

# IO thread

Enabled IO thread option is ignored when controller is other than VirtIO SCSI Single.

## KVM without iothread

```
-device virtio-scsi-pci,id=scsihw0,bus=pci.0,addr=0x5

-drive file=/dev/zvol/nvmpool/data/vm-701-disk-1,if=none,id=drive-
scsi0,discard=on,format=raw,cache=none,aio=io_uring,detect-zeroes=unmap
-device scsi-hd,bus=scsihw0.0,channel=0,scsi-id=0,lun=0,drive=drive-
scsi0,id=scsi0,rotation_rate=1,bootindex=100

-drive file=/dev/zvol/hddpool/data/vm-701-disk-1,if=none,id=drive-
scsi1,discard=on,format=raw,cache=none,aio=io_uring,detect-zeroes=unmap
-device scsi-hd,bus=scsihw0.0,channel=0,scsi-id=0,lun=1,drive=drive-
scsi1,id=scsi1

-drive file=/dev/zvol/nvmpool/data/vm-701-disk-0,if=none,id=drive-
scsi2,discard=on,format=raw,cache=none,aio=io_uring,detect-zeroes=unmap
-device scsi-hd,bus=scsihw0.0,channel=0,scsi-id=0,lun=2,drive=drive-
scsi2,id=scsi2,rotation_rate=1
```

## KVM with iothread

- virtio-scsi-single controller chosen which create separate SCSI controller for each drive
- I/O thread enabled on virtioscsi2

```
-device virtio-scsi-pci,id=virtioscsi0,bus=pci.3,addr=0x1
-drive file=/dev/zvol/hddpool/data/vm-601-disk-0,if=none,id=drive-
scsi0,cache=writeback,discard=on,format=raw,aio=io_uring,detect-zeroes=unmap
-device scsi-hd,bus=virtioscsi0.0,channel=0,scsi-id=0,lun=0,drive=drive-
scsi0,id=scsi0,bootindex=100

-device virtio-scsi-pci,id=virtioscsi1,bus=pci.3,addr=0x2
-drive file=/dev/zvol/hddpool/data/vm-601-disk-1,if=none,id=drive-
scsi1,cache=writeback,discard=on,format=raw,aio=io_uring,detect-zeroes=unmap
-device scsi-hd,bus=virtioscsi1.0,channel=0,scsi-id=0,lun=1,drive=drive-
scsi1,id=scsi1

-object iothread,id=iothread-virtioscsi2
-device virtio-scsi-pci,id=virtioscsi2,bus=pci.3,addr=0x3,iothread=iothread-
virtioscsi2
-drive file=/dev/zvol/nvmpool/data/vm-601-disk-0,if=none,id=drive-
scsi2,cache=writeback,discard=on,format=raw,aio=io_uring,detect-zeroes=unmap
-device scsi-hd,bus=virtioscsi2.0,channel=0,scsi-id=0,lun=2,drive=drive-
```

```
scsi2,id=scsi2
```

From KVM manual:

```
-object iothread,id=id,poll-max-ns=poll-max-ns,poll-grow=poll-grow,poll-shrink=poll-shrink,aio-max-batch=aio-max-batch
```

Creates a dedicated event loop thread that devices can be assigned to. This is known as an IOThread.

By default device emulation happens in vCPU threads or the main event loop thread. This can become a scalability bottleneck. IOThreads allow device emulation and I/O to run on other host CPUs.

The id parameter is a unique ID that will be used to reference this IOThread from -device ...,iothread=id.

Multiple devices can be assigned to an IOThread. Note that not all devices support an iothread parameter.

IOThreads use an adaptive polling algorithm to reduce event loop latency. Instead of entering a blocking system call to monitor file descriptors and then pay the cost of being woken up when an event occurs, the polling algorithm spins waiting for events for a short time.

The algorithm's default parameters are suitable for many cases but can be adjusted based on knowledge of the workload and/or host device latency.

From:

<https://niziak.spox.org/wiki/> - **niziak.spox.org**

Permanent link:

<https://niziak.spox.org/wiki/vm:proxmox:kvm:iothread>

Last update: **2021/12/30 07:51**

