KV SSD Host Software Stack

Changho Choi
changho.c@samsung.com

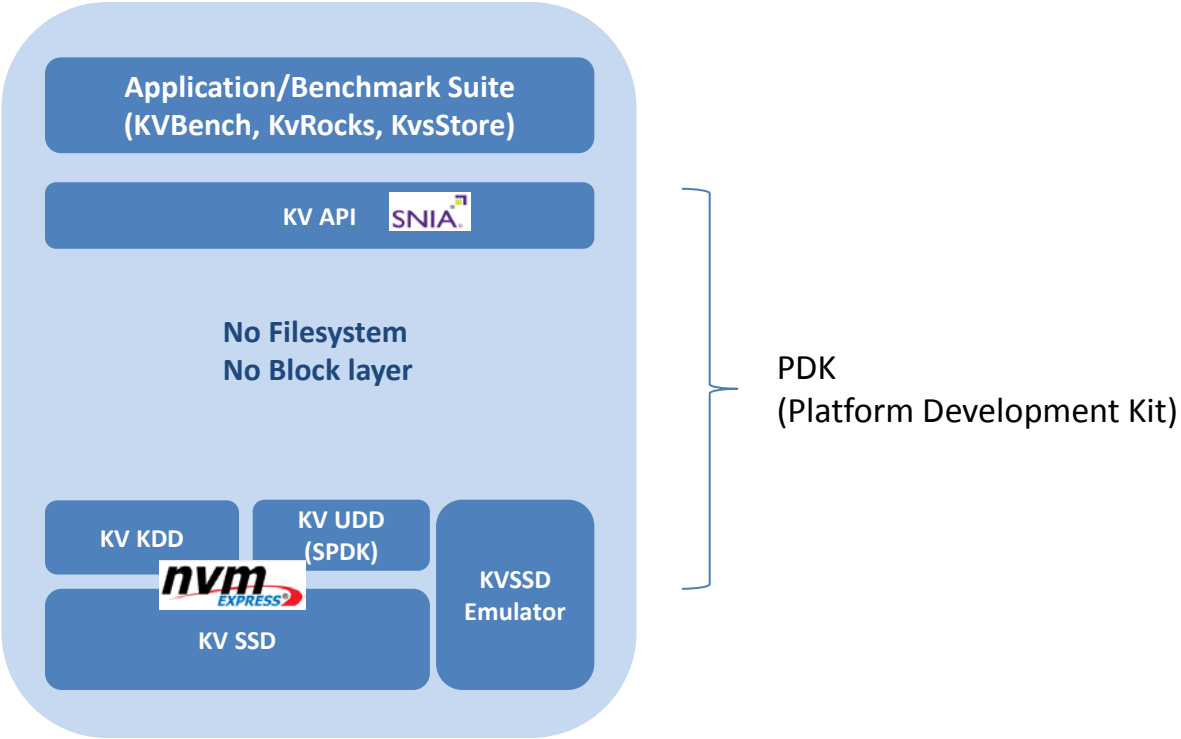
Device Solutions America

Agenda

- KV host software system architecture
- Kernel/User driver
- KV SSD emulator
- KV API (Application Programming Interface)
- Applications: KVBench
- Open source & github
- Standard progress
- System build and performance measurement process

COLLABORATE. INNOVATE. GROW.
SAMSUNG

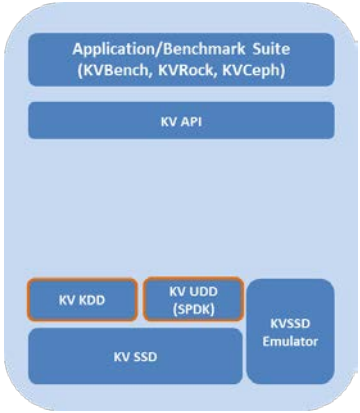
KV Host Software Stack



COLLABORATE. INNOVATE. GROW.
SAMSUNG

Device Driver

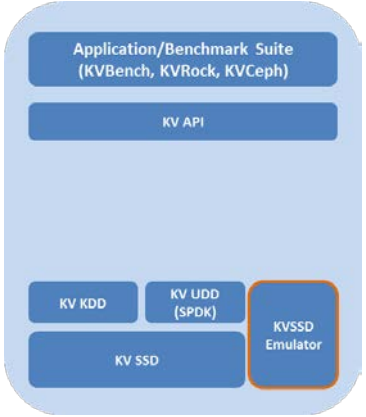
- **Implement NVMe KV commands with vendor specific opcodes**
- **NVMe KV command set standard discussion ongoing in NVMe TWG**
- **Kernel space driver**
  - Extend standard Linux Kernel driver by adding KV command support functions
  - Currently use ioctl and interrupt to communicate with KV SSD
  - IO queue management, IO scheduler, etc.
  - CentOS 7.2 Kernel v3.10
  - Ubutu 16.04
    - Kernel version v4.4, v4.9.5, v4.13, v4.15
- **User space driver**
  - Extend Intel SPDK by adding KV command support functions
- **Current open source software**
  - Command set: store/retrieve/delete/exist/iterator
  - Key is carried in NVMe command when key is smaller than or equal to 16-byte
  - Bigger key(>=16-byte) is delivered through PRP



COLLABORATE. INNOVATE. GROW.
SAMSUNG

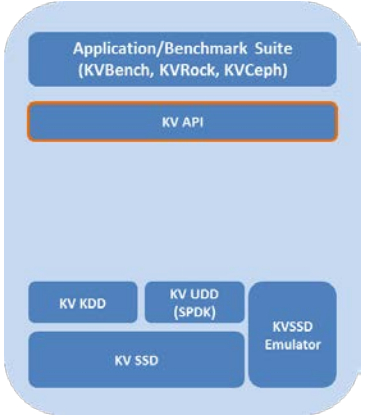
KV SSD Emulator

- **User space KV SSD emulator (no KV SSD device required)**
  - KV SSD emulator simulates KV SSD operations
  - Does not implement NVMe commands
- **Support async operation**
- **System setup operations**
  - Device initialization
  - Namespace setup (create/delete namespace, etc.)
  - Queue management (create/delete queues, etc.)
- **Key-value operations**
  - retrieve/store/delete for individual keys
  - exists: check key existence
  - iterator: key only or key-value pairs
    - Iterator group defined with MSB bit mask and bit pattern (up to 4-byte) in key



COLLABORATE. INNOVATE. GROW.
SAMSUNG

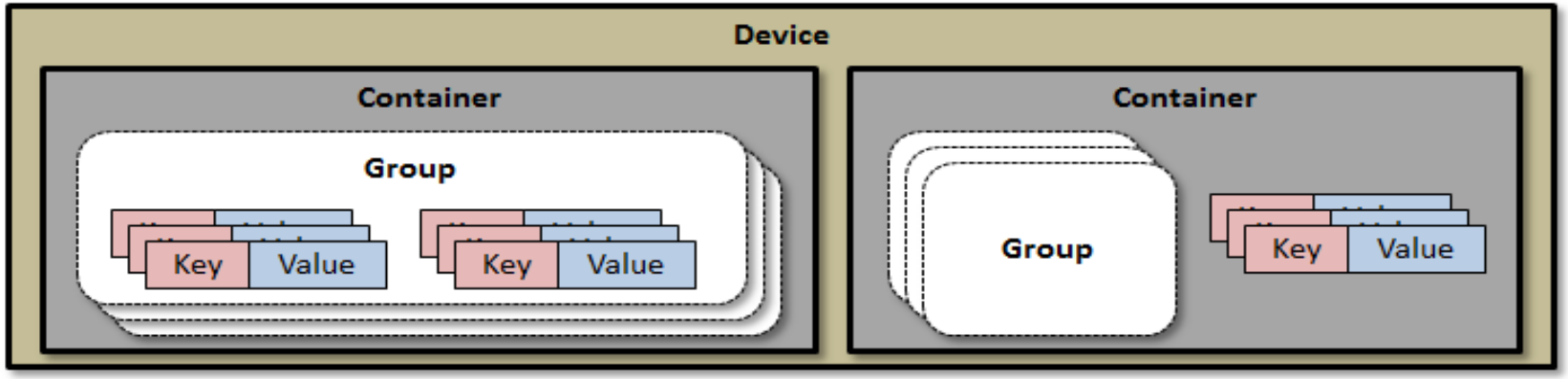
KV API (Application Programming Interface)



- **KV API**
  - KV API is a user space library that applications can utilize for system configuration and key-value operations
  - KV API supports both Kernel and user space drivers
  - Support sync and async operations
- **System configurations**
  - KV SSD device set up: open/close
  - User space device driver setup(e.g., SPDK): efficient memory management, etc.
  - Efficient host system setup: CPU core affinity, NUMA, etc.
- **Key value operations with options**
  - Basic KV API: store/retrieve/delete/exis/iterator
    - Sync/async mode command support
  - Retrieve device specific information (e.g., device utilization, etc.)

COLLABORATE. INNOVATE. GROW.
SAMSUNG

KV SSD API



- **Device**
- **Container (=key space):**
  - logical management unit like namespace in block device (e.g., nvme0n1)
- **Group**
  - logical set of key value tuples within a container which users can dynamically create (iterator, etc.)
- **Tuple: key value pair**

COLLABORATE. INNOVATE. GROW.
SAMSUNG

KV APIs – Device and Container

- **Device interface**
  - Kvs_open_device
  - Kvs_close_device
  - Kvs_get_device_info
  - Kvs_get_device_capacity
  - Kvs_get_device_utilization
  - Kvs_get_min_key_length
  - Kvs_get_max_key_length
  - Kvs_get_min_value_length
  - Kvs_get_max_value_length
  - Kvs_get_optimal_value_length
- **Container interface**
  - Kvs_create_container
  - Kvs_delete_container
  - Kvs_open_container
  - Kvs_close_container

COLLABORATE. INNOVATE. GROW.
SAMSUNG

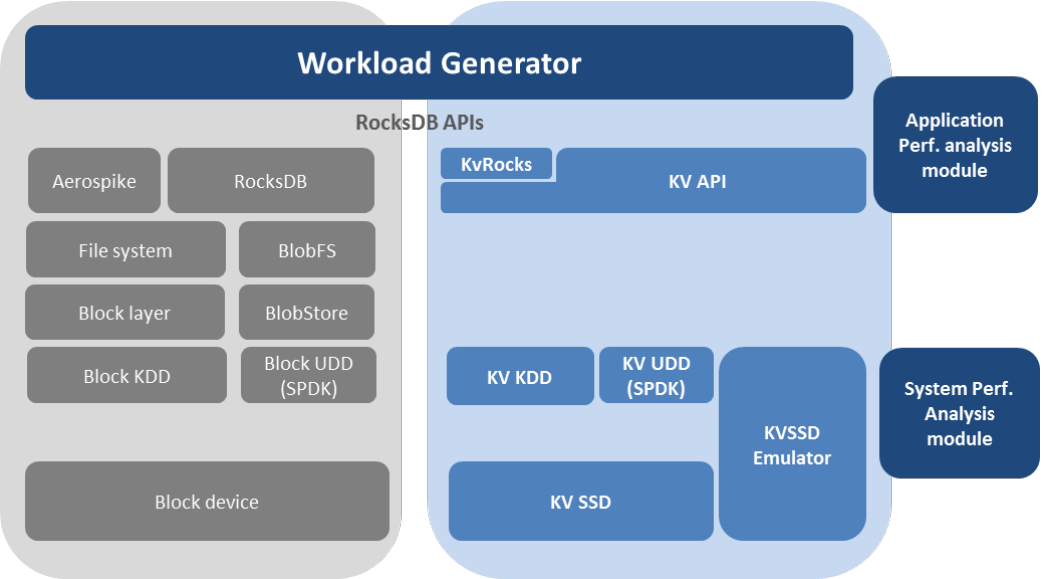
KV APIs – Tuple and Iterator

- **Key Value Tuple**
  - Kvs_get_tuple_info
  - Kvs_retrieve_tuple
  - Kvs_retrieve_tuple_async
  - Kvs_store_tuple
  - Kvs_store_tuple_async
  - Kvs_delete_tuple
  - Kvs_delete_tuple_async
  - Kvs_exist_tuples
  - Kvs_exist_tuples_async
- **Iterator**
  - Kvs_open_iterator
  - Kvs_close_iterator
  - Kvs_iterator_next
  - Kvs_iterator_next_async

COLLABORATE. INNOVATE. GROW.
SAMSUNG

kvbench: KV Benchmark Suite

- **Extended open source benchmark tool to support KV API**
- **Implemented additional workload and performance measurement features**



- **Workload generator**
  - Generate various workloads with different DB configurations
  - Generate different workloads directly to the KV SSD
  - Workload: insertion, mixed workload with uniform or Zipfian distribution, various key and value sizes, etc.
- **Application performance analysis module**
  - Reports application level stats, e.g. ops/sec, latency, etc.
- **System performance analysis module(separate module)**
  - Reports system level stats, e.g. CPU and memory utilization, etc.

COLLABORATE. INNOVATE. GROW.
SAMSUNG

KV SSD Host Software Open Source

- **KV SSD host software package is publicly released in github**
  - https://github.com/OpenMPDK/KVSSD
  - KV API, drivers, emulator, bench mark suite, etc.

COLLABORATE. INNOVATE. GROW.
SAMSUNG

KVSSD SDK github Architecture

- KVSSD
  - application
  - PDK
  - spec

- KVSSD
  - application
    - kvbench
      - include
      - lib
      - src
      - wrappers
  - PDK
    - cmake
    - core
    - driver
  - spec
    - nvme
    - snia

kvbench

- PDK
  - cmake
  - core
    - include
    - lib
    - sample_codes
    - src
      - api
    - device_abstract_layer
      - emulator
        - include
        - src
      - kernel_driver_adapter
        - include
        - src
    - driver
      - PCIe
        - kernel_driver
          - kernel_v3.10
          - kernel_v4.9.5
          - kernel_v4.13
          - kernel_v4.15
        - user_driver

API core library

emulator

Kernel driver adapter

Driver

Kernel driver

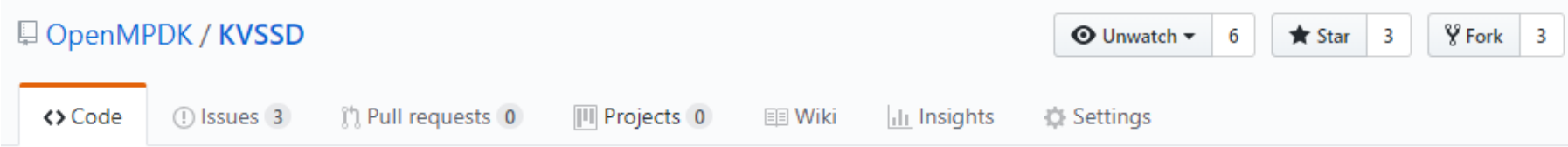
Link to uNVMe user driver

https://github.com/OpenMPDK/KVSSD

COLLABORATE. INNOVATE. GROW.

SAMSUNG

Open Source - GitHub Features



- **Code – code repository, README, etc.**
- **Issues – users report issues or ask questions**
- **Pull requests**
  - Inform others of changes developers pushed to a branch in a repository on GitHub. Once a pull request is opened, users can discuss and review the potential changes with collaborators and add follow-up commits.
  - The change would be merged to main branch as needed.
- **Wiki – FAQ, trouble shooting, public presentation materials, etc.**

Github - Issues

- **Use for bug report and responses**

OpenMPDK / KVSSD

Unwatch 6 | Star 3 | Fork 3

Code | Issues 2 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors discover issues labeled with help wanted or good first issue

Filters | is:issue is:open | Labels | Milestones | New issue

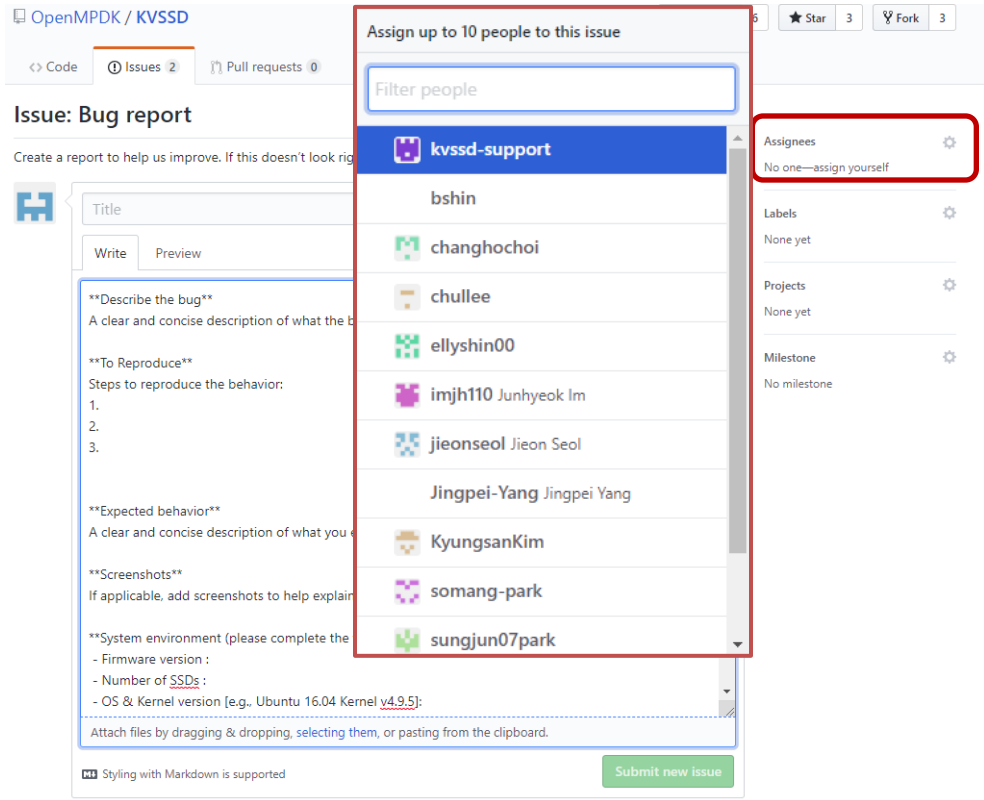
2 Open | 1 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

[emulator issue] kvs_retrieve_tuple value length
#3 opened 29 days ago by celeryfake

[thread safety issue] seg fault on multithread using sync api with kvs emulator
#2 opened on Sep 6 by celeryfake 1

ProTip! Find everything you created by searching author:changhochoi.

Issues Report – Template & Assignee



**** Location (Korea, USA, China, India, etc.) ****
Put your location to get prompt support

****Describe the bug****
A clear and concise description of what the bug is.

****To Reproduce****
Steps to reproduce the behavior:
1.
2.
3.

****Expected behavior****
A clear and concise description of what you expected to happen.

****Screenshots****
If applicable, add screenshots to help explain your problem.

****System environment (please complete the following information)****
- **Firmware version :**
- **Number of SSDs :**
- **OS & Kernel version [e.g., Ubuntu 16.04 Kernel v4.9.5]:**
- **GCC version [e.g., gcc v5.0.0] :**
- **kvbench version if kvbench runs [e.g., v0.6.0]:**
- **KV API version [e.g., v0.6.0]**
- **User driver version :**
- **Driver [Kernel or user driver or emulator] :**

****Workload****
- **number of records or data size**
- **Workload(insert, mixed workload, etc.) [e.g., sequential or random insert, or 50% Read & 50% write]**
- **key size :**
- **value size :**
- **operation option if available [e.g., sync or async mode] :**

****Additional context****
Add any other context about the problem here.

Issues Report – Labels

OpenMPDK / KVSSD

Watch 6 | Star 3 | Fork 3

Code | Issues 2 | Pull requests 0 | Wiki | Insights | Settings

Issue: Bug report

Create a report to help us improve. If this doesn't look right, c

Title

Write | Preview

Describe the bug
A clear and concise description of what the bug i

To Reproduce
Steps to reproduce the behavior:
1.
2.
3.

Expected behavior
A clear and concise description of what you expe

Screenshots
If applicable, add screenshots to help explain you

**System environment (please complete the follo
- Firmware version :
- Number of SSDs :
- OS & Kernel version [e.g., Ubuntu 16.04 Kernel

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Submit new issue

Apply labels to this issue

Filter or create labels

- bug — Something isn't working
- duplicate — This issue or pull request already exists
- enhancement — New feature or request
- help wanted — Extra attention is needed
- invalid — This doesn't seem right
- question — Further information is requested
- wontfix — This will not be worked on

Edit labels

Assignees
No one—assign yourself

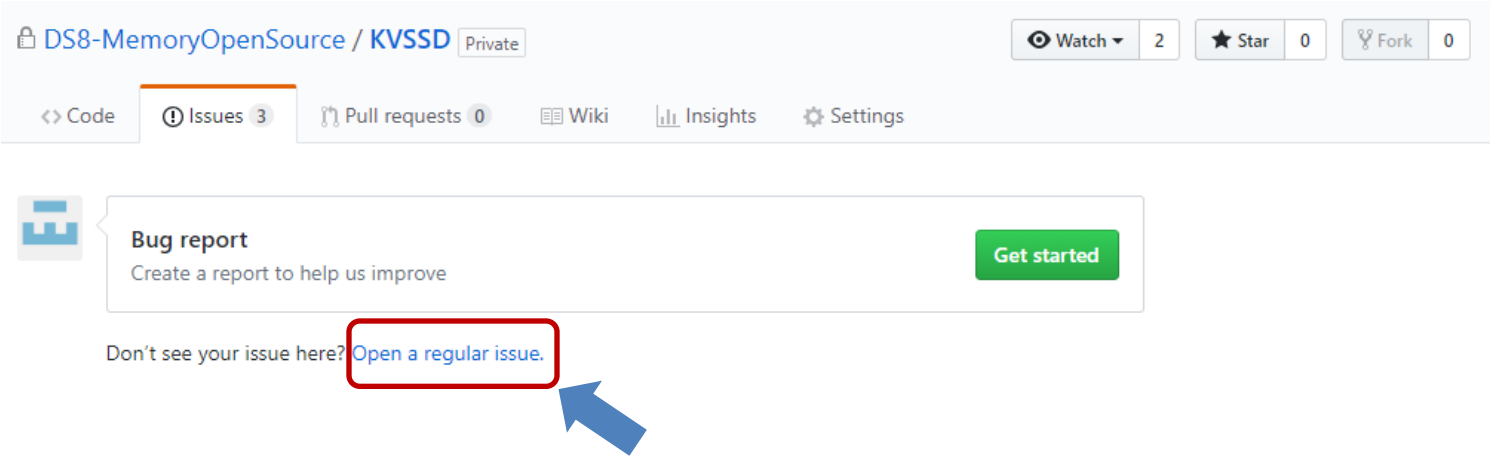
Labels
None yet

Projects
None yet

Milestone
No milestone

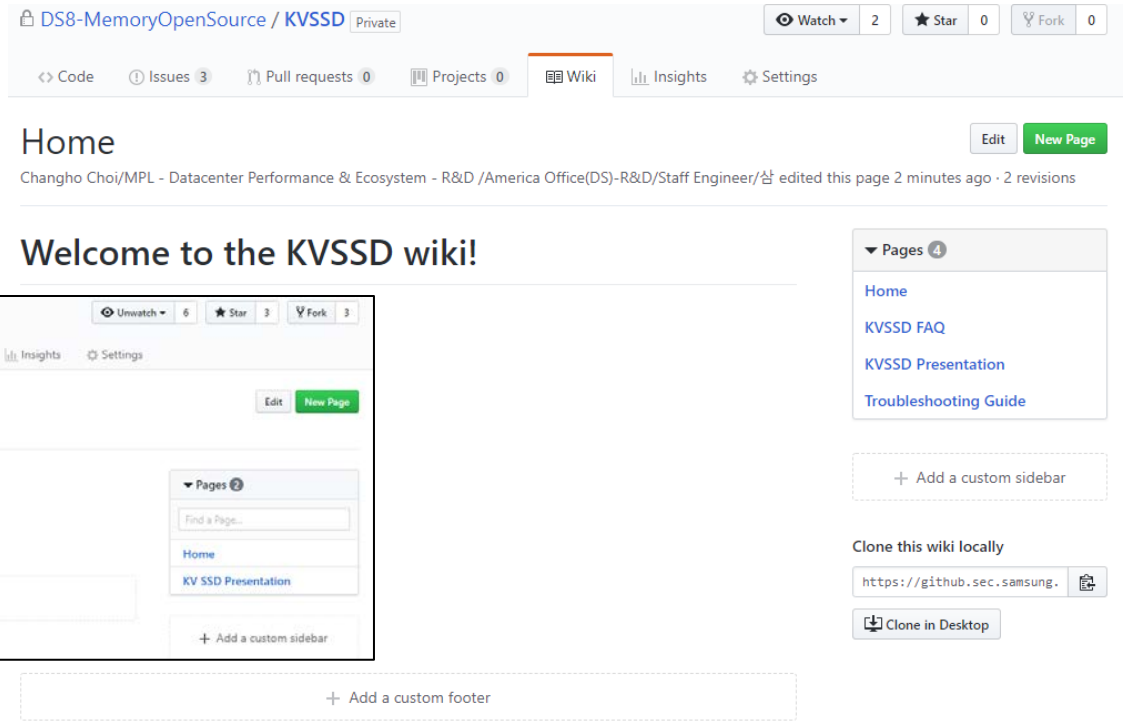
Issues – Regular Issue for Generic Questions

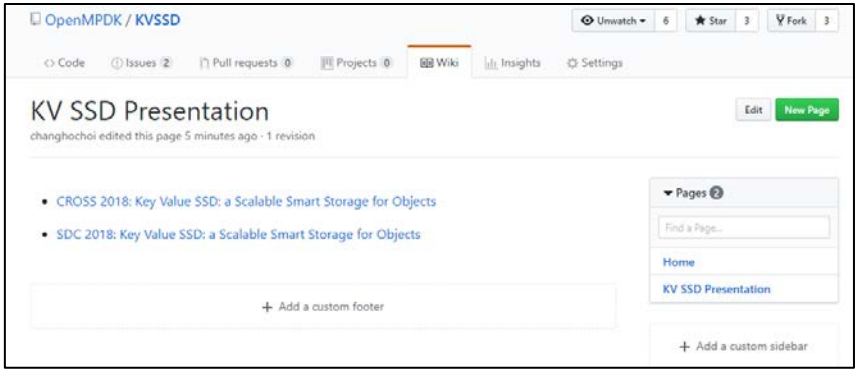
- **Open a regular issue instead of regular bug report when you have generic questions**



Github - Wiki

- KV SSD introduction
- FAQ, trouble shooting, etc.
- Public materials: whitepaper, presentation materials, etc.





Wiki - FAQ

DS8-MemoryOpenSource / KVSSD Private

Watch 2 | Star 0 | Fork 0

Code | Issues 3 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

KVSSD FAQ

Edit | New Page

Changho Choi/MPL - Datacenter Performance & Ecosystem - R&D /America Office(DS)-R&D/Staff Engineer/삼 edited this page a minute ago · 8 revisions

OPERATING SYSTEM & FIRMWARE

- What Linux OS and kernel version is supported?
  - CentOS 7.5 Kernel v 3.10
  - Ubuntu 16.04 Kernel v4.4.0 (distribution code only not Vanilla version) v4.9.5, v4.13, v4.15
- What is the supported firmware version on the KV SSD?

  All publicly released KV SSD firmware
- What is the supported uDD version?

  TBD

KEY VALUE DEVICE (KV DEVICE)

- Is multiple Key Space supported?

  Yes, device supports up to two Key Spaces. The current API support only one Key Space (keyspace id = 0). The multiple Key Space will be supported in the future release version.
- Does KV device also support block command?

  No, except for secure erase and smart log (KV SSD supports only KV interface)
- Is there dual port support?

  No

Pages 3

- Home
- KVSSD FAQ
- KVSSD Presentation

+ Add a custom sidebar

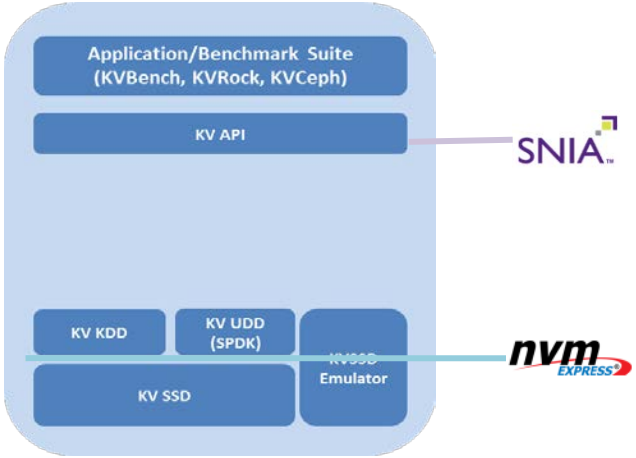
Clone this wiki locally

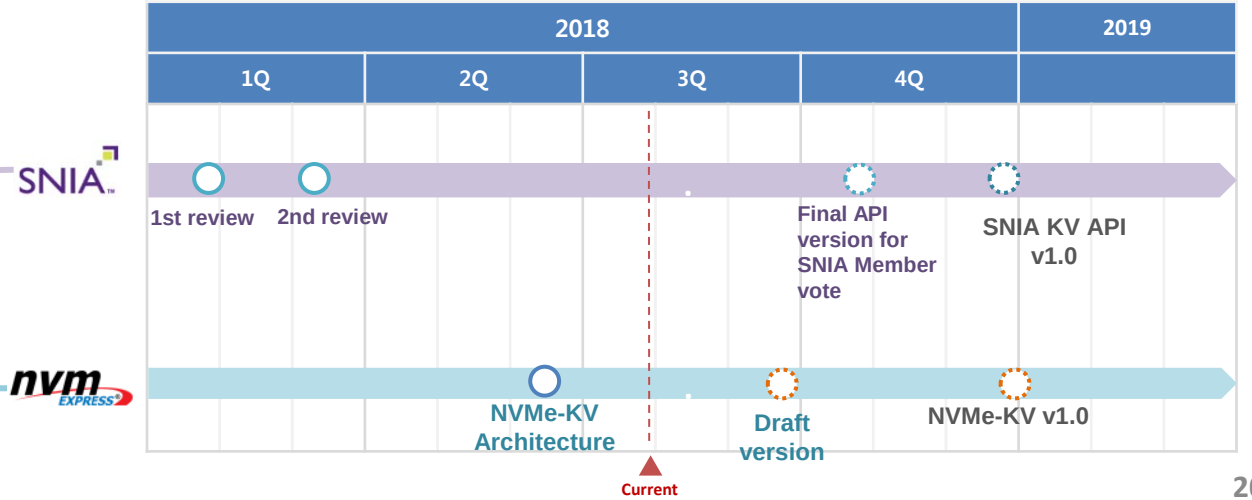
https://github.sec.samsung.

Clone in Desktop

KV SSD Standards

- **Samsung defines a KV architecture/command set based on NVMe spec.**
  - NVMe KV standard draft review meetings ongoing
- **KV API standard draft discussion ongoing**
  - KV APIs include store/retrieve/delete/exist/iterator





COLLABORATE. INNOVATE. GROW.
SAMSUNG

System build and performance measurement

System Build

1. Download software package

git clone https://github.com/OpenMPDK/KVSSD.git

2. Build and install device driver

```
cd KVSSD/PDK/driver/PCIe/kernel_driver/kernel_v4.9.5/
make all
sudo ./re_insmod.sh
```

3. Build KV API library

```
cd /KVSSD/PDK/core
mkdir build && cd build
cmake -DWITH_KDD=ON ../
make -j24
```

4. Test sample codes

```
sudo ./sample_code_async -d /dev/nvme0n1 -n 100 -q 64 -o 1 -k 16 -v 4096
```


COLLABORATE. INNOVATE. GROW.
SAMSUNG

System Build(Cont'd)

5. Build kvbench benchmark tool for KV SSD

(https://github.com/OpenMPDK/KVSSD/tree/master/application/kvbench)

```
cd KVSSD/application/kvbench
mkdir build_kv && cd build_kv
cmake -DCMAKE_INCLUDE_PATH=/KVSSD/PDK/core/include
        -DCMAKE_LIBRARY_PATH=/KVSSD/PDK/core/build ../
make kv_bench
```


COLLABORATE. INNOVATE. GROW.
SAMSUNG

Run Insertion & Mixed Workload (R50U50)

1. **create & modify cpu config file**

```
cd kvbench/build_kv
LD_LIBRARY_PATH=/KVSSD/PDK/core/build ./kv_bench -c
# This will generate default cpu.txt file
# Modify cpu.txt for (nodeid,coreid,deviceid) mapping if needed
```

2. **modify bench_config.ini for workloads**

```
ndocs=100000
device_path = /dev/nvme0n1
read_write_insert_delete = 50:50:0:0
```

3. **run benchmark**

```
sudo LD_LIBRARY_PATH=/KVSSD/PDK/core/build ./kv_bench -f bench_config.ini
```


COLLABORATE. INNOVATE. GROW.
SAMSUNG

kvbench Configuration (bench_config.ini)

[document]

- ndocs = 100 # number of records, insert 100 key-value pairs during load phase

[system]

- key_pool_unit=16 # size of unit of key in key memory pool in bytes. Should be same as key_length or maximum key length if various key size is used
- Value_pool_unit=4096 # size of unit of value in value memory pool in bytes. Should be same as value_length or maximum value length if various value size is used
- device_path=/dev/nvme0n1 # device path under /dev directory. It is used for cpu core and numa assignment.

[kvs]

- device_path=/dev/nvme0n1 # it should be same as device_path in [system] section for kDD or emulator. It would be different from device_path for uDD (e.g., 0000:06:00.0)

[population]

- seq_fill=true # sequential insertion; false means random insertion

[key_length]

- distribution=fixed # fixed, uniform, normal

[value_length]

- distribution=fixed # fixed, uniform, normal, ratio

[operation]

- Duration = 600 # benchmark duration in seconds of insertion or
- nops = 1000000 # number of operation after insertion
- Read_write_insert_delete=50:50:0:0 # operation type ratio for read/write/insert/delete. If insert is larger than 0, only nops must be used
- Batch_distribution=uniform # key space distribution (uniform, zipfian)

COLLABORATE. INNOVATE. GROW.

SAMSUNG

Compile Results

- **Performance measurement results are in ./logs directory**
  - **KVS-insert.latency.csv**: latency of insertion measured in sampling rate defined in [latency_monitor] section in bench_config.ini file. (sampling rate is in the unit of Hertz)
  - **KVS-insert.ops.csv**: operation per second of insertion operation measured in print_term_ms defined in bench_config.ini.
    - Time, iops (=average ops), iops_i(=tailing/instant ops), counter (=operation count)
  - **KVS-ops.txt**: summary of performance measurement
    - Total run time, ops, avg latency & latency distribution, total read/write count, etc.
  - **KVS-run.latency.csv**: latency of operations for performance measurement
    - pos(=position/index), write, read, delete
  - **KVS-run.ops.csv**: operation per second during performance measurement
    - time, ops_avg, ops_i, read_cnt, write_cnt, bytes_written

COLLABORATE. INNOVATE. GROW.
SAMSUNG

Thank You!