German Maglione <gmaglione@redhat.com>
Hanna Czenczek <hreitz@redhat.com>

KVM Forum 2024

Section 1

Outline

Red Hat

Outline

- Live migration state & Kubevirt integration
- Performance
- Connecting to host (/dev/fuse, VDUSE)
- Non-passthrough drivers
- Future plans, TODOs, experiments, etc.

Section 2

Live migration state

Red Hat

Live migration state

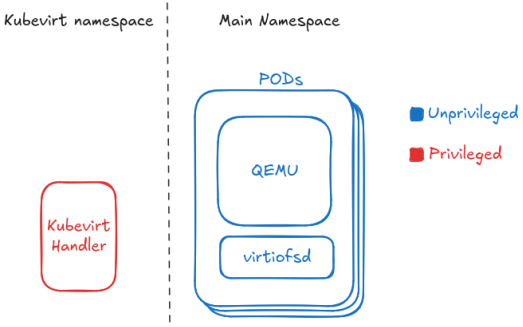
- It is our biggest and most complex feature
- We support a basic path-based migration since v1.11.0
  - Does not requires any privileges
  - Good enough for readonly/exclusive access file systems
- We are working to support a more robust file-handle-based mode[0]
  - Requires privileges: `CAP_DAC_READ_SEARCH`
- We started working on this motivated by the Kubevirt use case
- OpenStack will also take advantage of this feature

[0] `https://gitlab.com/virtio-fs/virtiofsd/-/merge_requests/244`

Red Hat

Kubevirt integration

Sharing ConfigMap, Secret, ServiceAccount & DownwardAPI

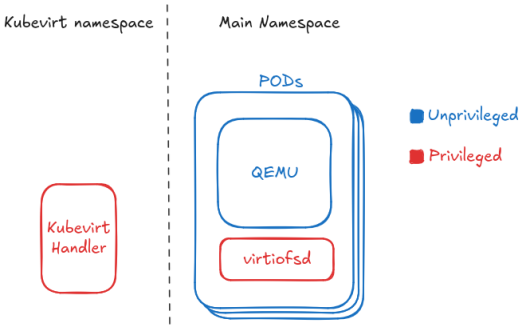


- Runs in an **unprivileged** container
- It will use the path-based migration mode

Red Hat

kubevirt integration

Sharing PVCs: Current state

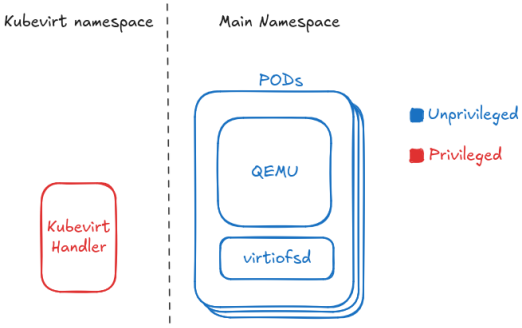


- Runs in a **privileged** container
- Can create root-owned files; problematic with mixed workloads: VMs + containers
- In the (near) future k8s will ban privileged user containers

Red Hat

kubevirt integration

Sharing PVCs: Can virtiofsd run unprivileged? Yes, but...



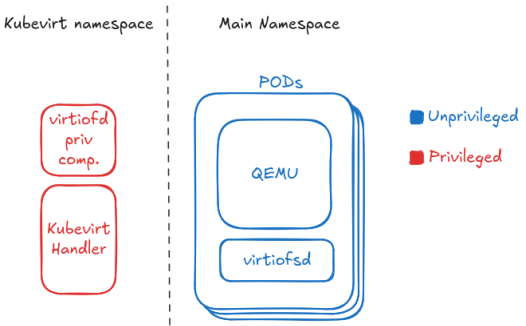
[0] virtio-fs – present and future (KVM Forum 2023)

- Non exclusive access to the PVCs
- Files can be removed/renamed during the migration
- Path-based migration mode is not robust enough[0]
- We need file handles

Red Hat

kubevirt integration

Sharing PVCs: Using file handles without privileges



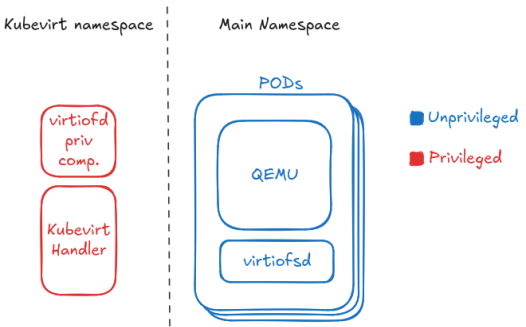
- We are working with Alice Frosi from the Kubevirt team
- A new privileged component as part of Kubevirt infrastructure[0]
- It is stateless, so no data needs to be migrated

[0] `https://kubevirt.io/user-guide/architecture/`

Red Hat

kubevirt integration

Sharing PVCs: Using file handles without privileges



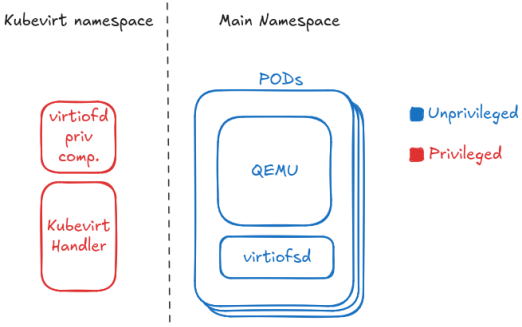
- Uses seccomp notify[0] to intercept `name_to_handle_at(2)` and `open_by_handle_at(2)`
- It runs these syscalls on behalf of virtiofsd, returning an HMAC-signed file handle

[0] `https://brauner.io/2020/07/23/seccomp-notify.html`

Red Hat

kubevirt integration

Sharing PVCs: Using file handles without privileges



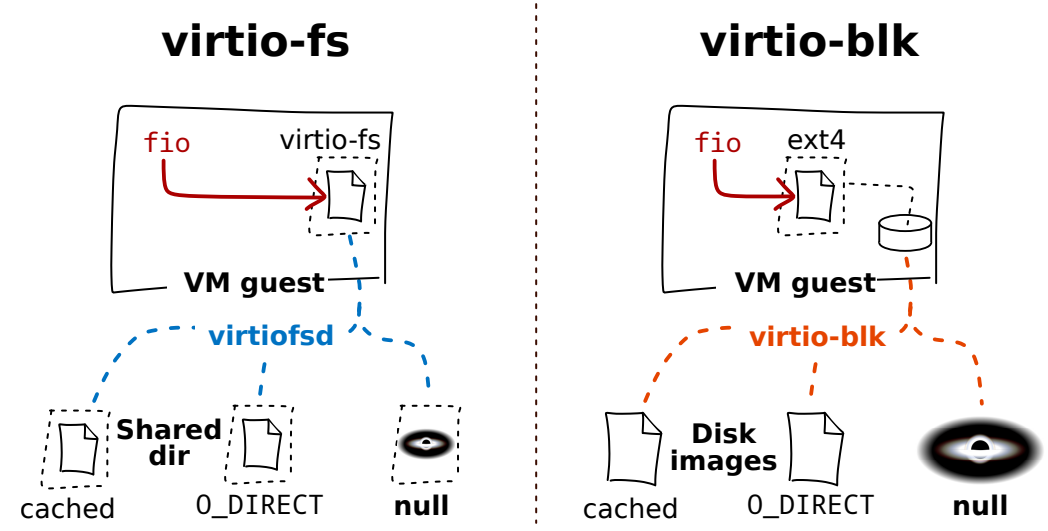
- In the future it can be extended to support SELinux and POSIX ACLs

Section 3

Performance

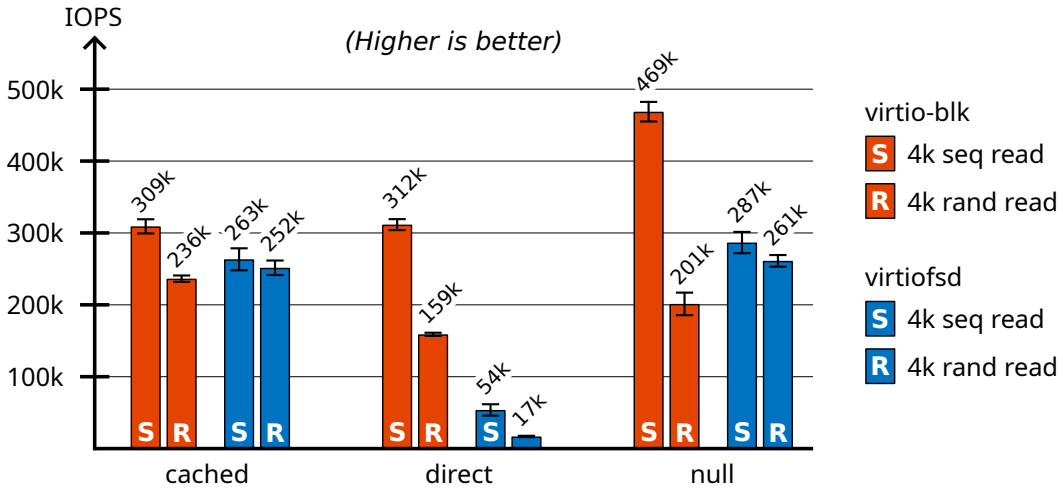
Red Hat

Benchmarking Methodology



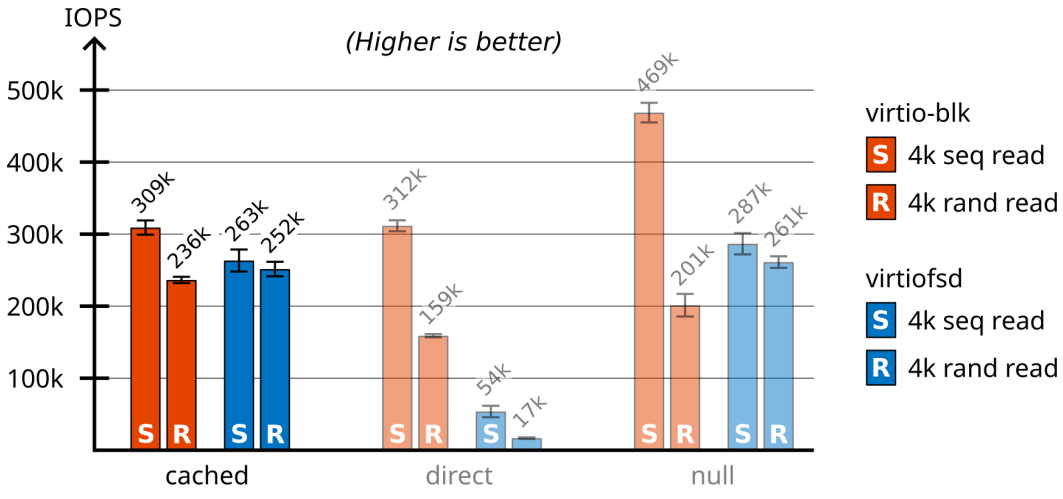
Red Hat

Overall Performance



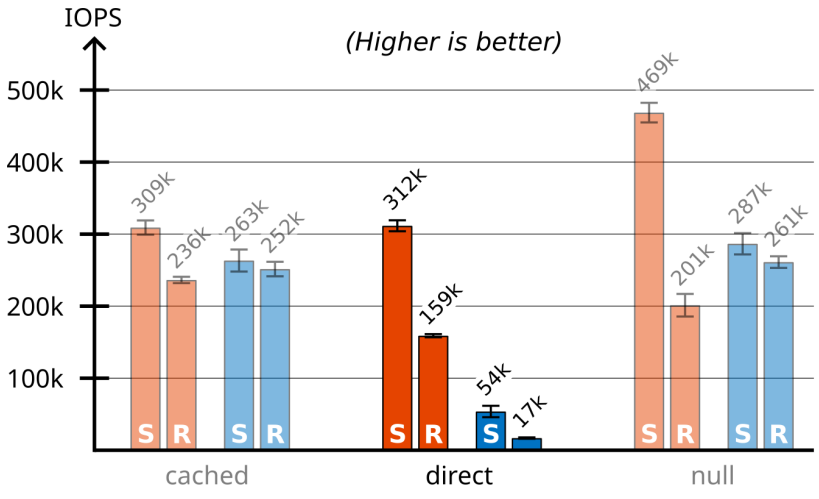
Red Hat

Overall Performance



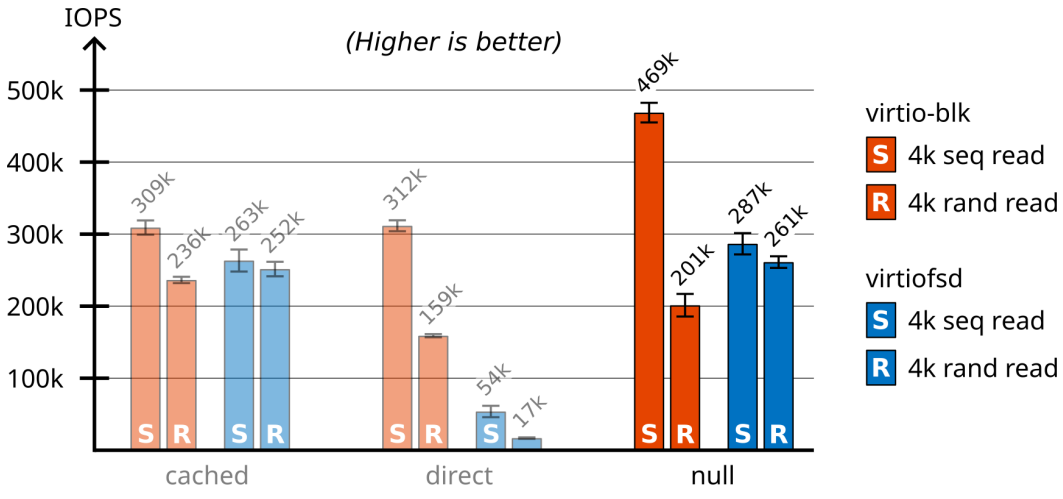
Red Hat

Overall Performance



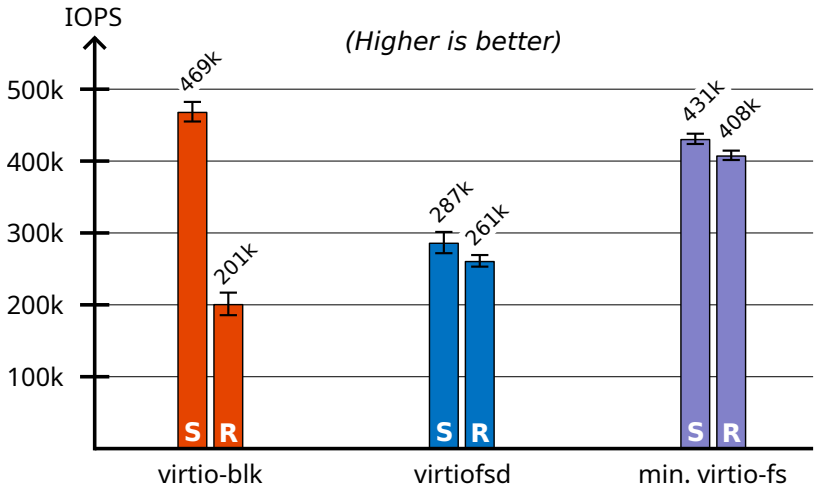
Red Hat

Overall Performance



Red Hat

virtio-fs/FUSE Performance (Null I/O)



S 4k seq read

R 4k rand read

Red Hat

Takeaway

- virtio-fs interface is good
- virtiofsd could make better use of it
- Need proper (asynchronous) I/O handling

- Non-I/O performance (FS ops) may be more important
- Guest io_uring with FUSE may require `Poll`?
- Multi-queue, multi-threading (4 queues: +90%, 408k → 783k)

Red Hat

Takeaway

- virtio-fs interface is good
- virtiofsd could make better use of it
- Need proper (asynchronous) I/O handling

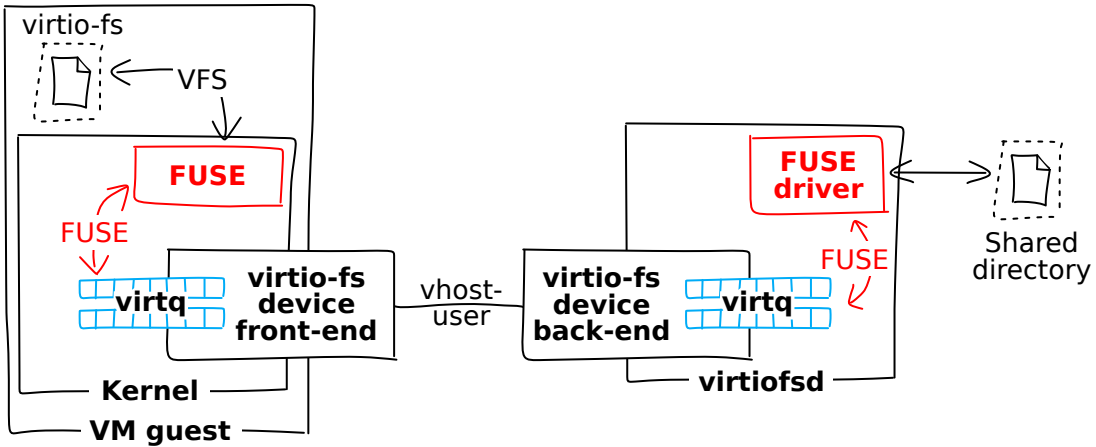
- Non-I/O performance (FS ops) may be more important
- Guest io_uring with FUSE may require `Poll`?
- Multi-queue, multi-threading (4 queues: +90%, 408k → 783k)

Section 4

Expanding the Scope

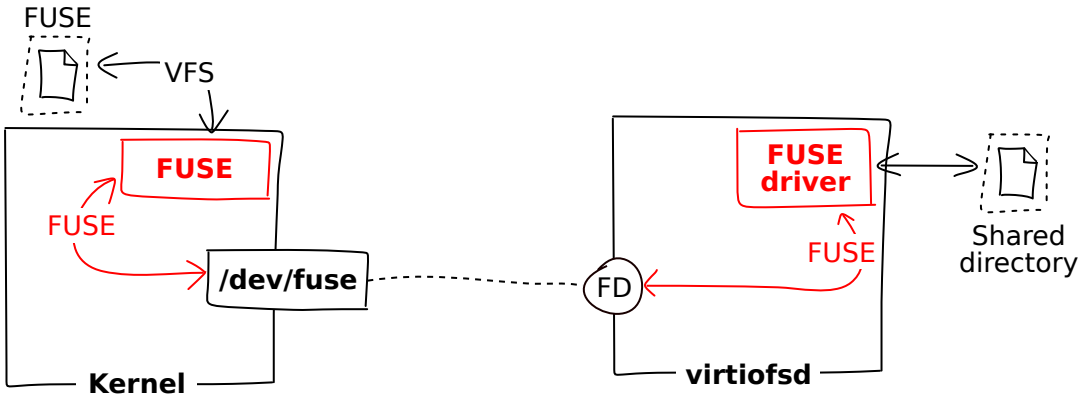
Red Hat

virtio-fs = FUSE over virtio



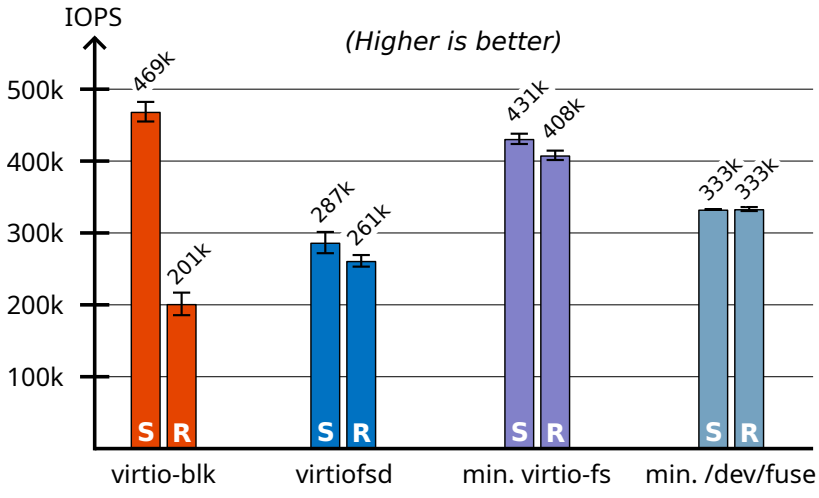
Red Hat

Connecting to the Host: /dev/fuse



Red Hat

Connecting to the Host: VDUSE

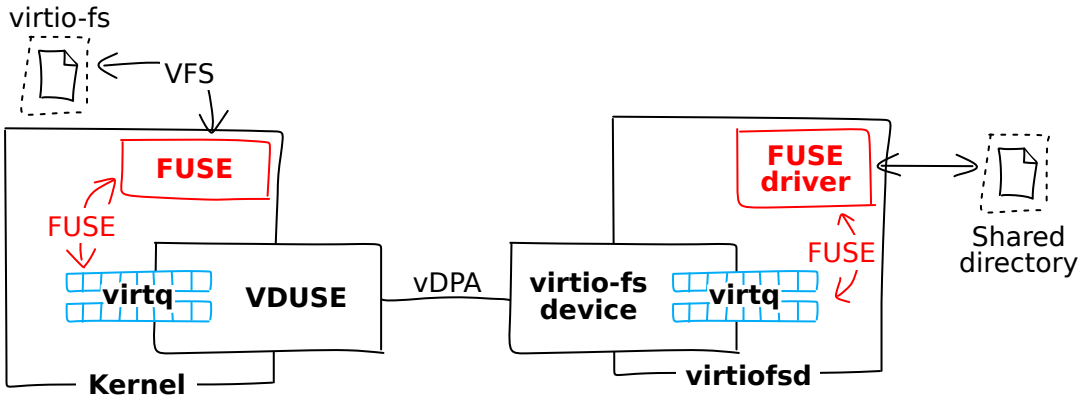


S 4k seq read

R 4k rand read

Red Hat

Connecting to the Host: VDUSE



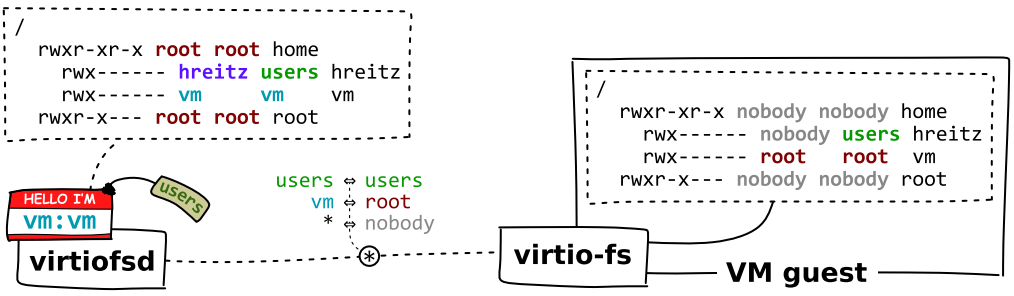
(Credit to Eugenio Pérez Martín <eperezma@redhat.com>)

Red Hat

Rich Passthrough

Transform filesystems:

- Enforce read-only
- UID/GID mapping



Red Hat

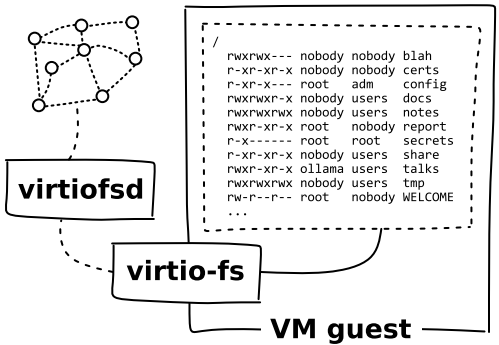
Other Filesystems

Bypass host VFS: (permissions)

- Network filesystems
- Archives

Completely virtual filesystems?

- LLM? virtiofsd+AI?



Red Hat

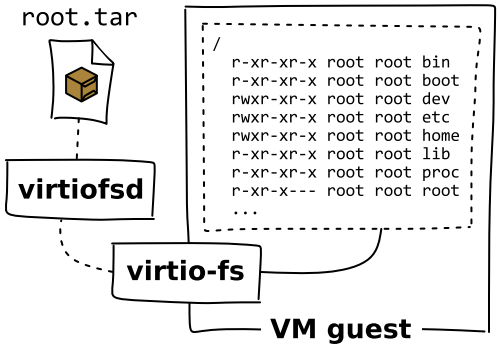
Other Filesystems

Bypass host VFS: (permissions)

- Network filesystems
- Archives

Completely virtual filesystems?

- LLM? virtiofsd+AI?



Section 5

More stuff

Red Hat

Future plans

- Read-only sharing
  - https://gitlab.com/virtio-fs/virtiofsd/-/merge_requests/220
- Idmapped mount
  - https://gitlab.com/virtio-fs/virtiofsd/-/merge_requests/245
- Improving virtiofsd to be consumed as a library
  - https://gitlab.com/virtio-fs/virtiofsd/-/merge_requests/248
- Reflink support
- ... and more


The End.
Thanks for listening!