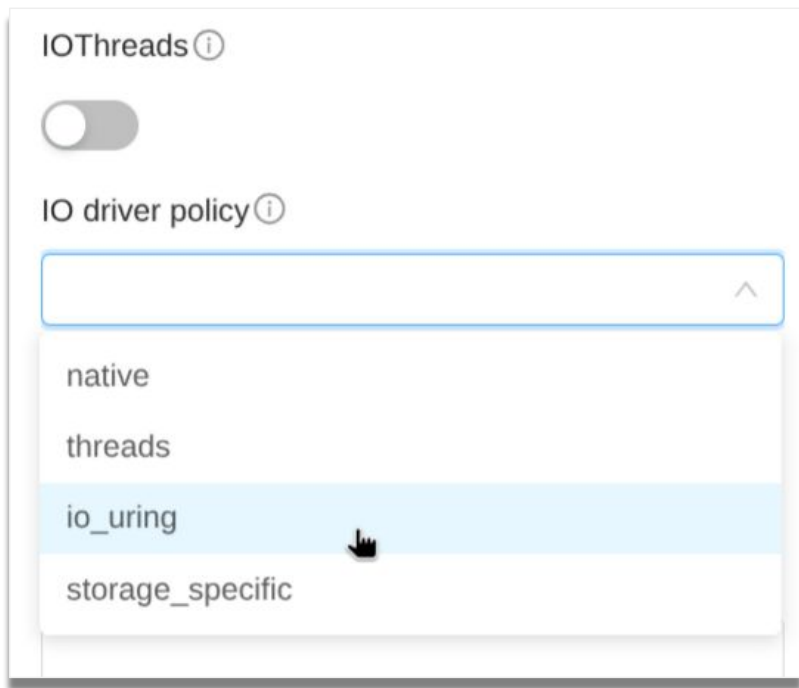


KVM Storage Performance: A Comparative Analysis of I/O Modes

Venko Moyankov, StorPool

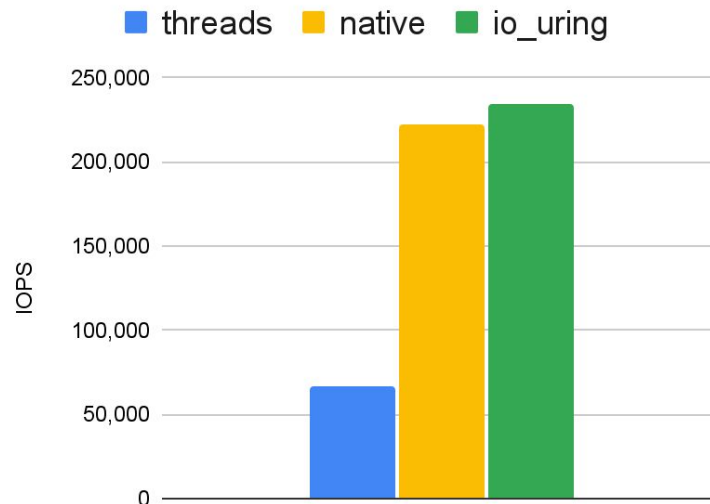
Which option should I choose?

A complex answer to a simple question.



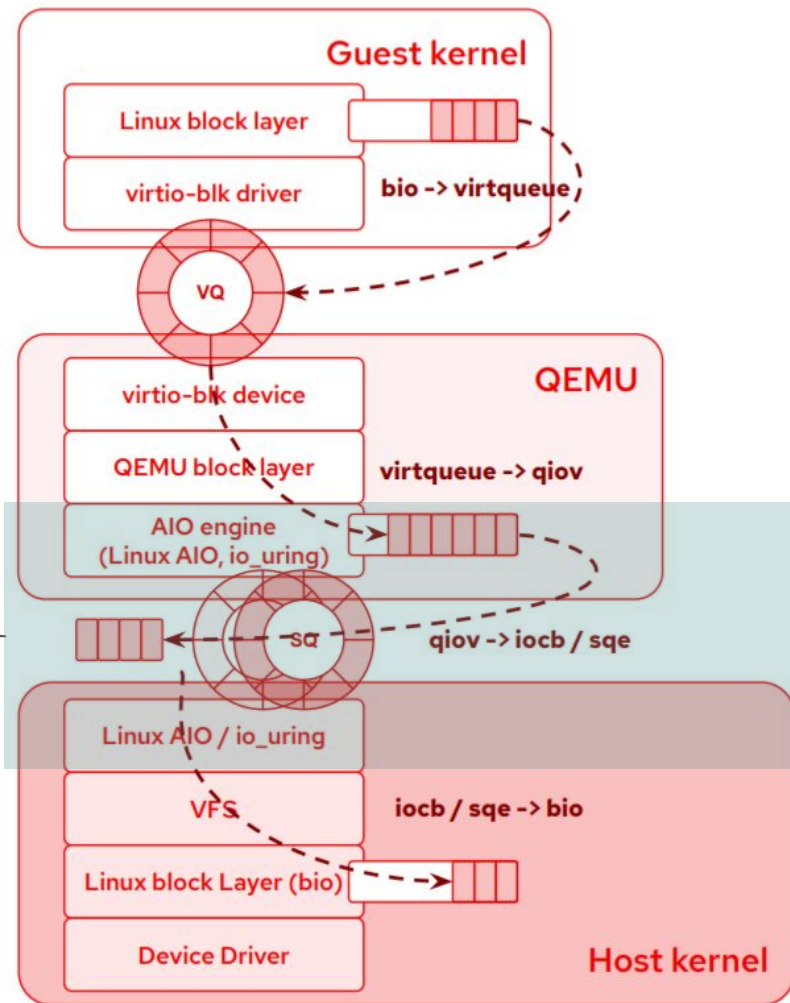
Max IOPS

Random write, BS=4KiB, QD=64



The QEMU Storage path

We'll deal with this part
(mostly)



Source: KVMForum_2021_vdpa_blk_presentation
https://static.sched.com/hosted_files/kvmforum2021/b1/KVMForum_2021_vdpa_blk_Stefano_Garzarella.pdf

Compare

- Async IO backends (threads / native / io_uring)
- IOthread vs main event loop
- Virtual device type: virtio-blk / virtio-scsi
- Number of IO threads
- Huge pages vs 4k pages

Measure:

- Read and write latency
- maximum IOPS
- How latency depends on load
- Application performance - pgbench, rsync
- CPU usage

This is not about

- Hardware - e.g. Intel vs AMD, Samsung vs Micron, SATA vs NVMe, etc.
- Storage types - Local vs NFS
- Image formats - QCOW2 vs raw
- Storage technologies - iSCSI vs NFS vs FC
- Storage Vendors
- Hypervisors - KVM vs Xen vs ESXi
- Kernel and Qemu versions
- Other technologies like vhost-scsi, vhost-user, vdpa-blk, VDUSE, PCI passthrough, io_uring passthrough

Test Setup - Host

- CPU: AMD EPYC 7203P 8-core, 3.4GHz
- RAM: 128GB
- Storage:
 - NVMe Micron 7450 PRO 7.68TB (all tests except bandwidth)
 - 7 x NVMe Samsung PM9A3 7.68TB (for bandwidth tests)
- OS: Debian 13 (trixie), kernel 6.12.48+deb13-amd64
- kernel options: `iommu=pt`
- QEMU emulator version 10.0.3 (Debian 1:10.0.3+ds-0+deb13u1)
- libvirt 11.3.0
- LVM (for BW tests only): 7 pvs striped:
`lvcreate -i 7 --stripesize 1024k vg0 --name lv1`

Test Setup - Guest

- UEFI, pc-q35-10.0
- 8 vCPUs host-passthrough (pinned to physical CPUs)
- 8 GiB RAM
- OS: Debian 13 (trixie), kernel 6.12.48+deb13-amd64
- fio 3.39
- pgbench

Test setup - VM definition

```
<memory unit='KiB'>8388608</memory>
<vcpu placement='static'>8</vcpu>
<iothreads>1</iothreads>
<cputune>
  <vcpupin vcpu='0' cpuset='0' />
  <vcpupin vcpu='1' cpuset='1' />
  <vcpupin vcpu='2' cpuset='2' />
  <vcpupin vcpu='3' cpuset='3' />
  <vcpupin vcpu='4' cpuset='4' />
  <vcpupin vcpu='5' cpuset='5' />
  <vcpupin vcpu='6' cpuset='6' />
  <vcpupin vcpu='7' cpuset='7' />
  <emulatorpin cpuset='15' />
  <iothreadpin iothread='1' cpuset='14' />
</cputune>
```

```
<disk type='block' device='disk'>
  <driver name='qemu' type='raw'
    cache='none' io='native'
    discard='unmap' iothread='1' />
  <source dev='/dev/nvme8n1' />
  <target dev='vdb' bus='virtio' />
</disk>
```

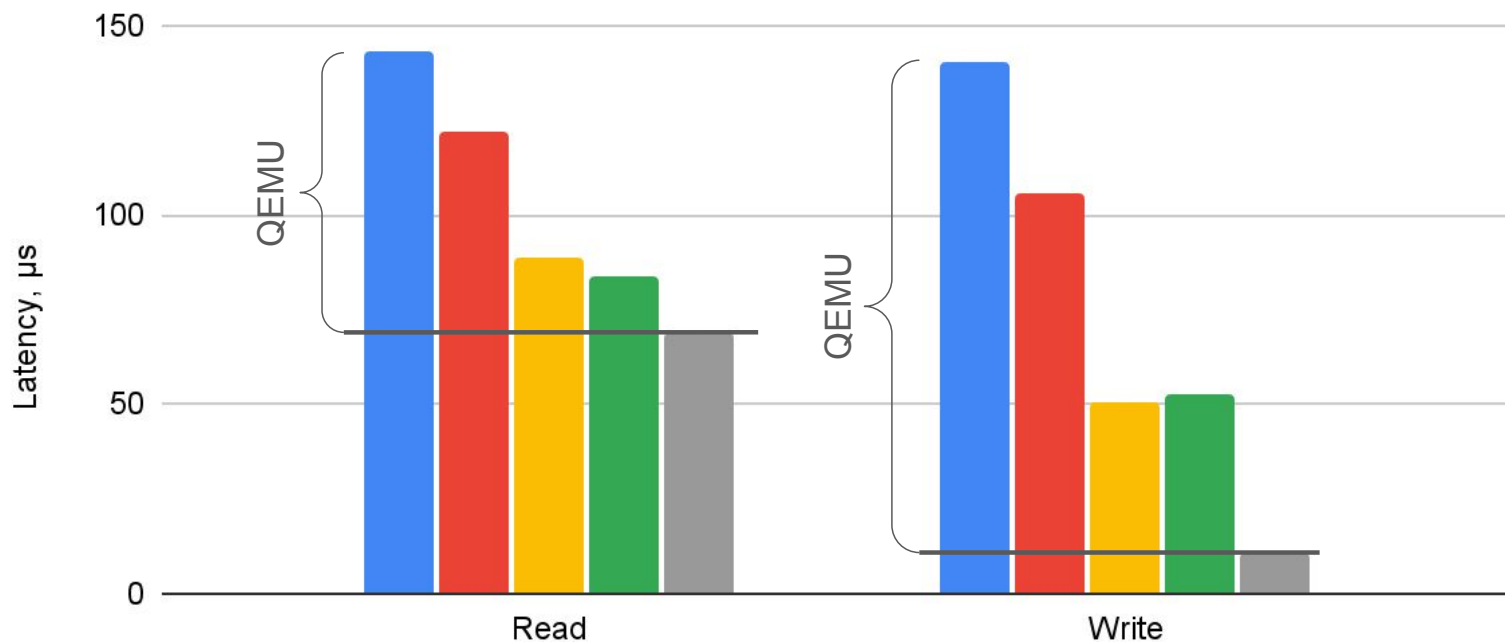
Results

Qemu Storage Backend Drivers

- threads
- Linux Native AIO
 - in the main QEMU emulator thread
 - in a dedicated IO thread
- io_uring (in IO thread)

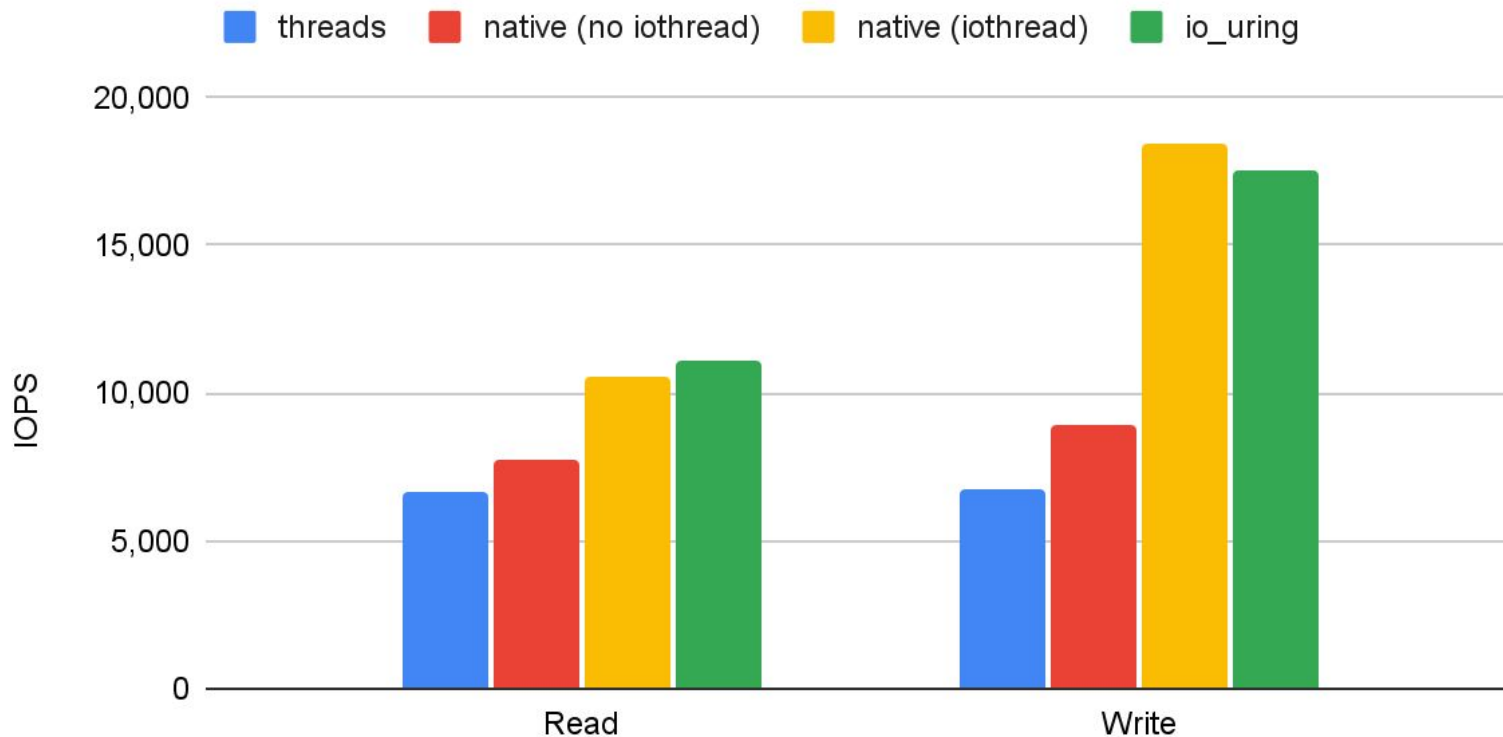
Latency, BS=4k QD=1 (lower is better)

■ threads ■ native(no iotreads) ■ native (iothread) ■ io_uring
■ host

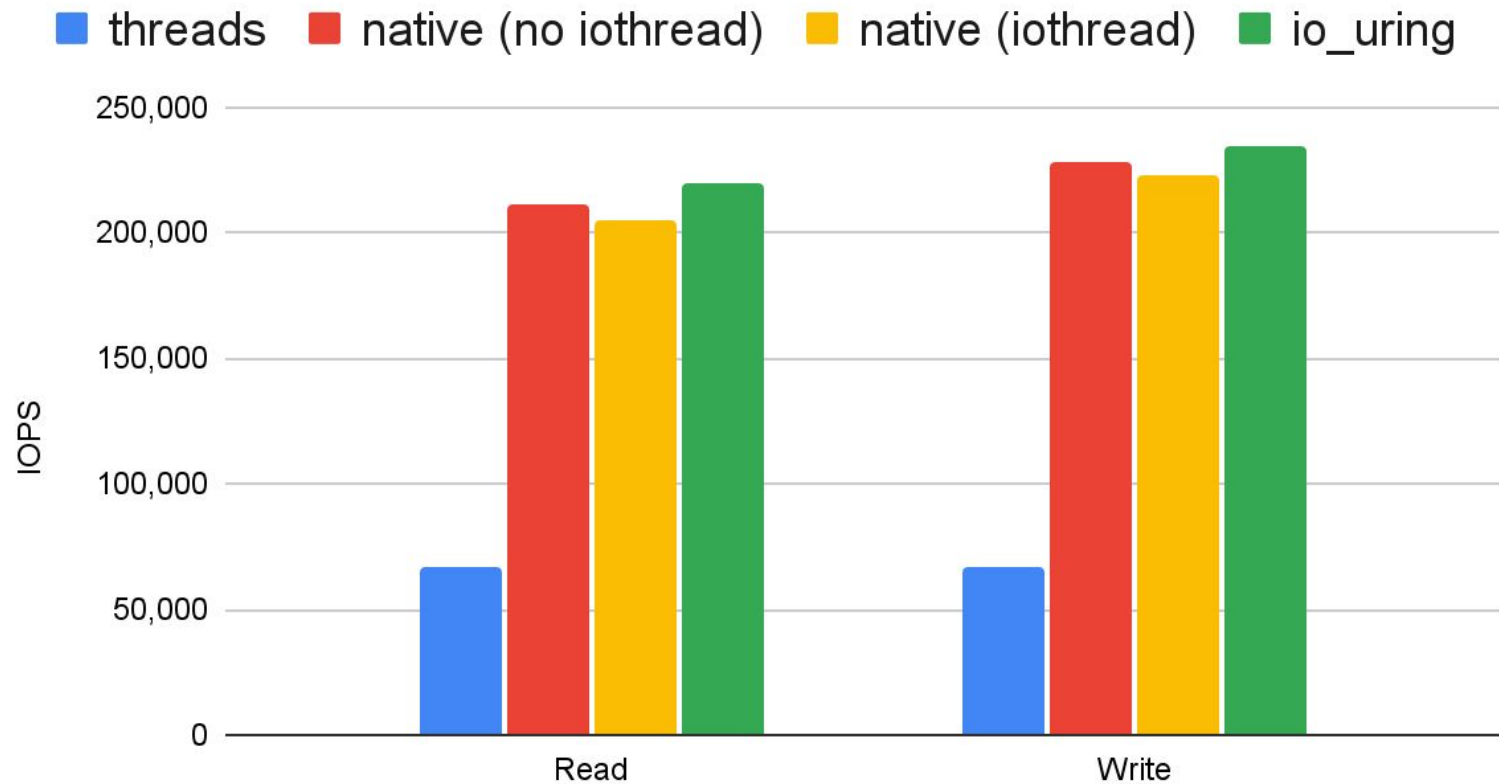


IOPS, QD=1

Random Read, Random Write, BS=4kB, QD=1

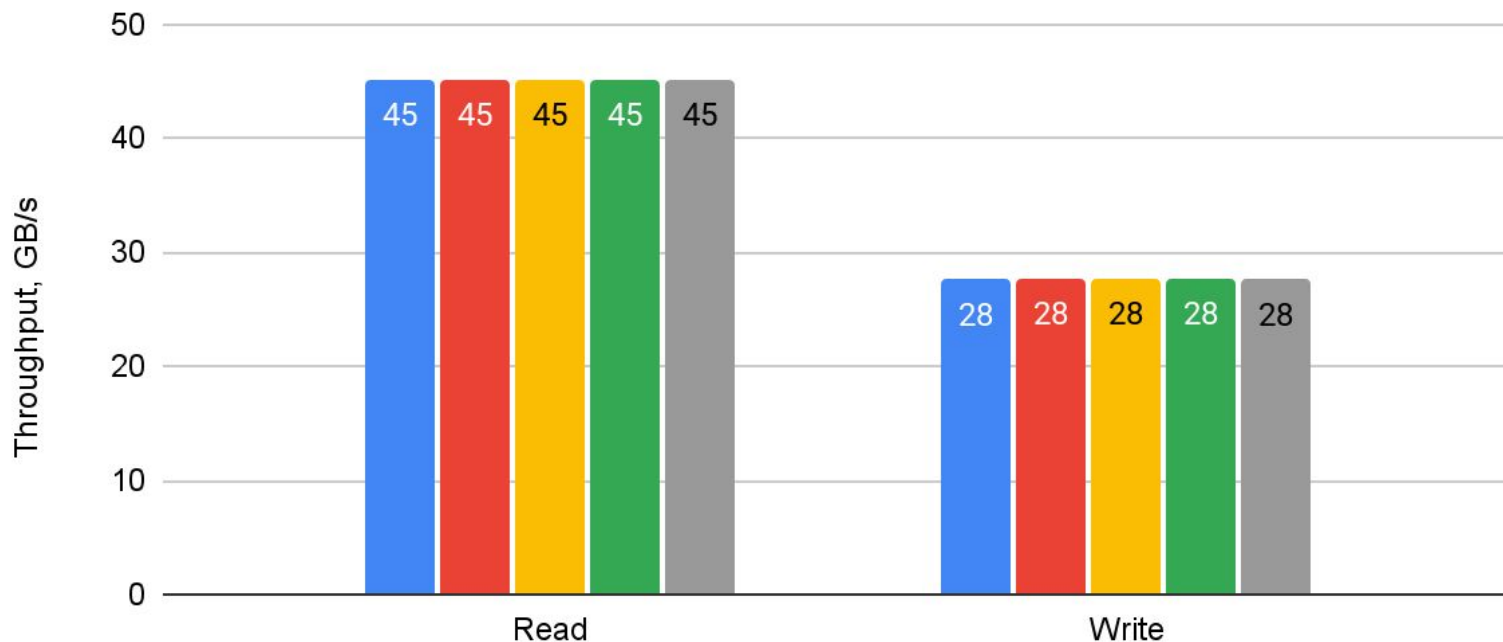


Max IOPS, BS=4k QD=64

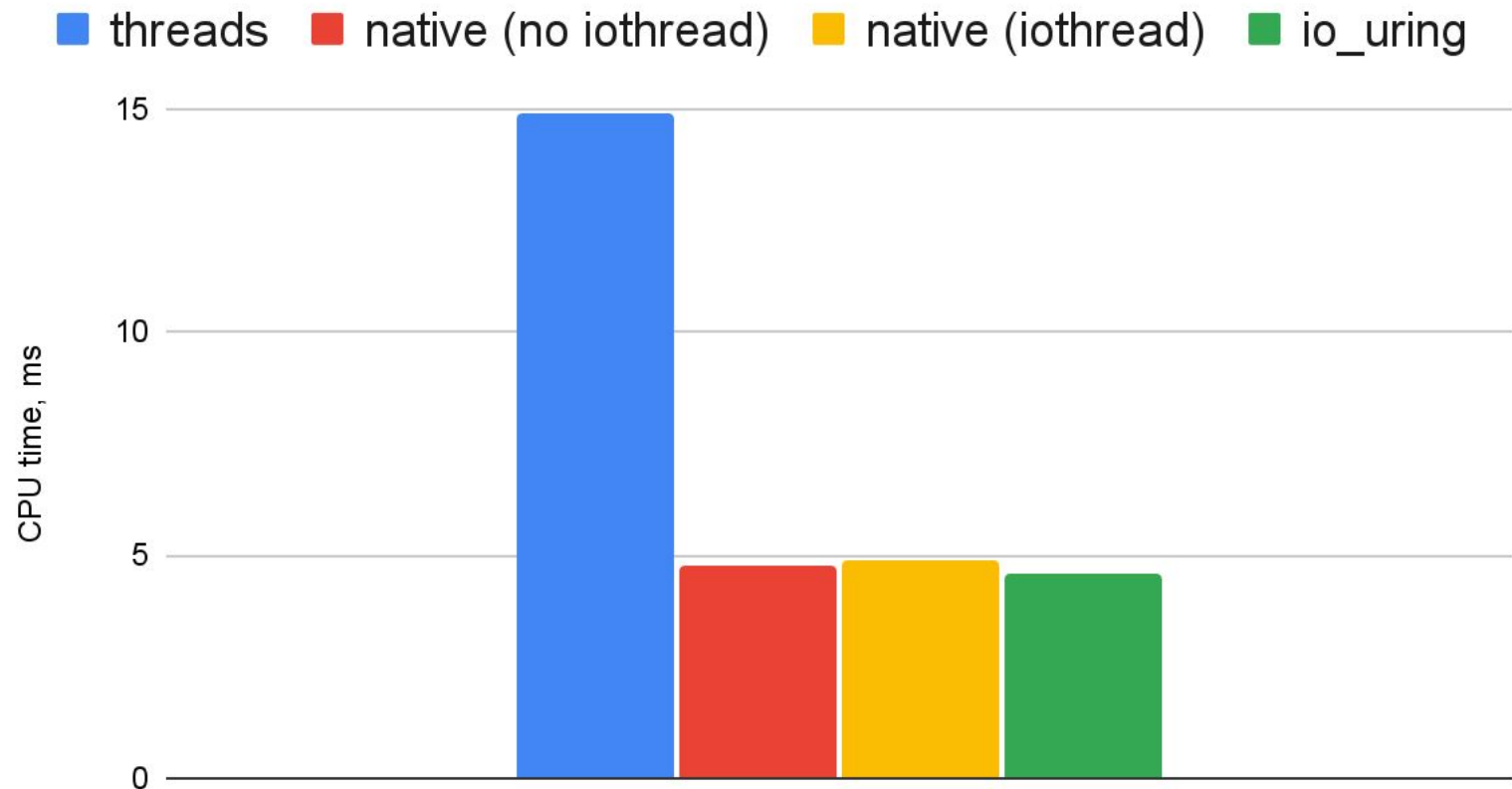


Throughput, BS=16M, QD=16

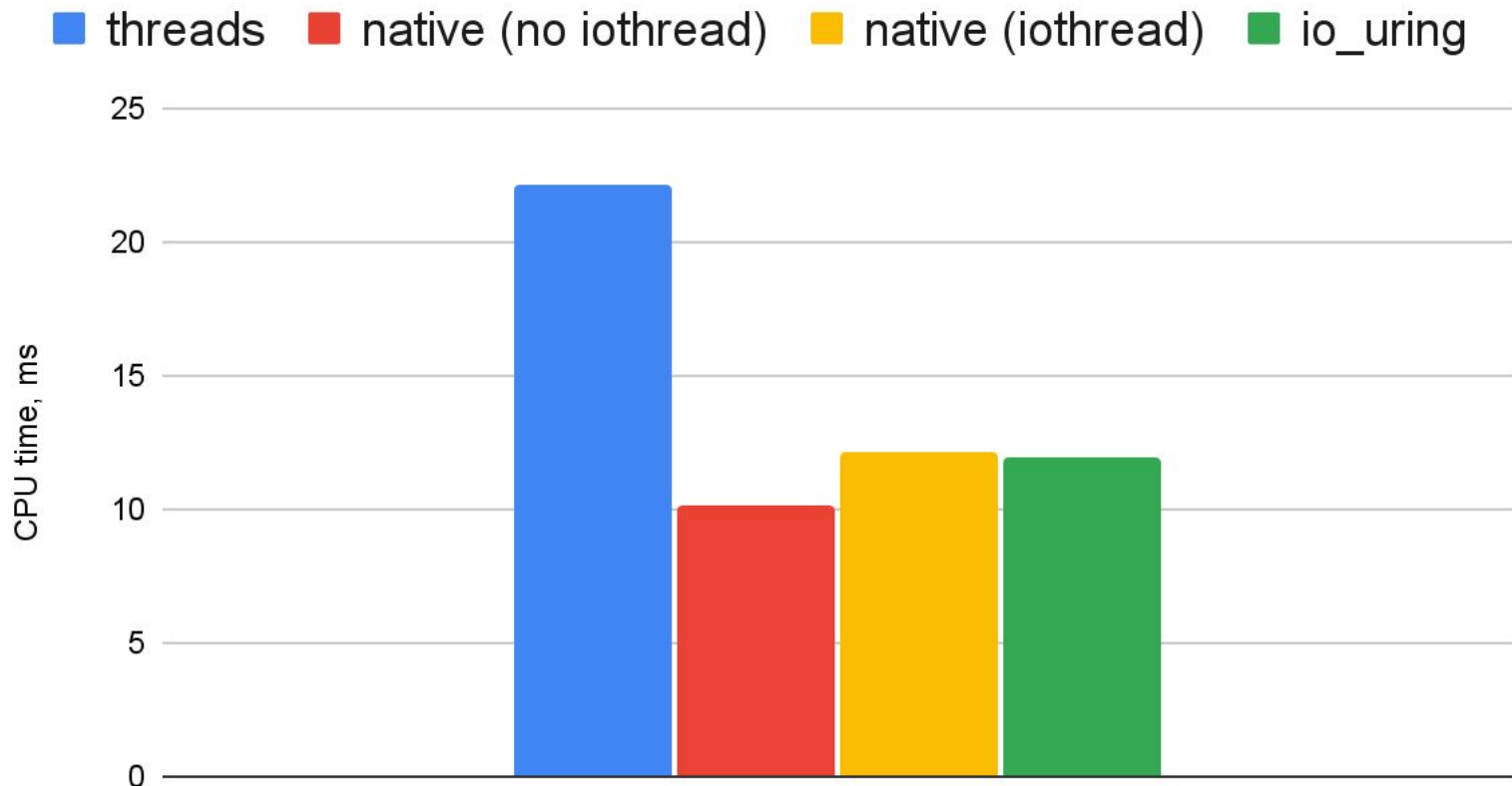
■ threads ■ native (no iothread) ■ native (iothread) ■ io_uring
■ host



CPU Usage per 1000 IOPS

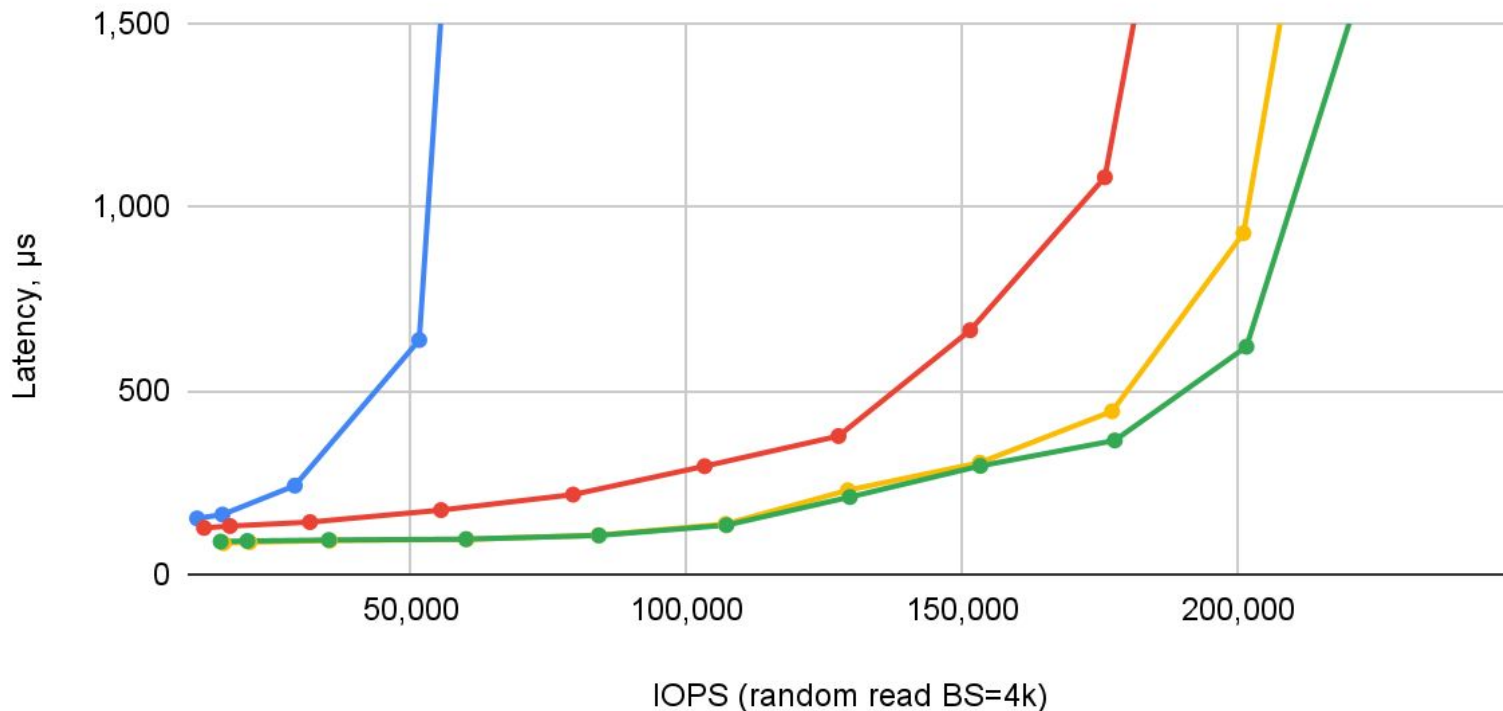


CPU Usage per 1GB/s

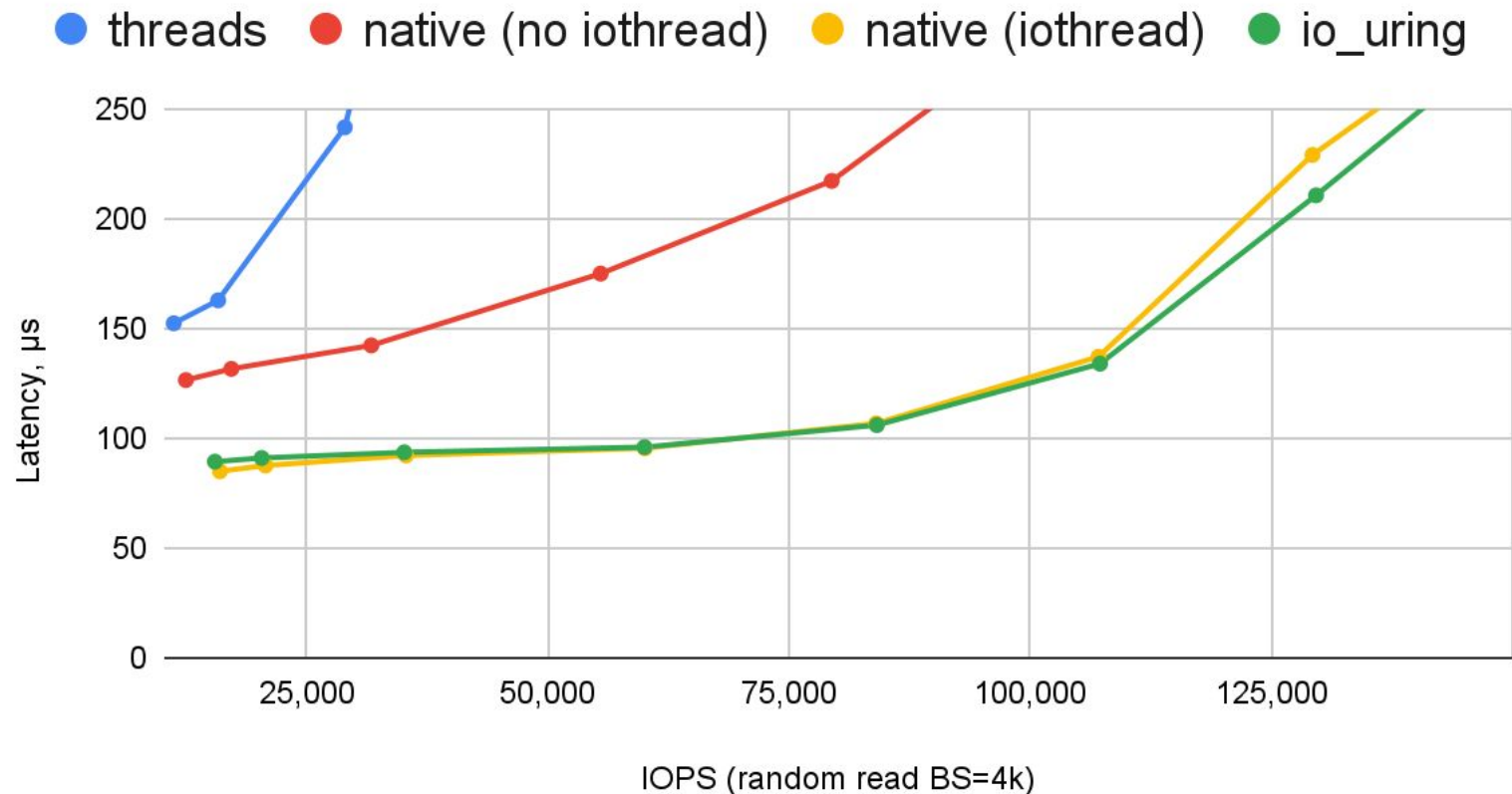


Latency vs. IOPS

● threads ● native (no iothread) ● native (iothread) ● io_uring

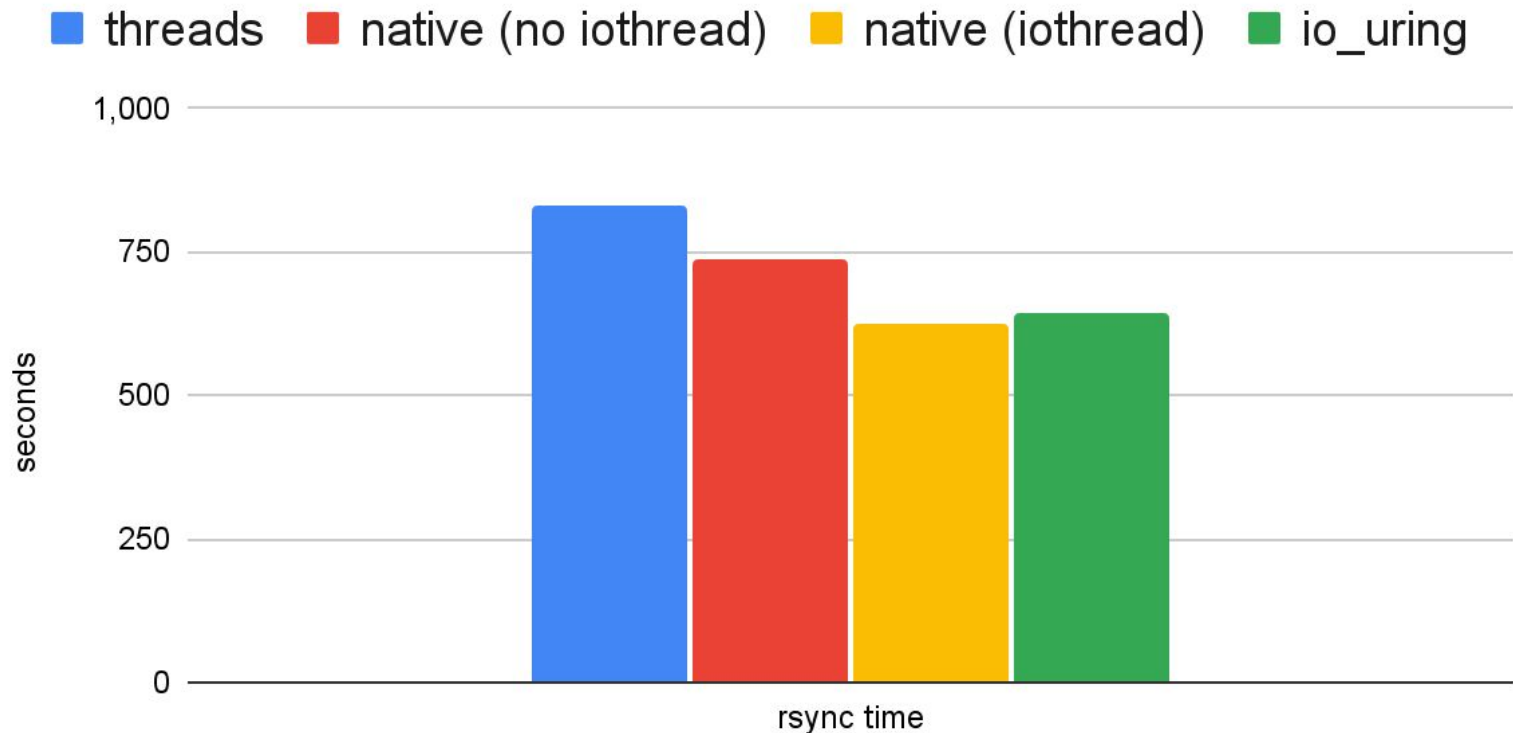


Latency vs. IOPS (zoomed in)



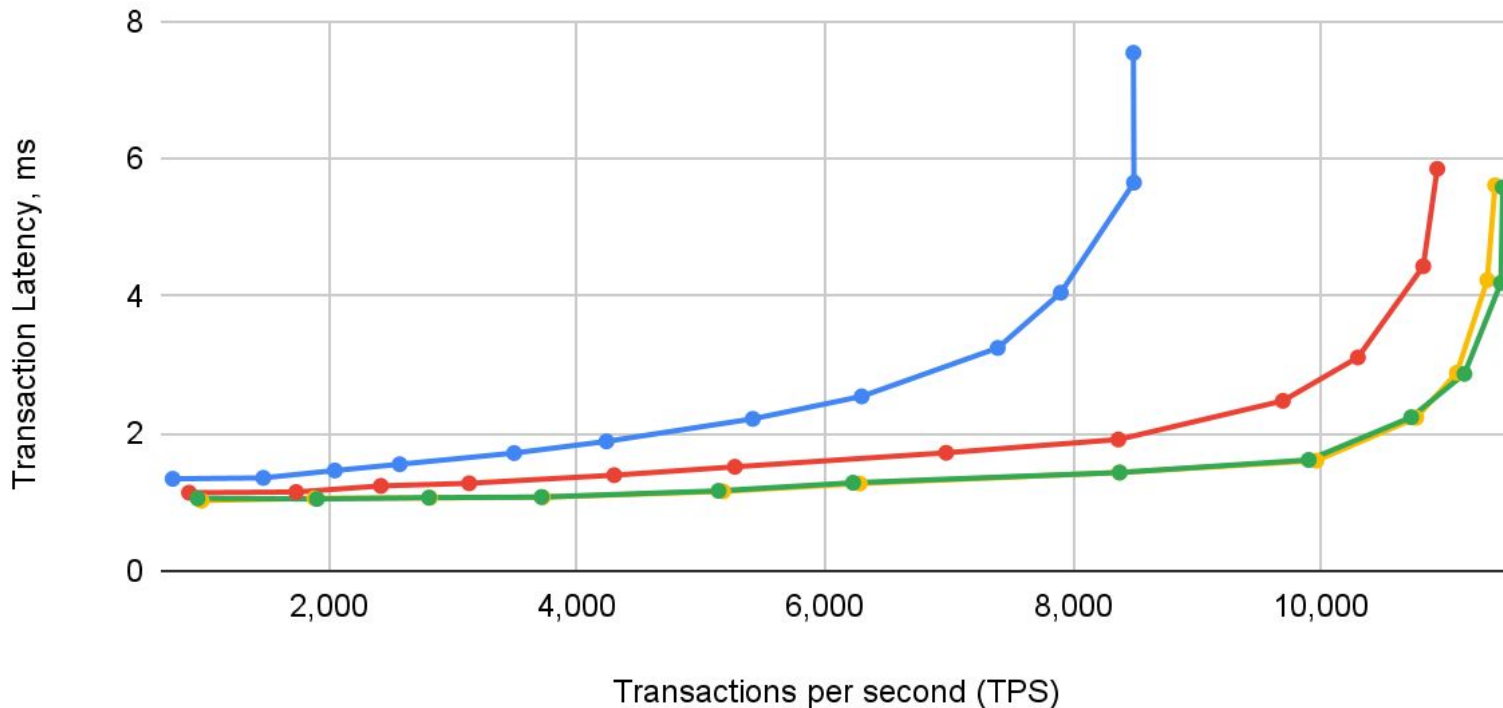
rsync time (lower is better)

3.1M files, 46 GB, ext4



pgbench - Latency vs. TPS

● threads ● native (no iothreads) ● native (iothreads) ● io_uring



Multiple IO Threads

Multiple IO Threads

Don't confuse IO threads with number of queues

threads backend is always multi-threaded

- compare number of physical CPU cores used

Linux **native** AIO and **io_uring** can run:

- in the main QEMU event loop
- in a dedicated I/O thread
- in multiple I/O threads

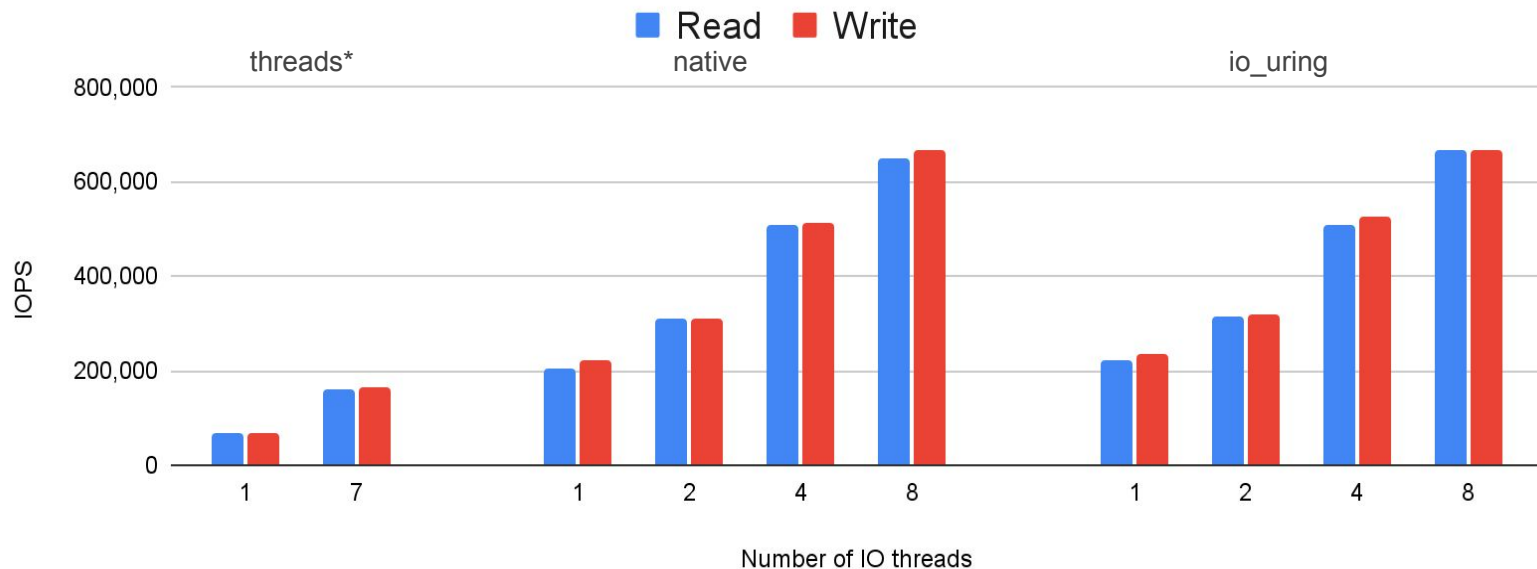
Latency vs Number of IO threads (lower is better)

Random Read/Write BS=4k, QD=1



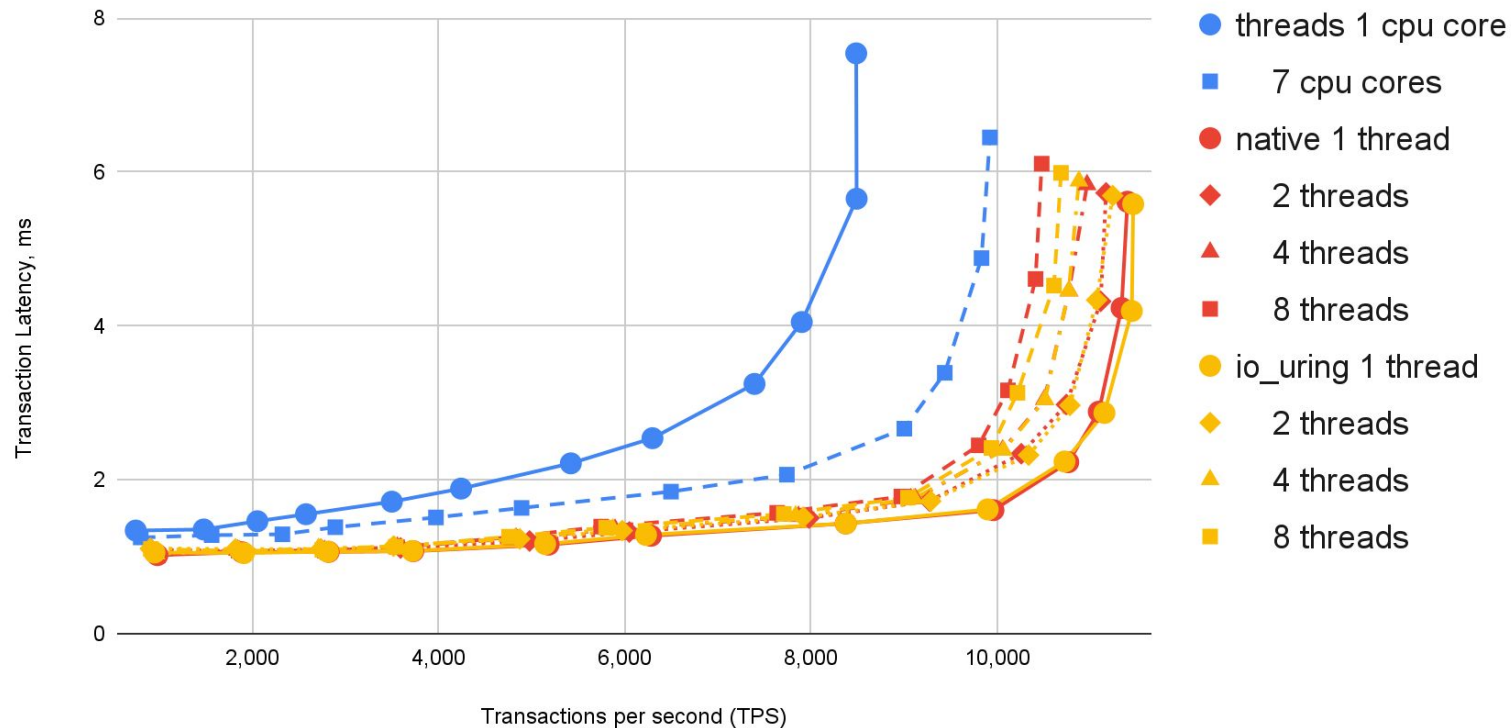
Max IOPS vs Number of IO threads

Random Read/Write, BS=4k, QD=64



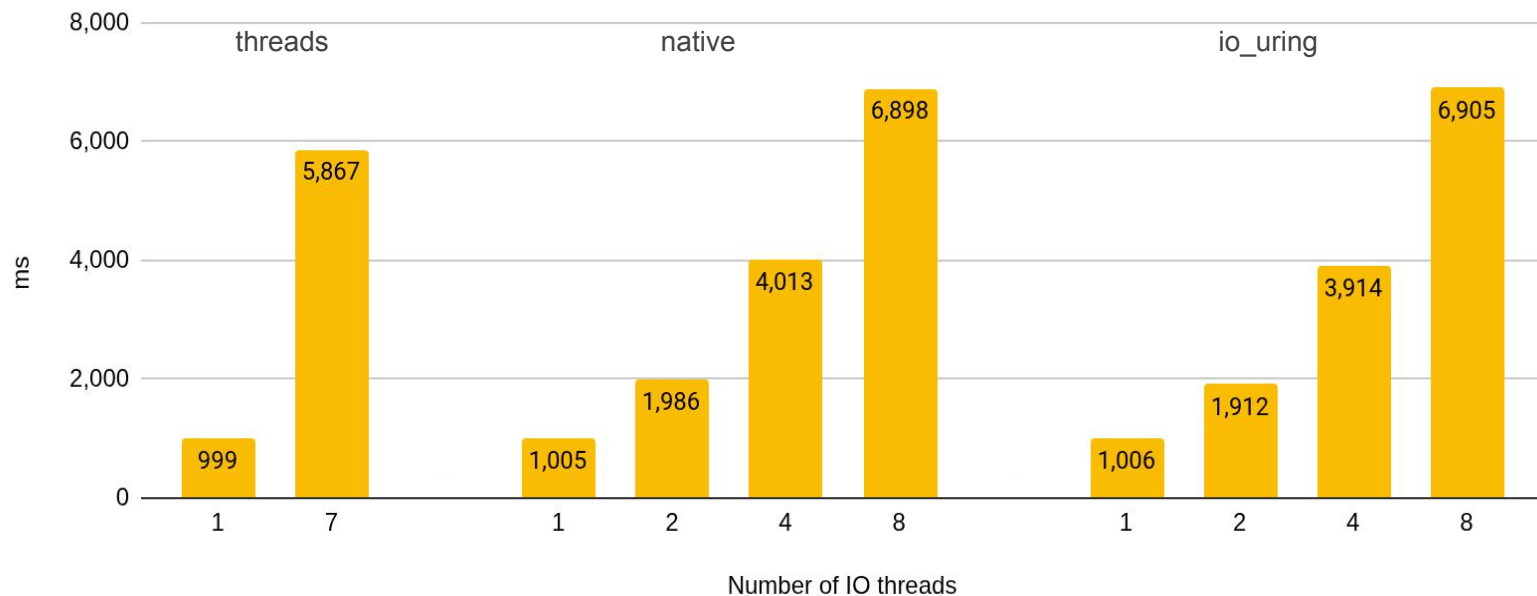
pgbench - Latency vs TPS

Multi-queue



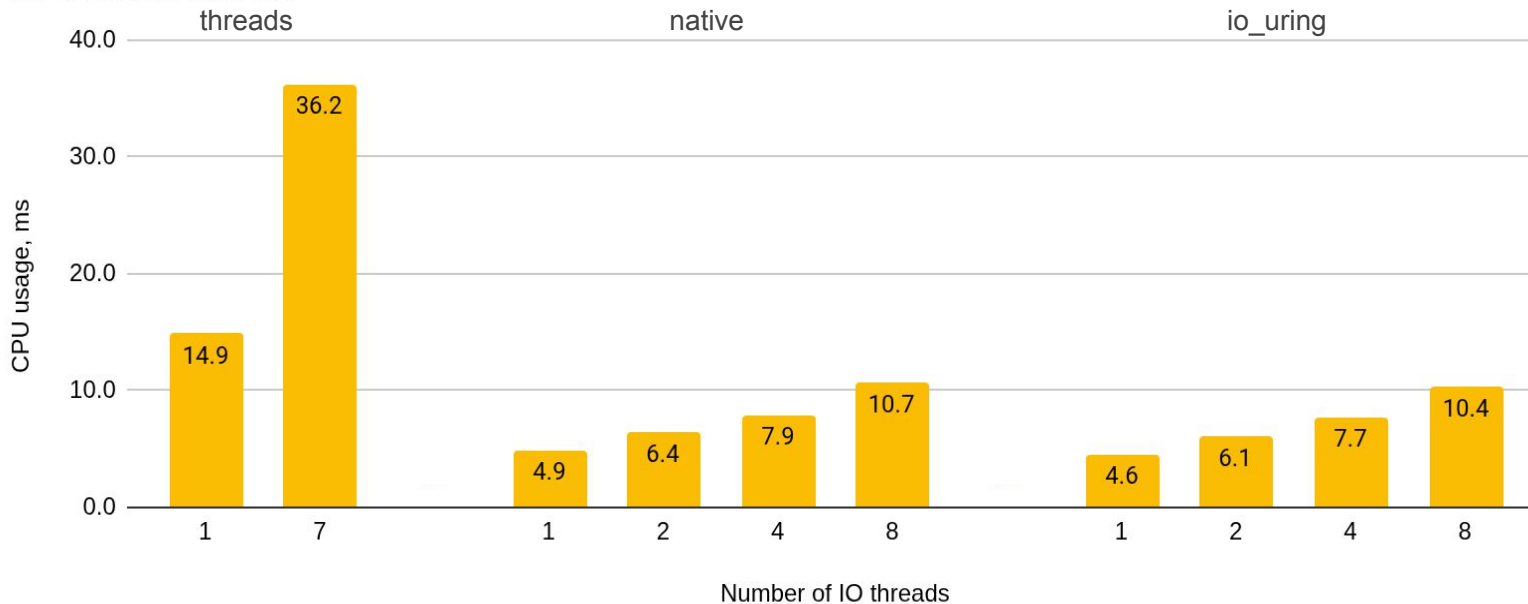
CPU Usage vs Number of IO threads

IOPS random read test



CPU Usage per 1000 IOPS

IOPS random read test

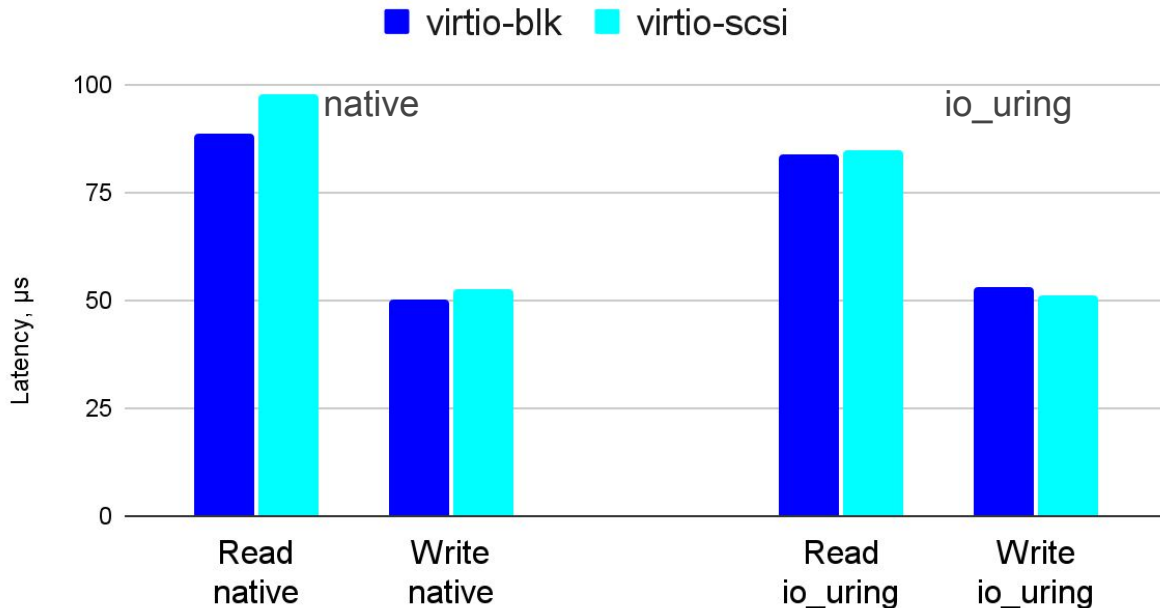


virtio-blk and virtio-scsi

virtio-blk vs virtio-scsi - Latency

virtio-blk vs virtio-scsi

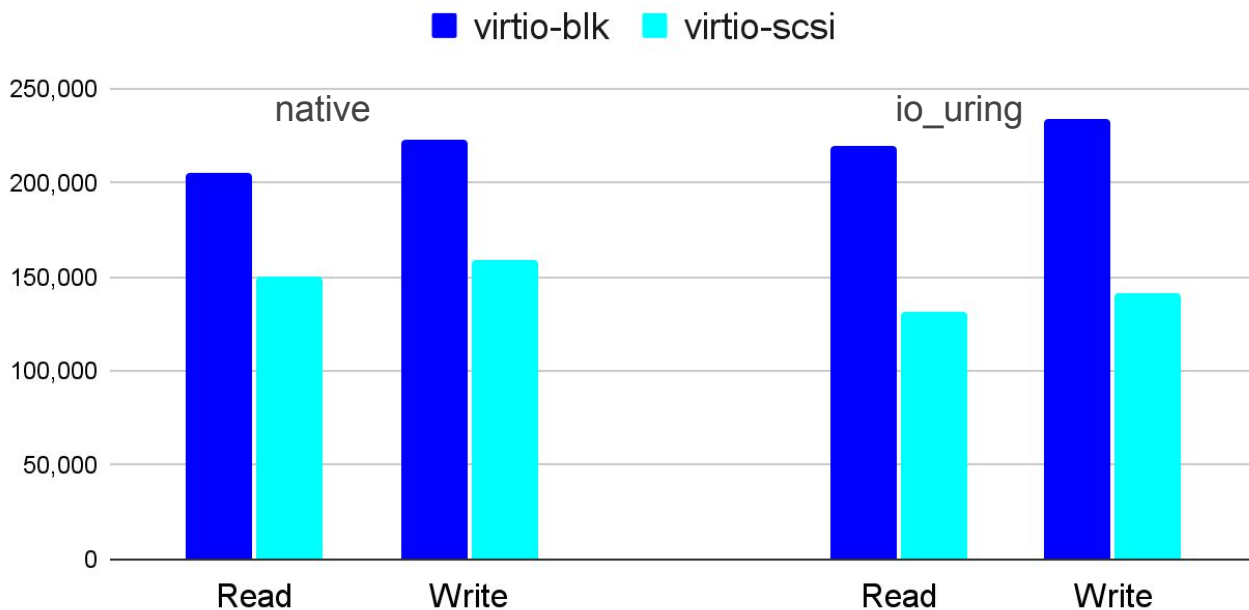
Latency (lower is better)



virtio-blk vs virtio-scsi - IOPS

virtio-blk vs virtio-scsi

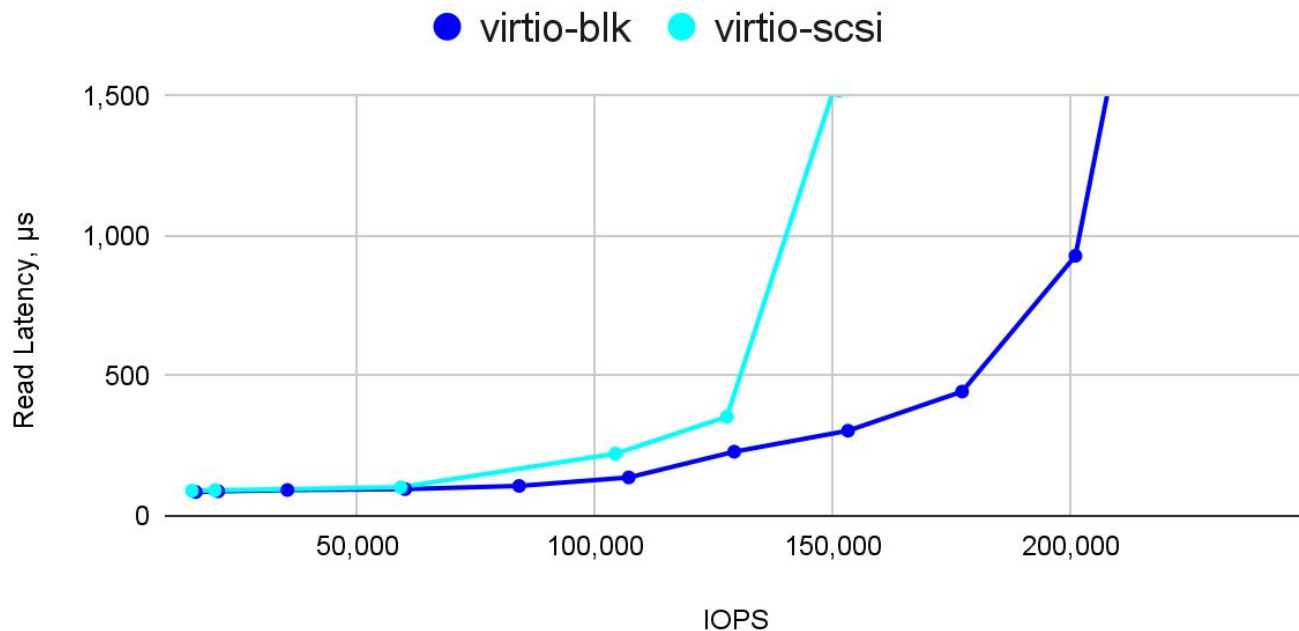
IOPS, BS=4k QD=64



virtio-blk vs virtio-scsi

Latency vs IOPS (aio=native)

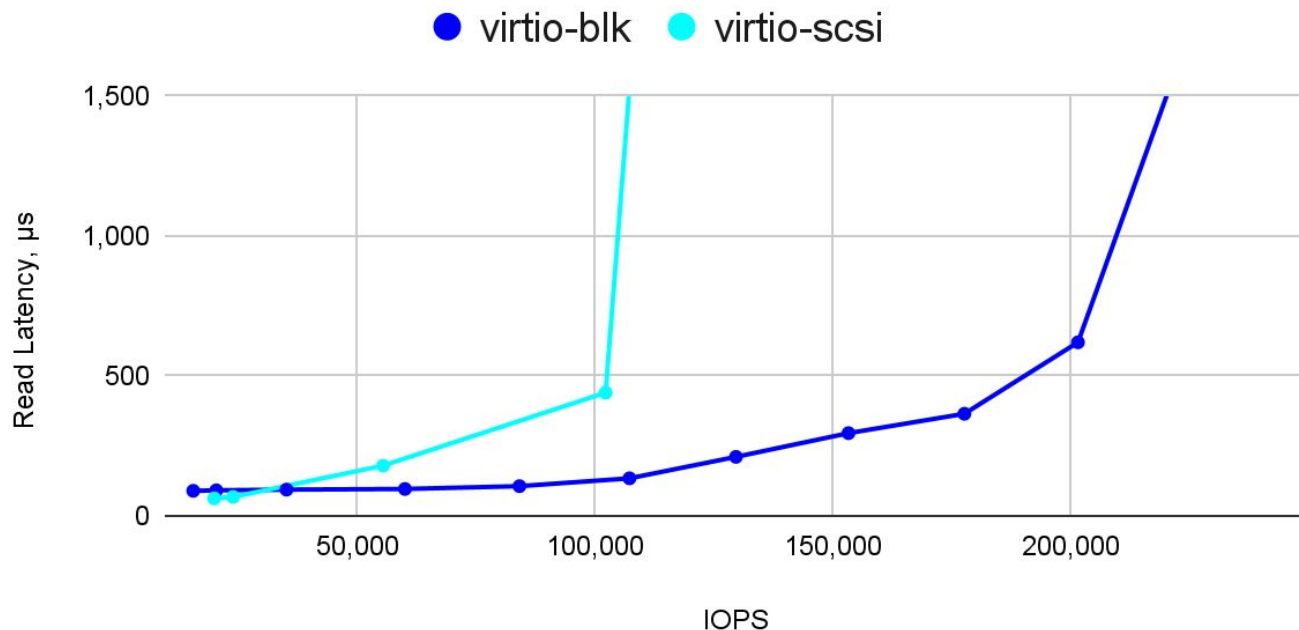
Random Read, BS=4kB, QD=64



virtio-blk vs virtio-scsi

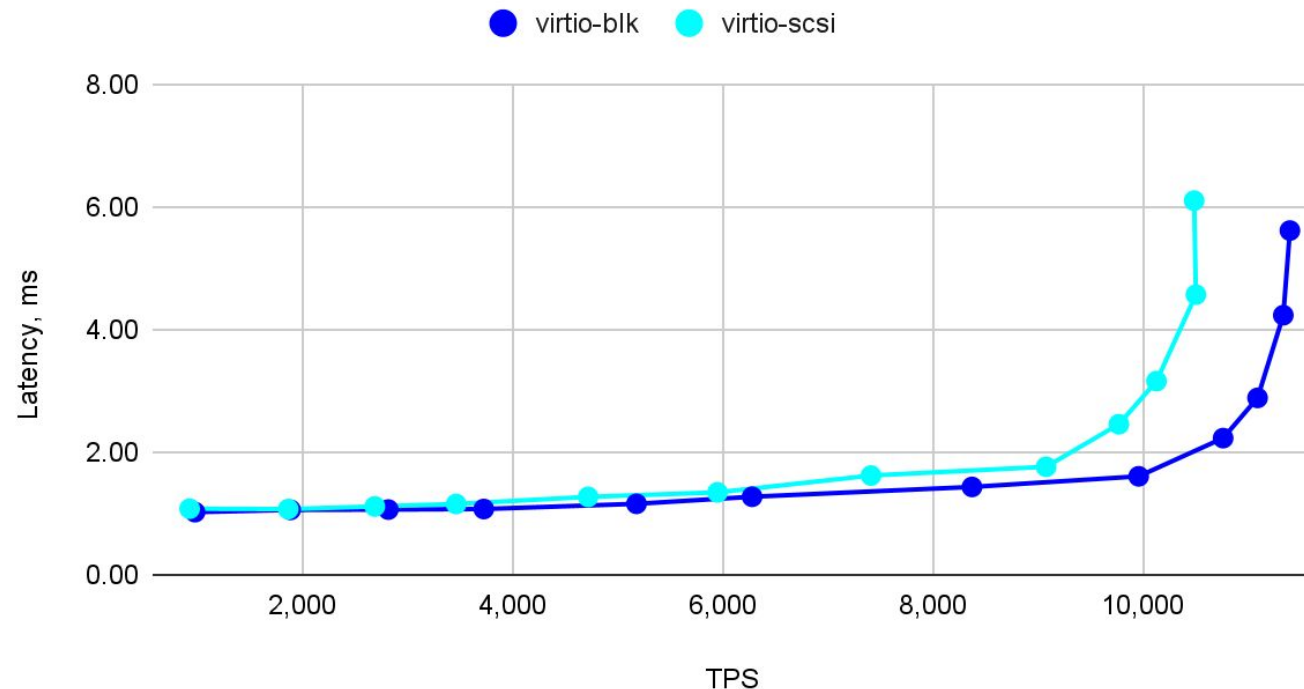
Latency vs IOPS (io_uring)

Random Read, BS=4kB, QD=64



virtio-blk vs virtio-scsi - pgbench

pgbench - Latency vs TPS

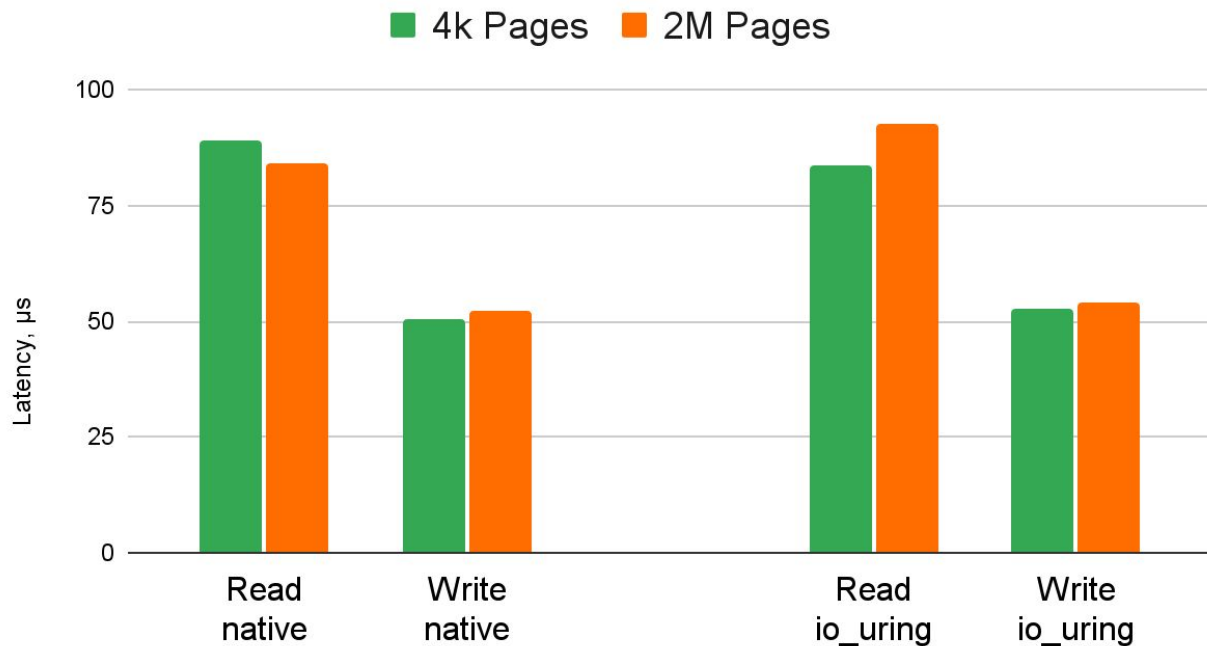


Huge Pages

Huge Pages - Latency

Huge Pages vs 4k Pages

Latency (lower is better)



Huge Pages - max IOPS

Huge Pages vs 4k Pages

IOPS, BS=4k QD=4



Conclusion

Limitations

virtio-blk

- max 28 disks on i440fx VM
- no SCSI commands

io_uring

- still has stability issues

```
[763134.450950] show_signal_msg: 7 callbacks suppressed
[763134.450954] IO iotthread1[37282]: segfault at 7ff05adffffe0 ip 00007ff06251e47c sp
00007ff05adffffe0 error 6 in liburing.so.2.9[347c,7ff06251d000+3000] likely on CPU 14
(core 6, socket 0)
[763134.452235] Code: c3 41 83 ca 02 eb d6 0f 1f 80 00 00 00 00 f3 0f 1e fa 48 83 ec
38 66 0f ef c0 64 48 8b 04 25 28 00 00 00 48 89 44 24 28 31 c0 <89> 14 24 48 89 e2 0f
11 44 24 08 89 4c 24 04 c7 44 24 0c 08 00 00
```

Summary

- No real benefit of using **io_uring** over **aio=native**. They have similar performance and indistinguishable for real workload.
- Enable **IOthread** - significant effect on latency
- **threads** is much slower at max IOPS, but for real application the difference is not so big
- **threads** can consume a lot of CPU. Avoid it
- **virtio-blk** is better than **virtio-scsi** at very high loads (100k IOPS). In typical use cases there is no significant difference.
- **Huge pages** have no significant effect on the storage performance. Can be beneficial for CPU efficiency with other workload, though.

Summary (cont.)

- Linux **native** aio may become synchronous in some circumstances. Always use it with **IOthread** to avoid blocking the VM
- **io_uring** can crash Qemu in some cases (with LVM in my tests). Use with caution!
- User-space access to storage, like libiscsi, librbd, NBD, NVMe disk images, use different path and may behave differently.
- File-based storage (qcow2 images, NFS) may behave differently.
- **Always test YOUR system and configuration!**

Fill in the CloudStack User Survey

Help us understand the CloudStack Ecosystem

