Introduce a SPDK vhost FS target to accelerate File Access in VMs and containers

Changpeng Liu, Intel

Xiaodong Liu, Intel

Notices & Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at intel.com.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/benchmarks .

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

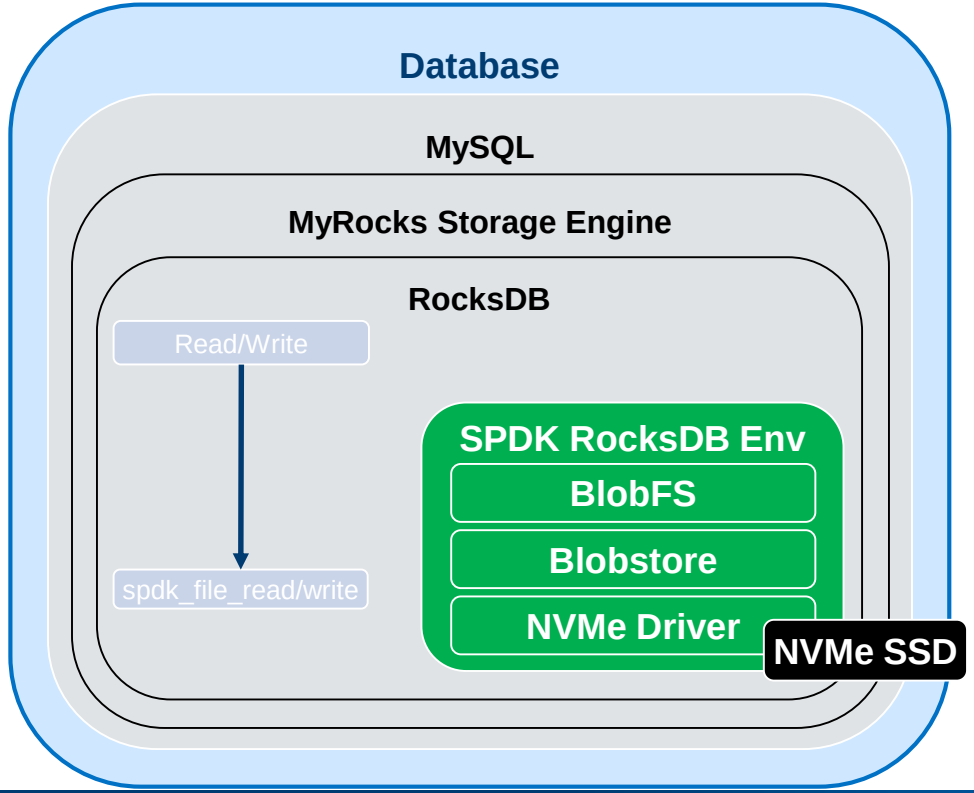
Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Agenda

- Introduction to SPDK Vhost-fs
- SPDK Vhost-fs with Kata Container
- Future plans

Introduction to SPDK Vhost-fs

Application Acceleration (Local Storage)

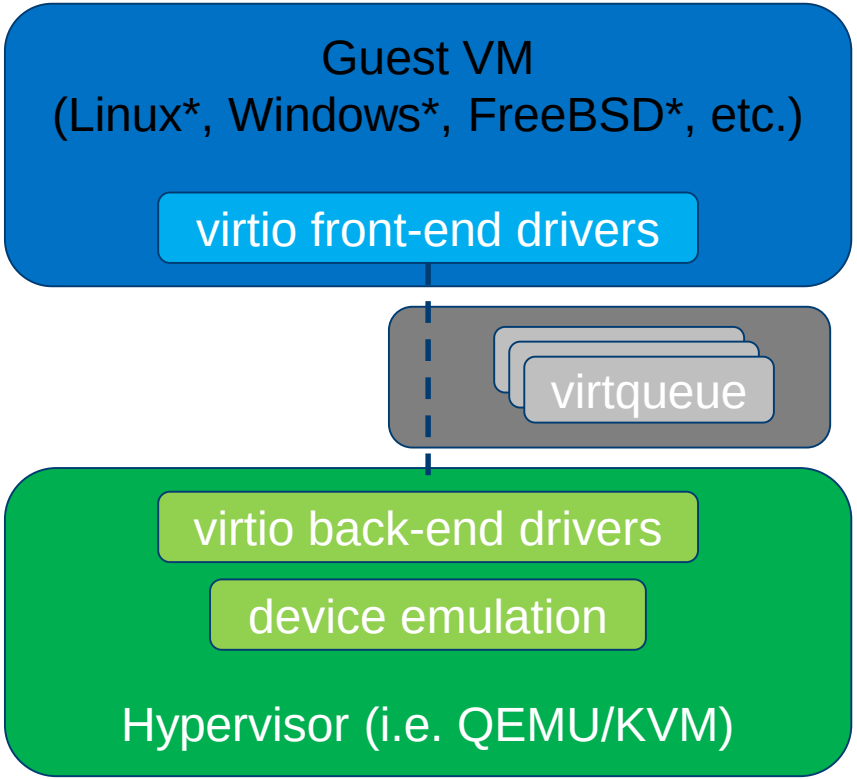


Implementation of RocksDB “env” abstraction

- Drop-in storage engine replacement
- Accelerate application access to local storage
- Benefits: removes latency and improves I/O consistency

What if running RocksDB in a virtual environment? Is there any protocol can transfer file APIs between VM and Host ?

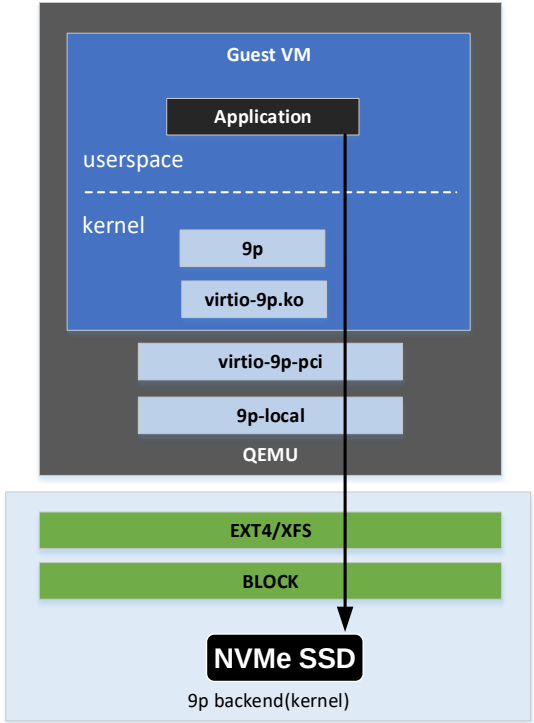
virtio

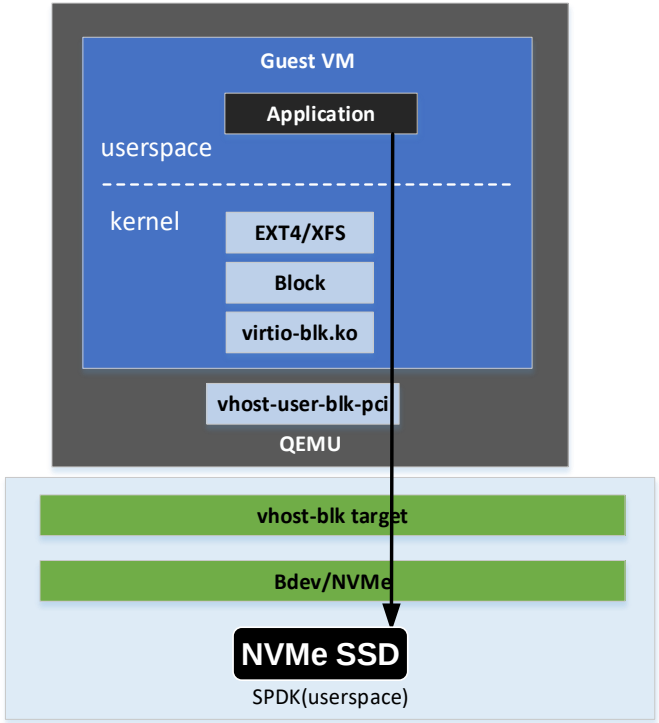


- Paravirtualized driver specification
- Common mechanisms and layouts for device discovery, I/O queues, etc.
- virtio device types include:
  - virtio-net
  - virtio-blk
  - virtio-scsi
  - virtio-9p
  - virtio-fs

Optional solutions using file APIs in VM

- Using 9p as the file transport protocol
- Format file system with block device



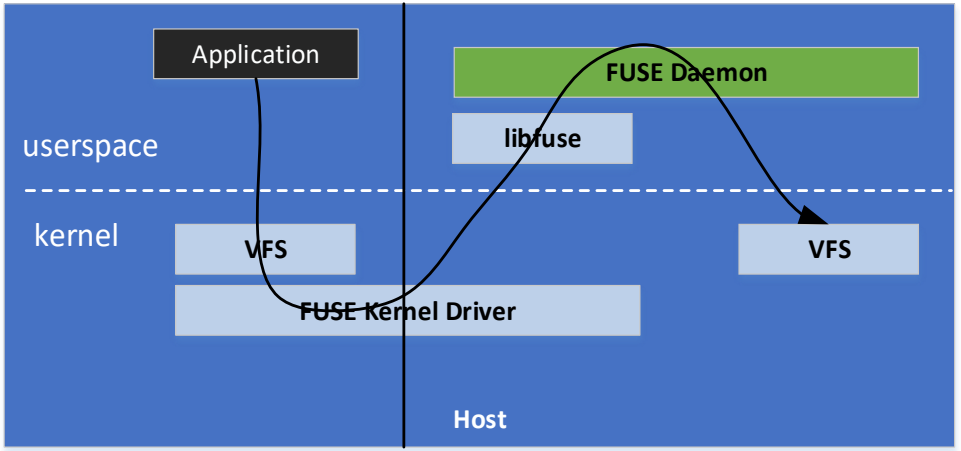


Introduction to FUSE

- FUSE (Filesystem in Userspace) is an interface for userspace programs to export a filesystem to the Linux kernel.
- The FUSE project consists of two components:

  - fuse kernel module and the libfuse userspace library.

  - libfuse provides the reference implementation for communicating with the FUSE kernel module.



Example usage of FUSE(passthrough)

Virtio-fs

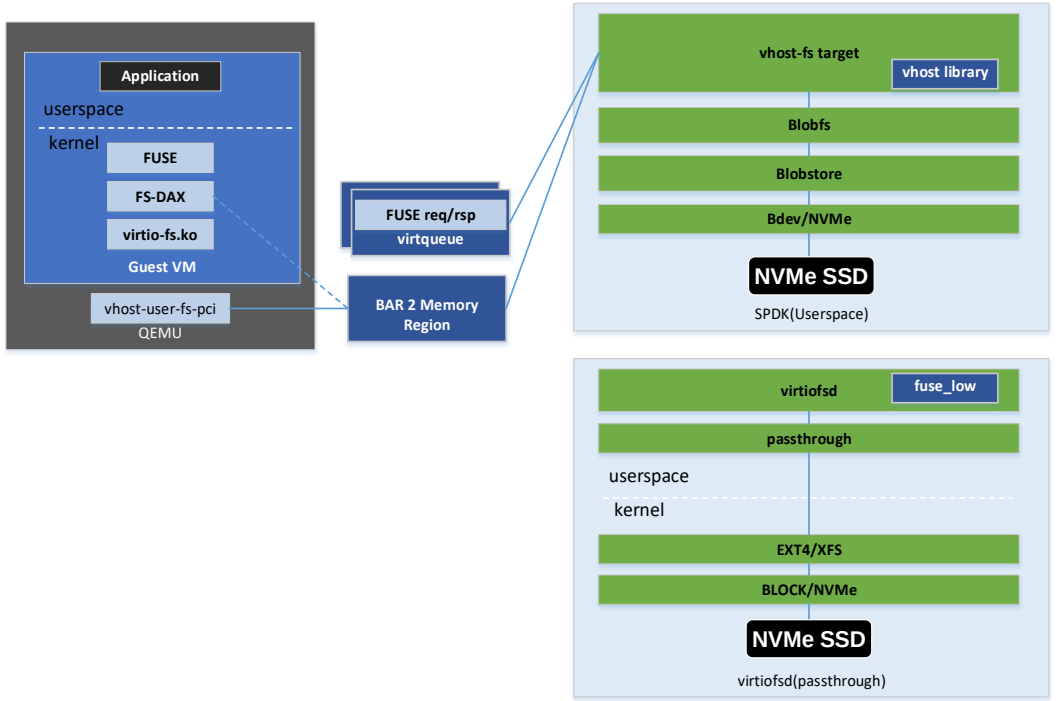
- virtio-fs is a shared file system that lets virtual machines access a directory tree on the host. Unlike existing approaches, it is designed to offer local file system semantics and performance. This is especially useful for lightweight VMs and container workloads, where shared volumes are a requirement.
- virtio-fs was started at Red Hat and is being developed in the Linux, QEMU, FUSE, and Kata Containers communities that are affected by code changes.
- virtio-fs uses FUSE as the foundation. A VIRTIO device carries FUSE messages and provides extensions for advanced features not available in traditional FUSE.
- DAX support via virtio-pci BAR from host huge memory.

SPDK Vhost-fs Target vs. Virtiofsd

- Eliminate userspace/kernel space context switch by providing a user space file system
- IO thread model

  - SPDK uses one poller to poll all the virtqueues while virtiofsd uses one thread per queue

- Page cache in Host can be shared for virtiofsd
- Easy to add new features in userspace



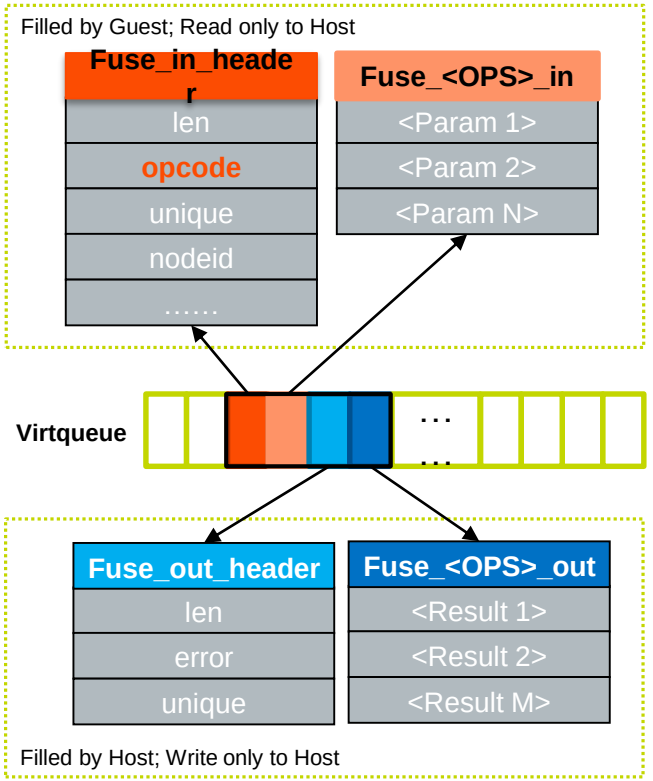
SPDK Blobfs APIs vs. FUSE

- Open, read, write, close, delete, rename, sync interface to provide POSIX similar APIs
- Asynchronous APIs provided

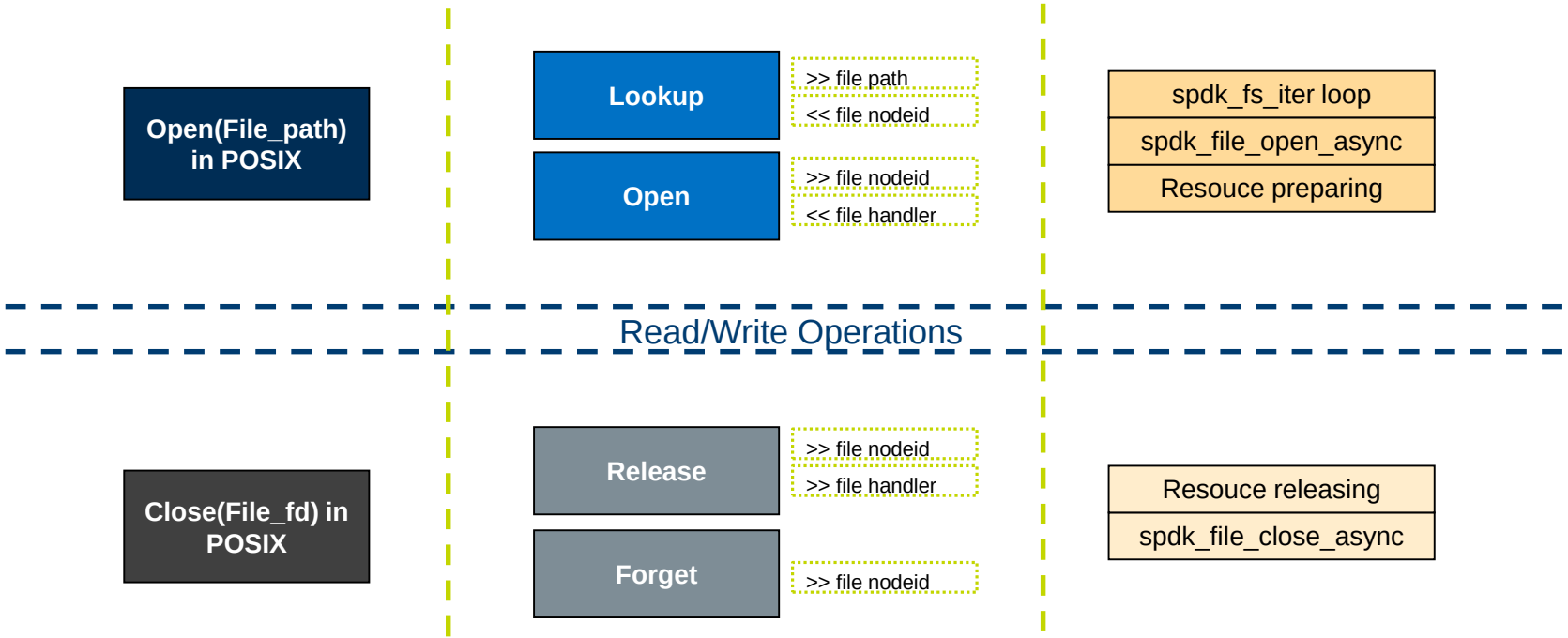
FUSE Command	Blobfs API
Lookup	spdk_fs_iter_first, spdk_fs_iter_next
Getattr	spdk_fs_file_stat_async
Open	spdk_fs_open_file_async
Release	spdk_file_close_async
Create	spdk_fs_create_file_async
Delete	spdk_fs_delete_file_async
Read	spdk_file_readv_async
Write	spdk_file_writev_async
Rename	spdk_fs_rename_file_async
Flush	spdk_file_sync_async

Operation Mapping of FUSE in Virtqueue

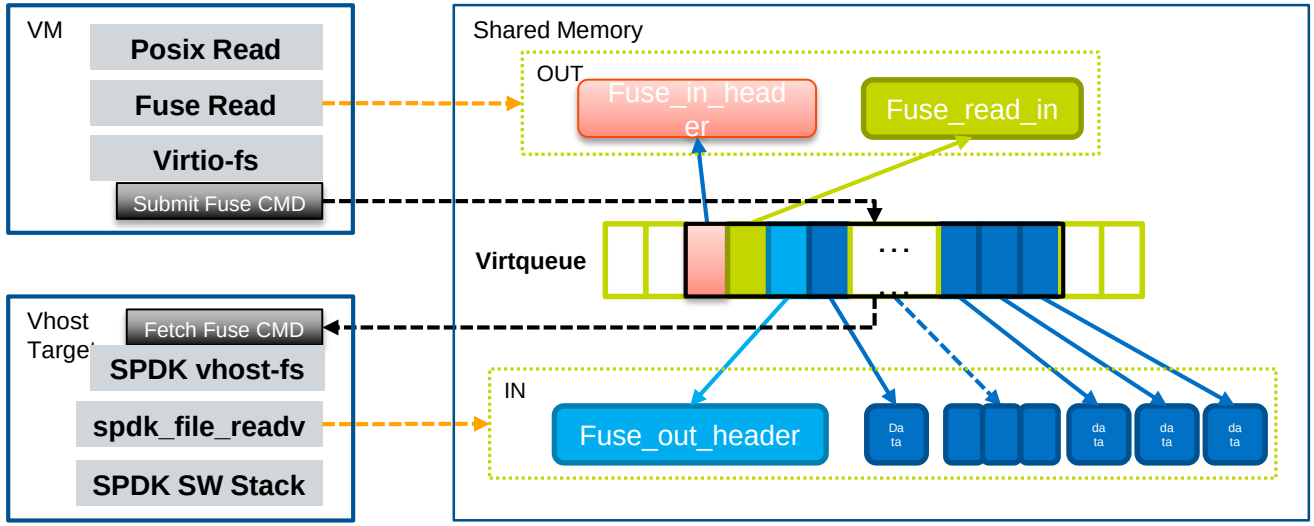
- General FUSE command has 2 parts: request and response.
- General FUSE request is consisted with IN header and operation specific IN parameters.
- General FUSE response is consisted with OUT header and operation specific OUT results.



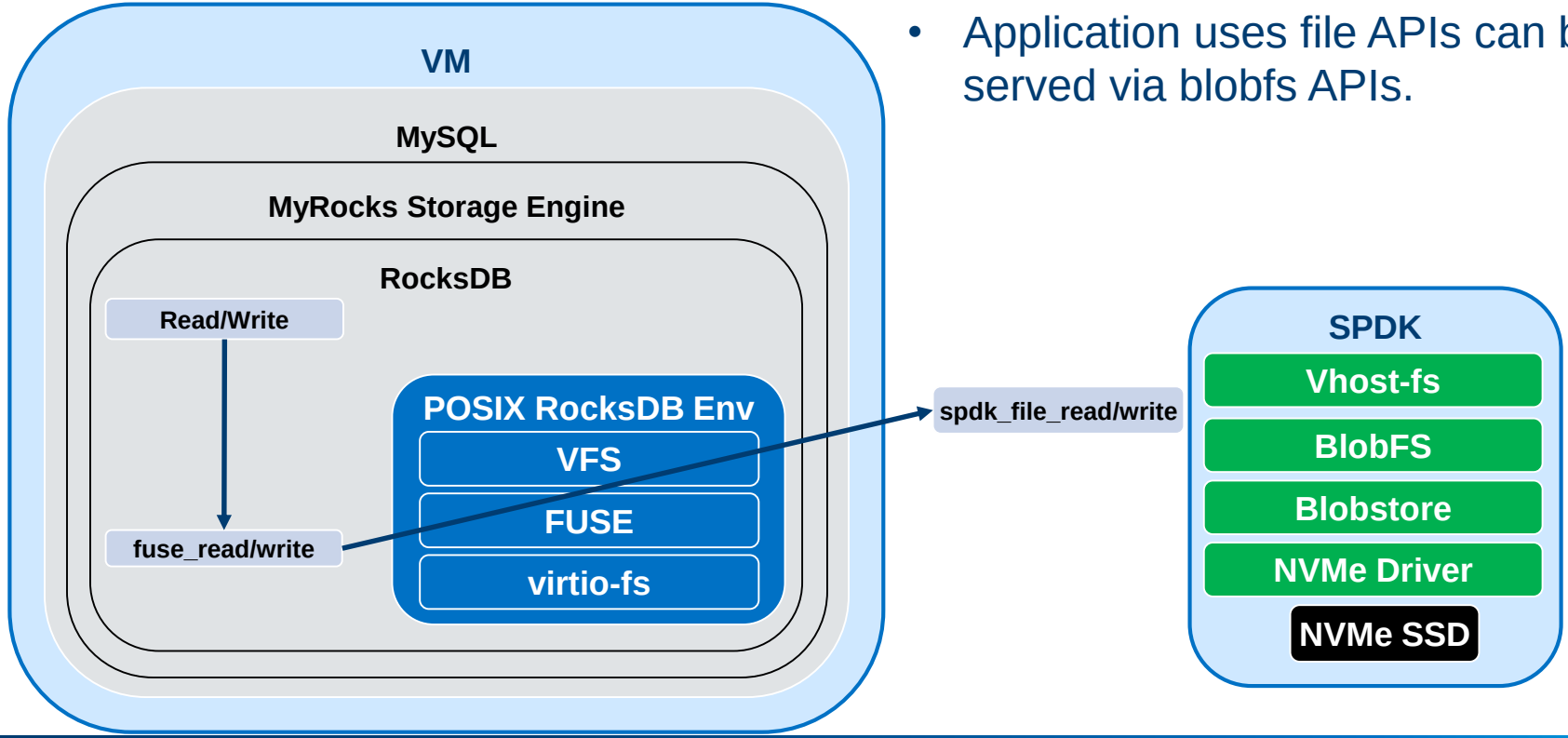
Open and Close Operations in FUSE and SPDK



Implementation Details with Read



Application Acceleration in VM

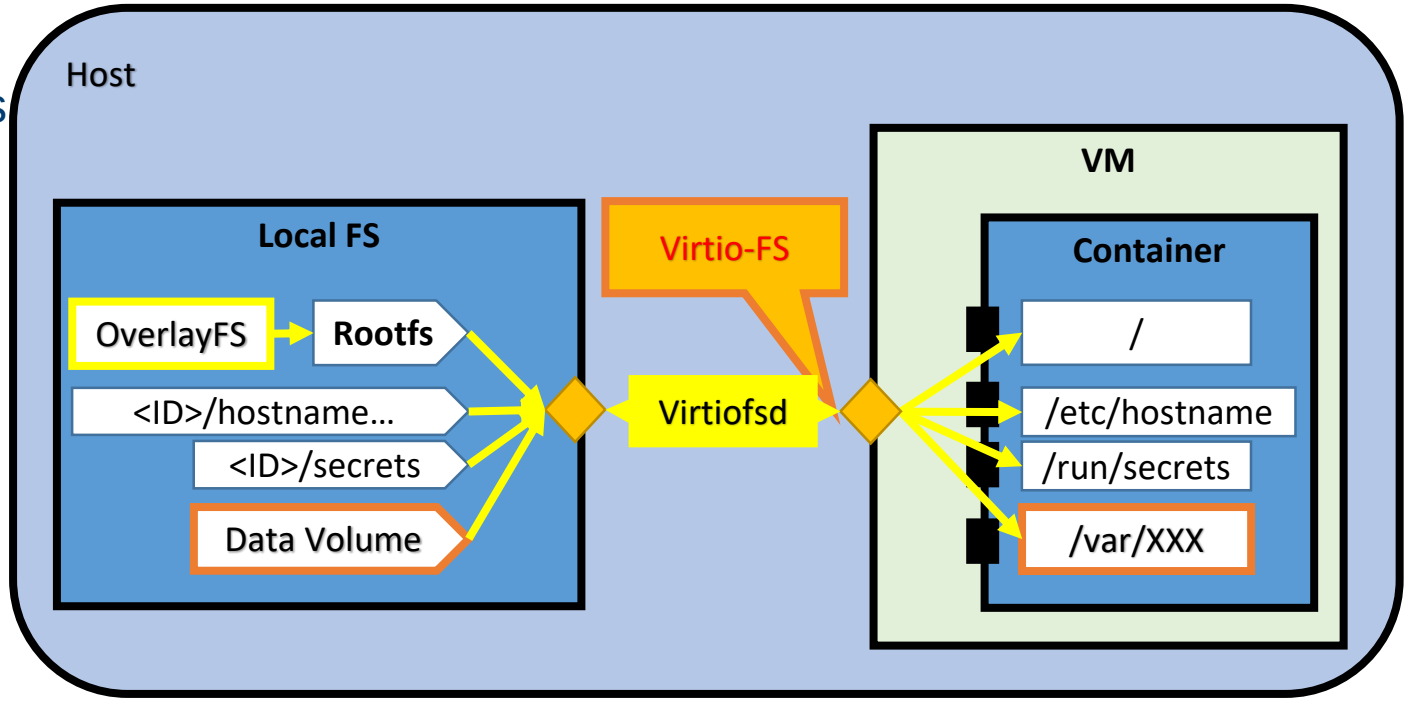


- Application uses file APIs can be served via blobfs APIs.

SPDK Vhost-fs with Kata Container

Virtio-fS in Kata Container Storage

- Offer local file system semantics and performance
- Virtiofsd daemon handles VM requests
- Virtiofsd daemon performs IO with file system calls

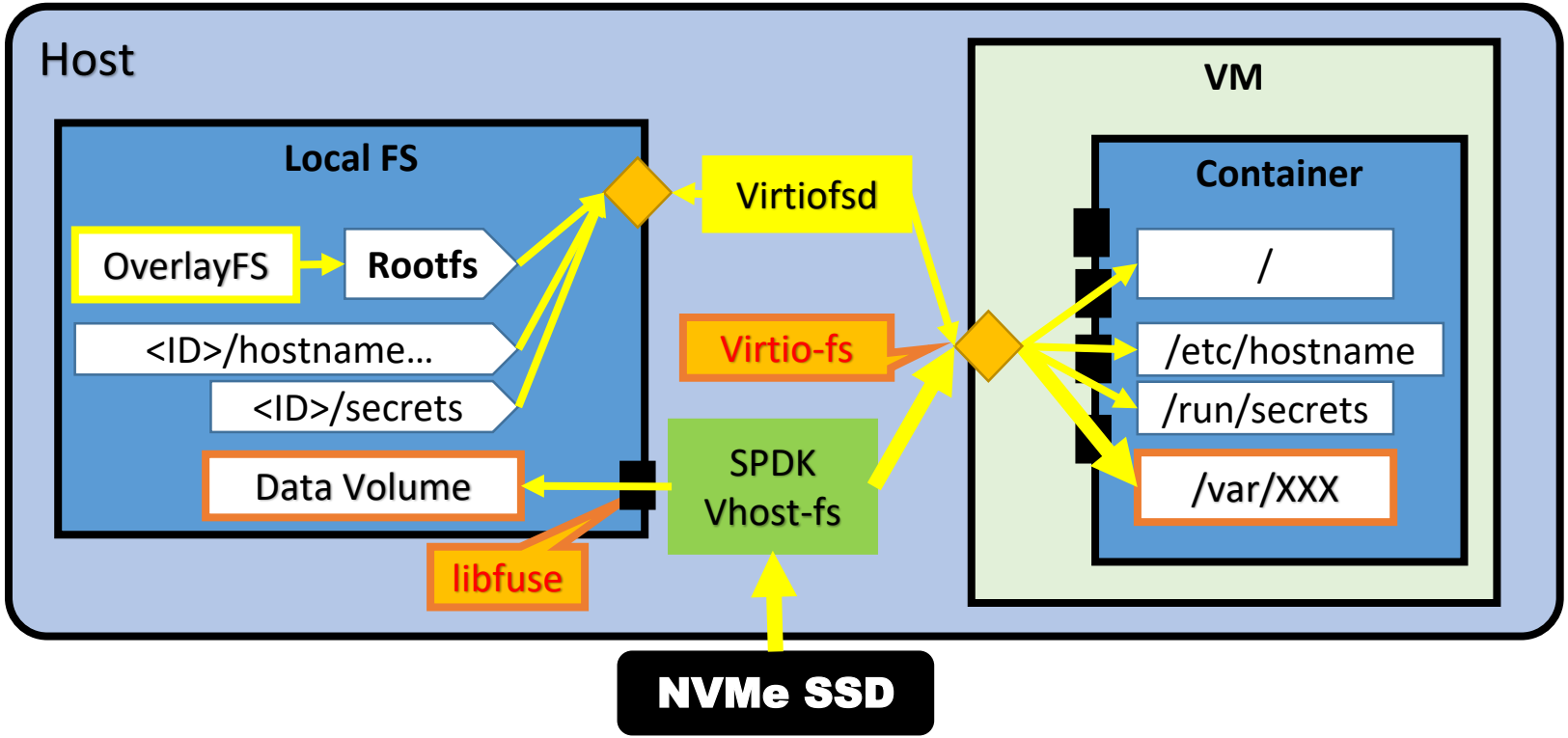


Kata-container

- The challenge when using with Kata-container
  - Shared file system is required for Kata-container
  - Overlay file system for container image
  - No directory view from Host side when using SPDK vhost-fs

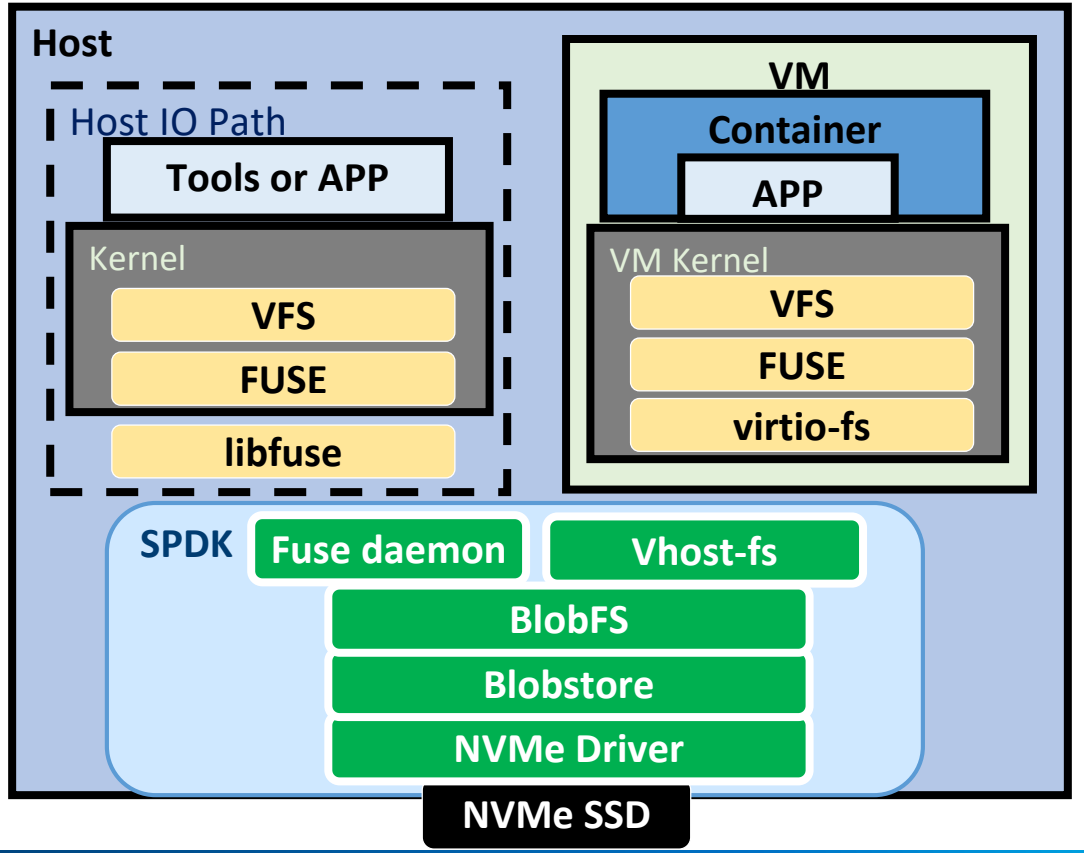
- How to use SPDK vhost-fs with Kata-container
  - Data volume can be used for shared data between different containers

SPDK vhost-fs in Kata Container Storage



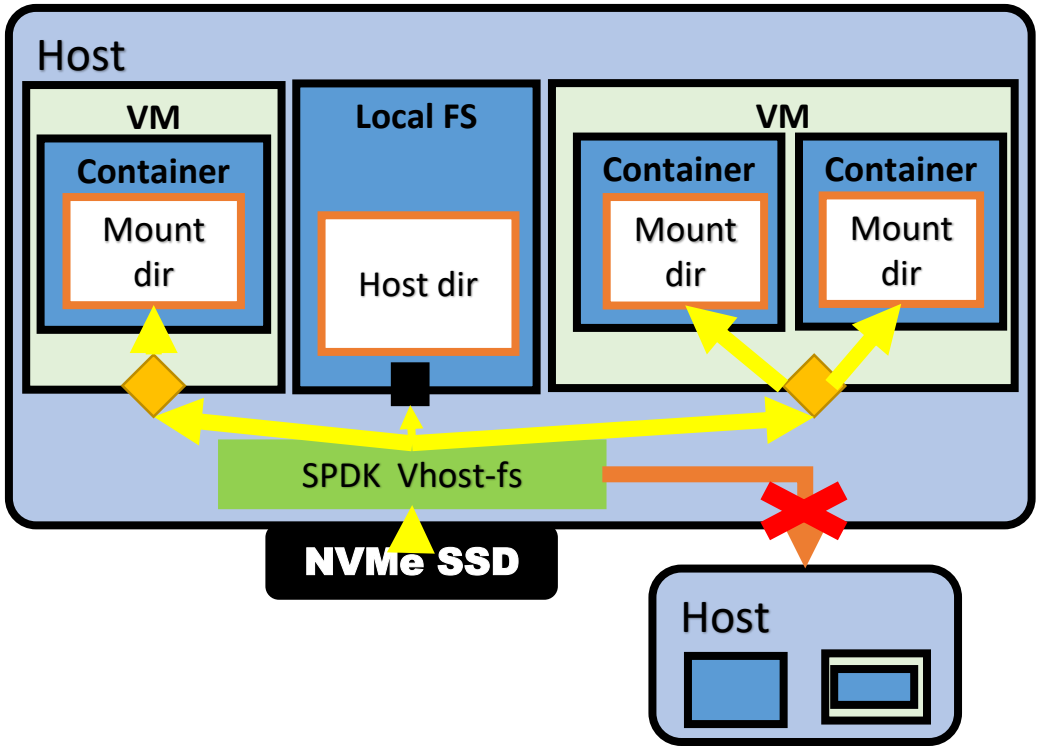
Software stack of vhost-fs for Kata container

- Vhost-fs for VM/container
- SPDK Fuse daemon for host



Summary for SPDK vhost-fs

- Sharing between Container and host
- Sharing between containers in different VM
- Sharing between containers in one VM
- How to sharing between containers in different host



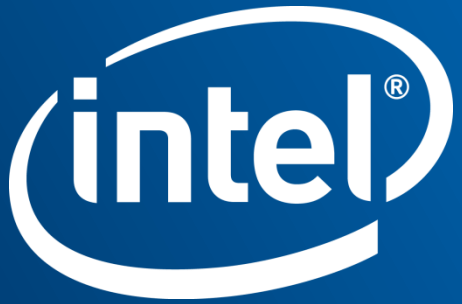
Future plans

- More generic features in user space file system

  -integrated with PMDK for metadata management

- Rootfs image used with the user space file system

  -snapshot support


intel®

Patch

- SPDK Vhost-fs:

SPDK: https://review.gerrithub.io/c/spdk/spdk/+/449162