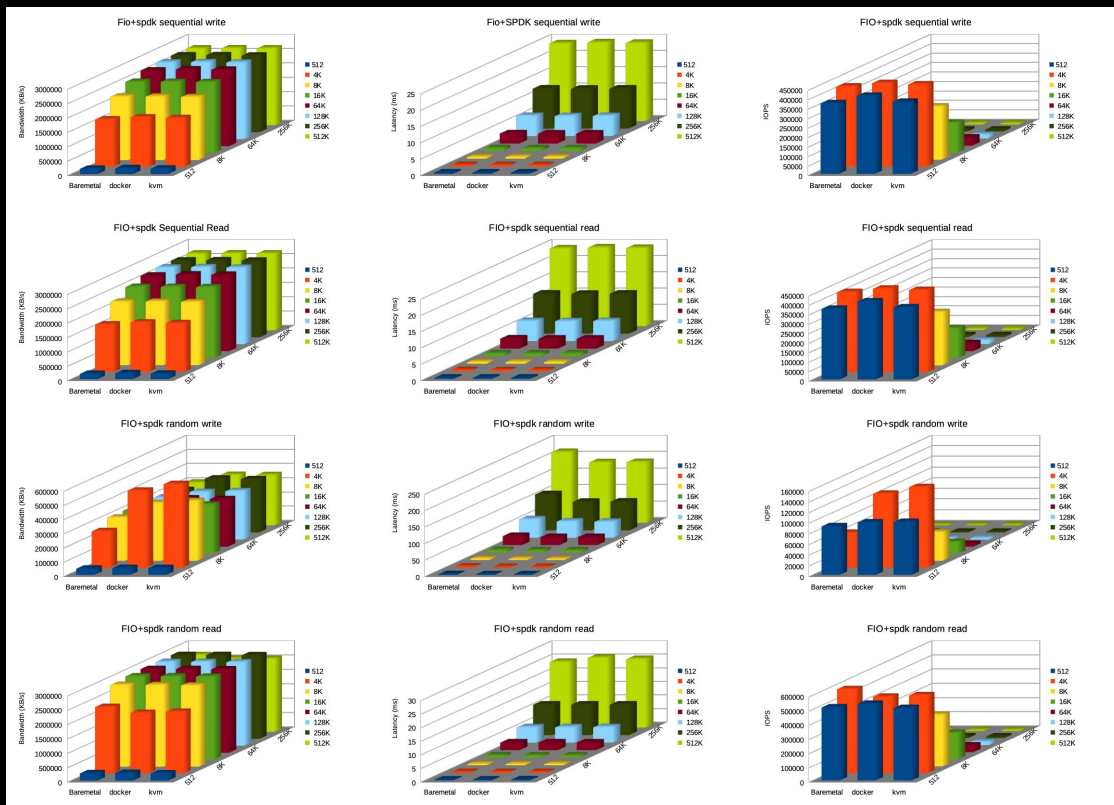
- Running the experiment

```
for DEVFILE in $(python3 ~/dev/test/findrive/findrive.py |cut
-d \' -f 8| tr ':' '_' |tr '.' '_'); do echo
"OUTDIR=$DEVFILE_$(date +%F); mkdir $OUTDIR; docker run -d
--rm --privileged --env DEVFILE=$DEVFILE -v
/dev/$DEVFILE:/dev/$DEVFILE -v $PWD/$OUTDIR:/export/results -t
torturedocker"; done
```

Results



Results



Conclusion

- Automate your experiment for reproducibility, sanity, and laziness
 - Ansible
 - pre-built (docker/kvm) images
 - Have a script to do the actual testing for you



Useful Stuff

- <https://www.snia.org/>
- <https://github.com/axboe/fio>
- <https://github.com/spdk/spdk> <https://spdk.io/>
- <https://software.intel.com/en-us/articles/evaluate-performance-for-storage-performance-development-kit-spdk-based-nvme-ssd>
- <https://www.whatisfabric.net/>
- <https://github.com/raubvogel/findrive>

Questions





KVMM FORUM