Virtio-fs for Kata Containers storage

Stefan Hajnoczi <stefanha@redhat.com>

February 19th 2019
Kata Containers Architecture Committee Call

Virtio-fs overview

New host↔guest file system:

- Takes advantage of co-location of host and guest – not a network protocol
- Designed with containers and lightweight VMs in mind
- virtio-9p replacement in Kata Containers

Developed by a team of file systems and virtualization developers

Project started and first patches published in 2018

https://virtio-fs.gitlab.io/

Usage in Kata Containers

Container storage:

- Container images
- Volumes

“kataShared” mount contains shared directories for each container

Virtio-fs is a drop-in replacement for virtio-9p

Kata configuration.toml

```
enable_virtio_fs = true
virtio_fs_daemon = "path/to/virtiofsd"
virtio_fs_cache_size = 2 # GB

# Cache mode: none (default), auto (NFS-like), always (cache everything)
virtio_fs_cache = "none"

# Use shared version metadata (experimental feature)
virtio_fs_shared_versions = false
```

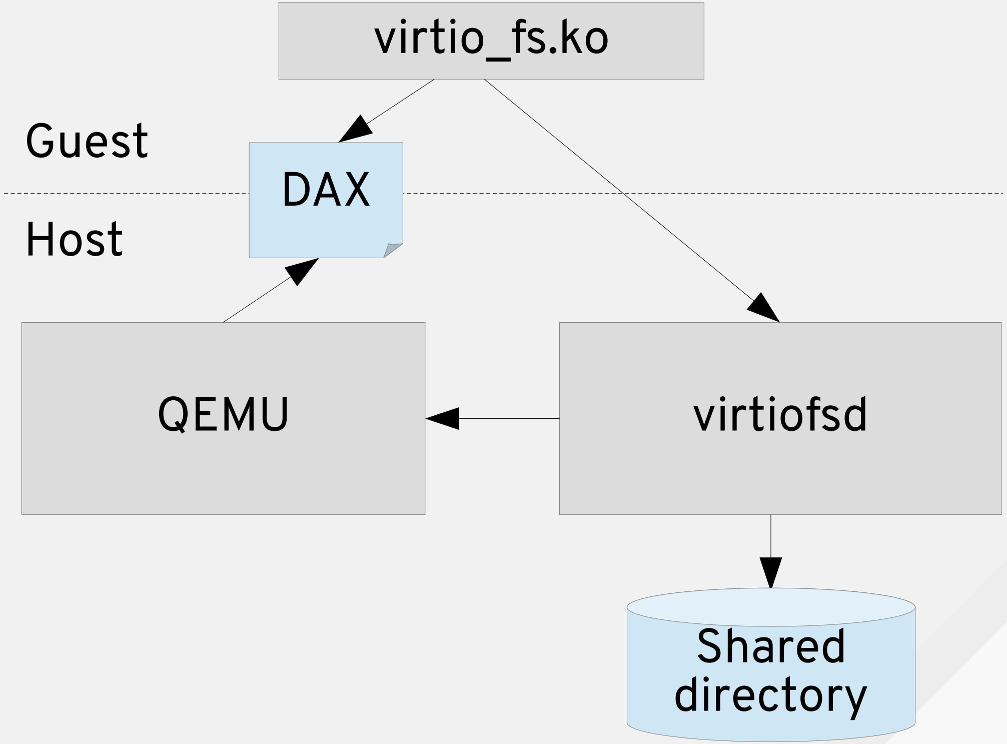
Architecture

Guest sends requests to virtiofs daemon

Daemon performs I/O in a sandboxed process

Daemon arranges DAX mappings with QEMU

DAX allows I/O without vmexits



Virtio-fs is built on FUSE

The core vocabulary is Linux FUSE with virtio-fs extensions

Guest acts as FUSE client, host acts as file system daemon

Arbitrary FUSE file system daemons cannot run over virtio-fs

virtiofsd is a FUSE file system daemon and a vhost-user device

Alternative file system daemon implementations are possible

- Other storage backends, security features, etc

DAX – host page cache sharing

Regions of files can be mapped into guest memory space

Guest accesses these pages from host page cache

- Avoids a data copy into guest RAM
- Reduces memory footprint
- Reduces vmexits when page is accessed repeatedly
- Coherent access to shared writable files on host (MAP_SHARED)

Already functional with ongoing optimization work

Security

File system daemon runs in a sandboxable process on the host

One step closer to non-root QEMU in Kata Containers

Virtiofsd passes through uid/gid values (same model as Kata 9p)

Performance

Notes:

32 GB host RAM

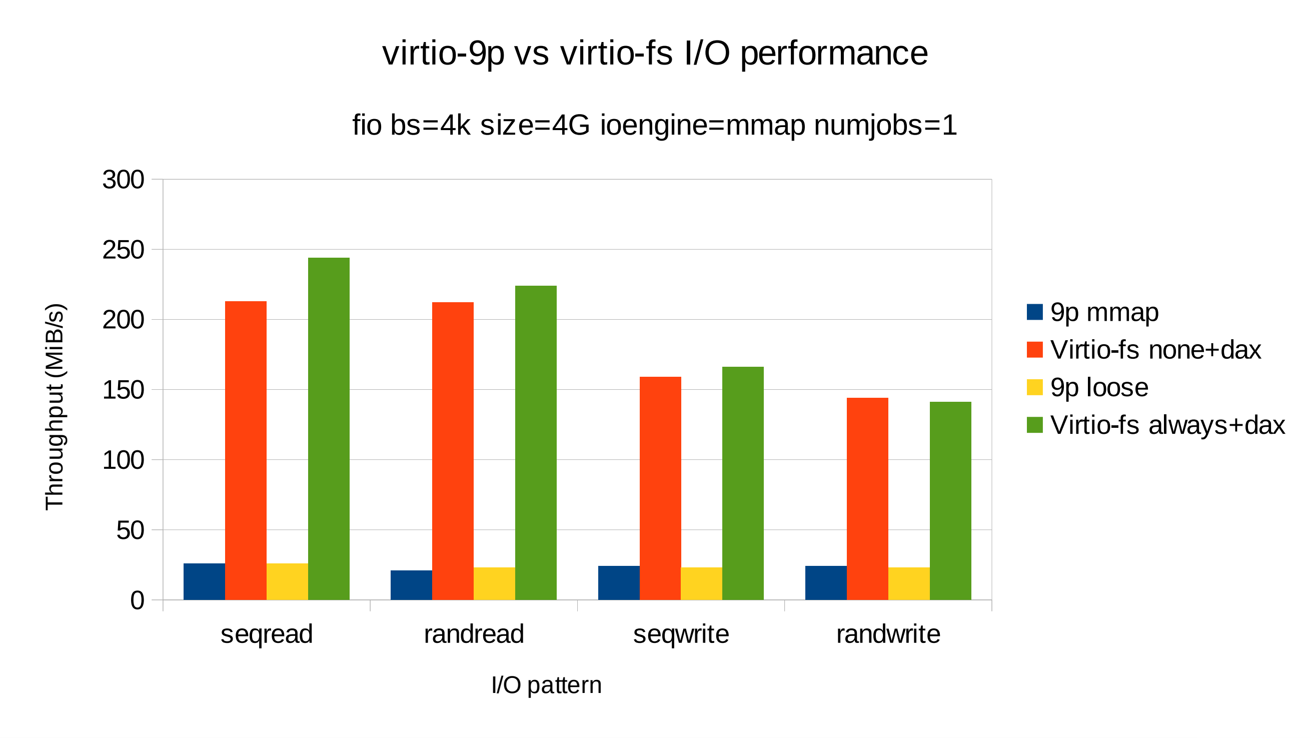
8 GB guest RAM

8 GB DAX window

Kernel version 4.20.7-100.fc28.x86_64

XFS host file system on PCI SSD

Intel Xeon E5-2620 v2 @ 2.10GHz (2 sockets x 6 cores)



Status

Working on getting the following changes merged (roughly in order):

- VIRTIO specification
  - Spec extension for shared memory resources
  - Virtio-fs device type
- Linux kernel FUSE client
- QEMU
  - vhost-user-fs device
  - Virtiofsd vhost-user device backend
- Kata Containers
  - Runtime & agent changes

Ongoing performance optimization (DAX, cache modes)

Questions?

https://virtio-fs.gitlab.io/

Contributors:
David Gilbert, Miklos Szeredi, Vivek Goyal, myself