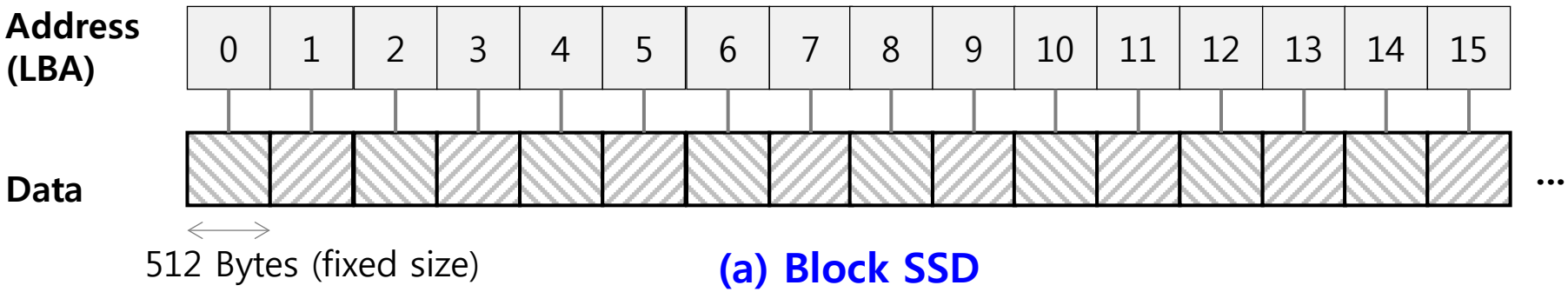
KV-SSD Seminar

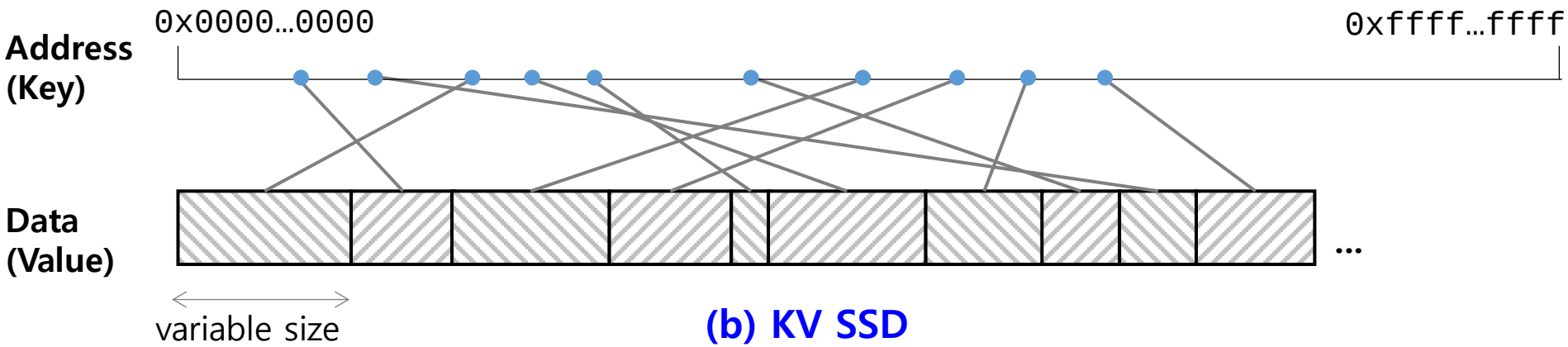
Sang-yoon Oh
S/W Development Team
Memory Division, Samsung Electronics

Why KV SSD?

What is Key Value SSD?

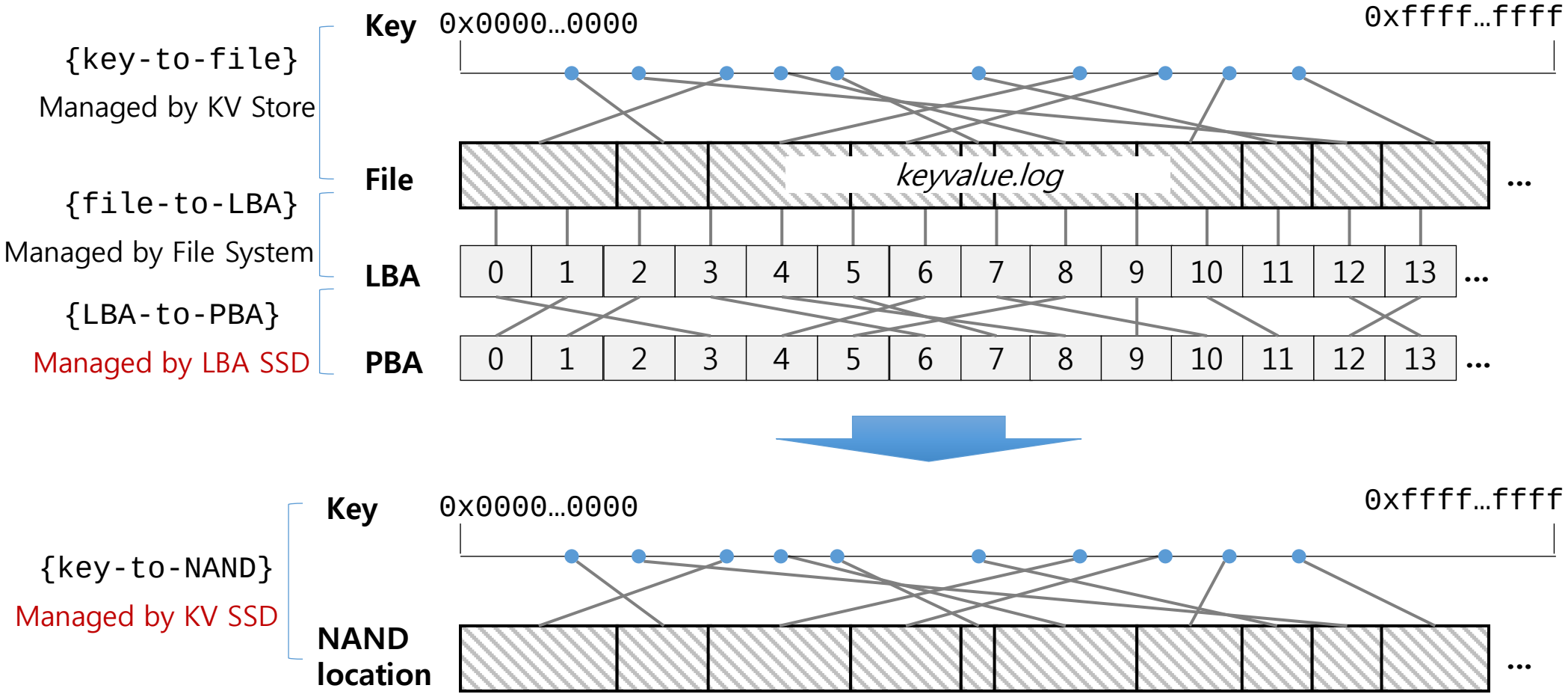
- Supports Key-Value Interface composed of Key (variable address) & Value (variable length)
  - Write: put(key, value); Read: value = get(key)
- Unstructured Data (Photo, Video, Document, etc.) is mostly expressed in Key-Value Pair
  → Can be directly stored into KV SSD w/o any conversion





Why KV SSD? – Simpler S/W Stack

- **Simpler Mapping → WAF ↓, Host S/W Complexity ↓, SSD Capacity Per Node ↑**
- **Unnecessary Block layer can be removed in certain applications**



Samsung KV SSD in detail

Samsung KV SSD Features

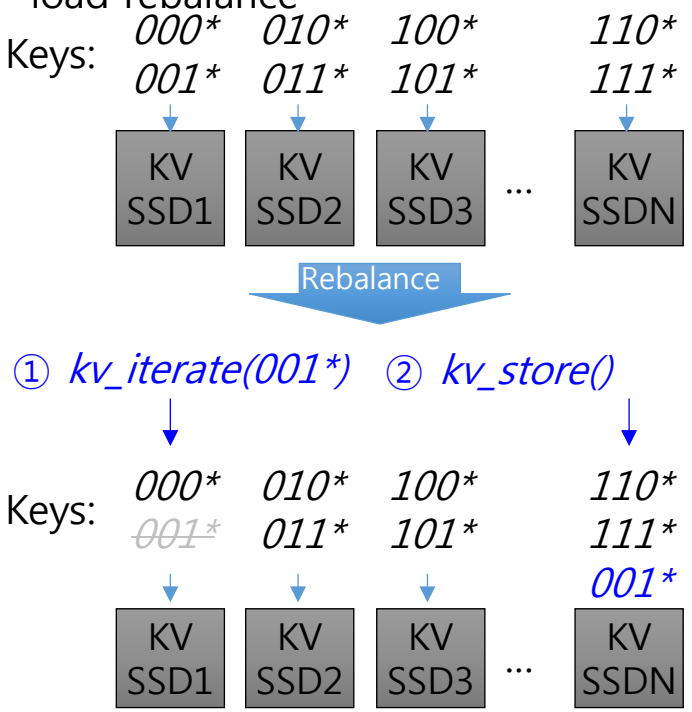
Item	Description
Key Size	4 ~ 255 Bytes
Value Size	0B~2MB (*small value less than 1KB is padded up to 1KB internally)
Value Size Alignment	1 Byte
KV command	Store, Retrieve, Delete, Exist, Iterate
Iterate Command	Key only iterate Key iterate with delete
Host buffer size in key-only iterate	Should be 32kB
Multi Key Space	Supported (2 KSID, fixed)
Write Atomicity	Supported
SED	Supported (TCG Opal single user mode)

Iterate Use Case

Iterate API: returns keys starting w/ specified Prefix (4Byte)

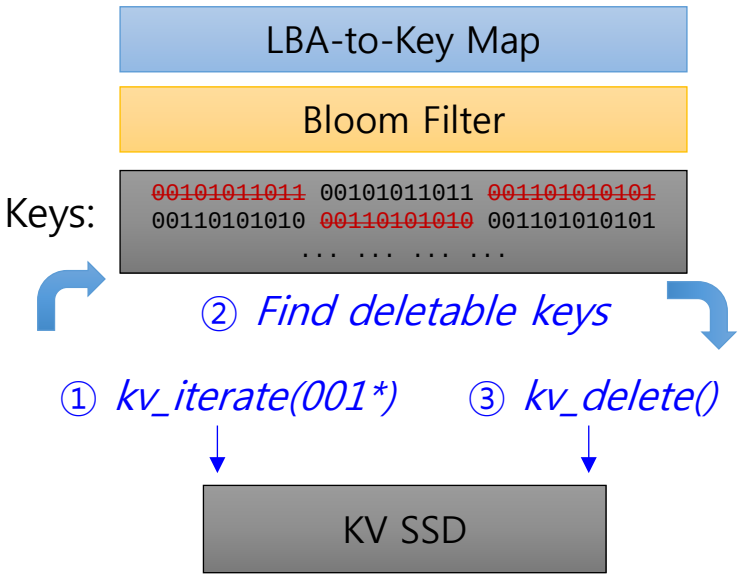
Use Case #1: Load Rebalance

- KV request is located based on Key prefix map → KV prefix map is updated during load rebalance



Use Case #2: Key Compaction

- Find deletable keys & Delete
- Part of keys may be loaded into DRAM to find deletable keys iteratively due to limited Host DRAM space

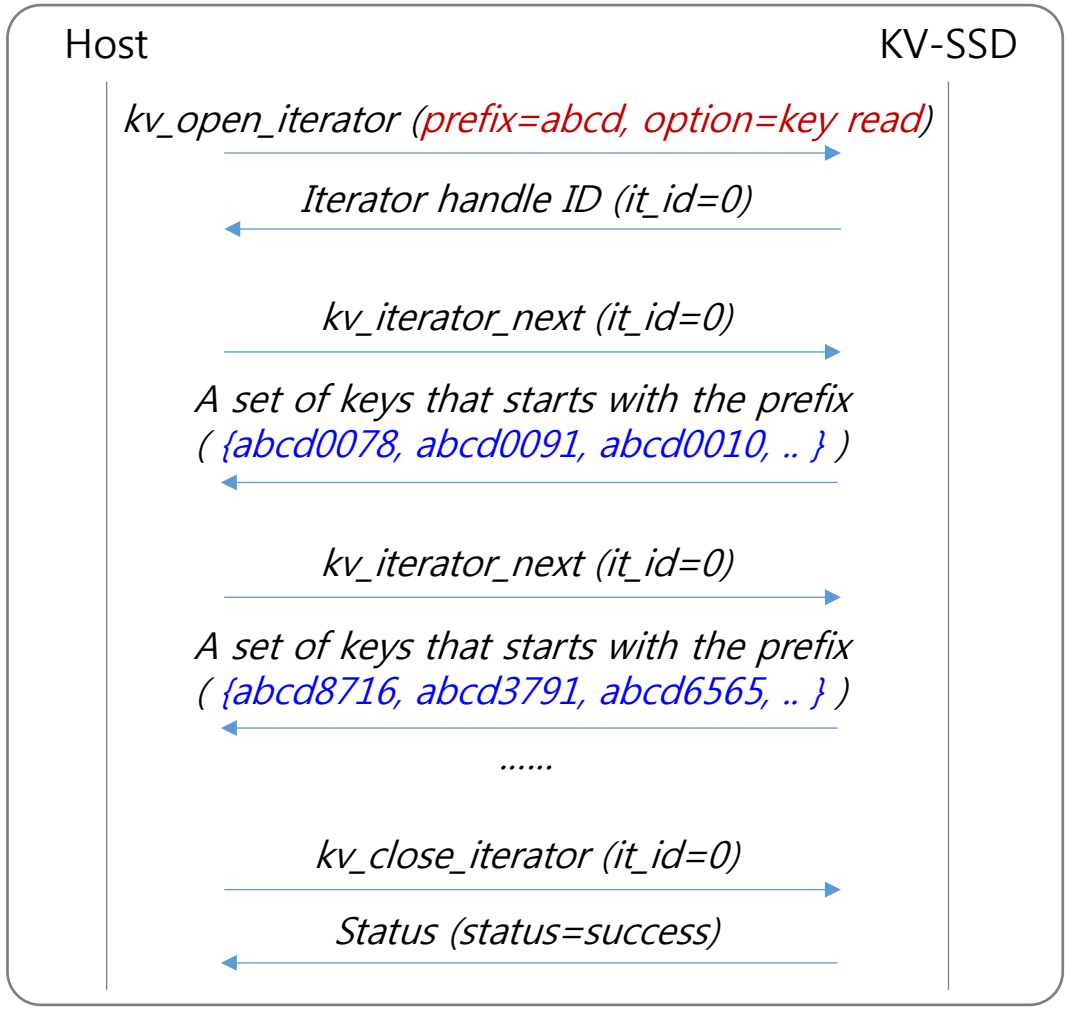


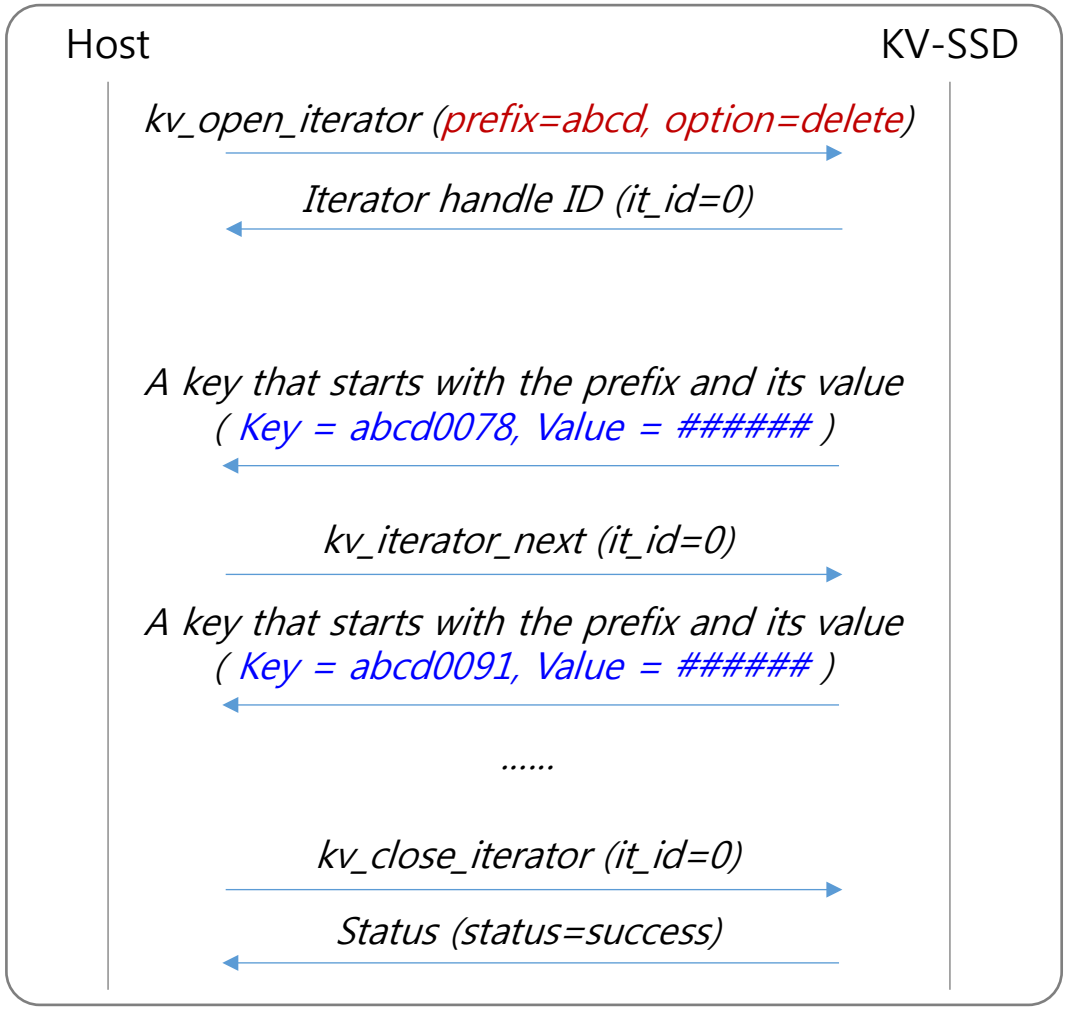
Iterate APIs

- *kv_open_iterator*
  : returns iterate handle
- *kv_close_iterator*
  : close handle
- *kv_iterator_next*
  : returns keys

Iterate API

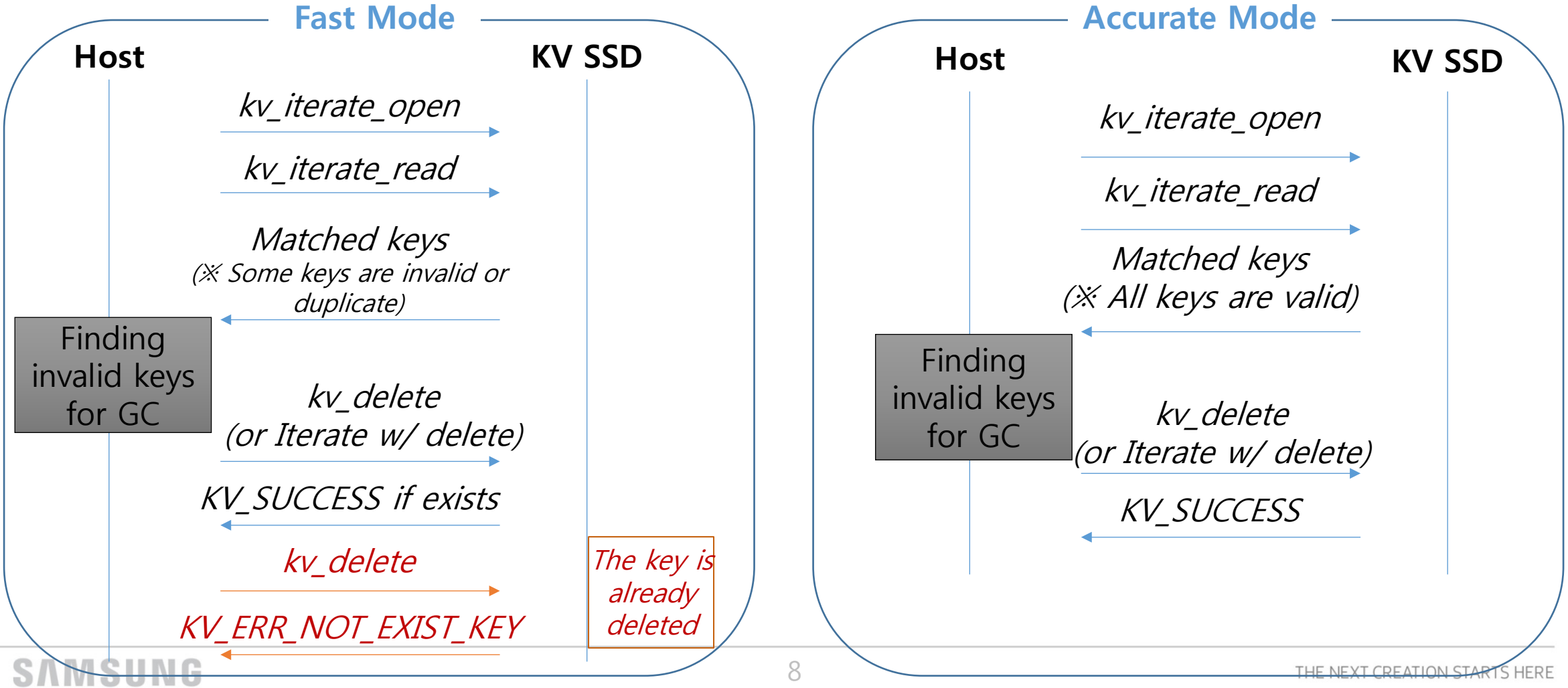
- **Iterator allows applications to read or delete a certain group of keys that matches with given condition**





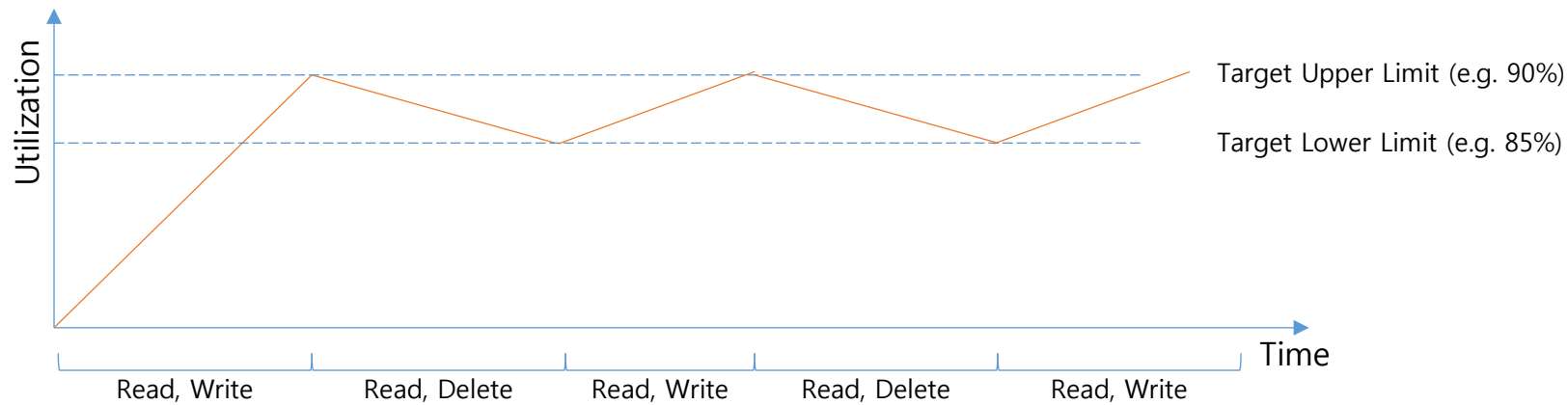
Iterate Mode

- **Current KV SSD supports Fast mode (contains false positive keys)**
  - Accurate mode will be enabled in future release



Workload Management Required

1) **Determine Target Utilization**
2) **Trigger Host compaction to find deletable keys when utilization exceeds the target**
3) **Issue Delete command or Iterate w/ delete**



KV SSD Command Format

KV SSD Command Set

KV I/F extends NVMe Standard (KV standardization in progress)

1. Store: write a {key, value} pair into KV SSD
2. Retrieve: read a {key, value} pair from KV SSD
3. Delete: delete a {key, value} pair from KV SSD
4. Exist: check if a specified key exists
5. Iterate: return keys w/ specified Prefix or delete them

※ New opcodes for KV Commands is defined

Bit	Description
31:16	Command Identifier (CID)
15:14	PRP or SGL for Data Transfer (PSDT)
13:10	Reserved
09:08	Fused Operation (FUSE)
07:00	**Opcode**

NVMe Standard

Bytes	Description
63:60	Command specific DWORD 15
59:56	Command specific DWORD 14
55:52	Command specific DWORD 13
51:48	Command specific DWORD 12
47:40	Command specific DWORD 11/10
39:24	Data Pointer (PRP1/2)
23:16	Metadata Pointer (MPTR)
15:08	Not used
07:04	Namespace Identifier (NSID)
03:00	Command Dword 0 (CDW0)

KV SSD Extension (ex. Store)

Bytes	Description
63:60	Value size in bytes
56:59	Key length (0~255) : key length – 1
55:52	key[15:12]
51:48	key[11:8]
47:40	key[7:0] or key PRP
39:24	Value PRP1/2
23:16	[31:2] The offset of value in bytes, [1:0] Option
15:08	Not used
07:04	Namespace Identifier (NSID)
03:00	Command Dword 0 (CDW0)

KV Command Format : Store

- **Store option : 0x00 (Default), 0x01 (Compression), 0x02 (Idempotent)**
- **Response: Status Code = IdempotentStoreFail (if the same key exists already)**

Bytes			Description	
			key size ≤ 16B	Key size > 16B
63:56	CDW14-15	8 Bytes	key[8:15]	Key PRP2
55:48	CDW12-13	8 Bytes	key[0:7]	Key PRP1
47:44	CDW11	4 Bytes	[18:31] Reserved, [16:17] Number of invalid bytes, [8:15] Store option, [0:7] Key length (0's based)	
43:40	CDW10	4 Bytes	Value size in DWORD	
39:24	DPTR	16 Bytes	Value PRP1/2	
23:20	MPTR	4 Bytes	Not used	
19:16		4 Bytes	Not used (It would be used for offset later)	
15:08	Reserved	8 Bytes	Not used	
07:04	NSID	4 Bytes	[8:31] Reserved [0:7] Keyspace ID	

Bit	Description	
31:16	Command Identifier (CID)	As specified in the NVMe 1.2 specification
15:14	PRP or SGL for Data Transfer (PSDT)	As specified in the NVMe 1.2 specification
13:10	Reserved	Not used
09:08	Fused Operation (FUSE)	Always 00b, Normal operation
07:00	Opcode	0x81

KV Command Format : Retrieve

- **Assumption: Host may not know the actual value size of the key to retrieve**
- **Host buffer size < actual value size : value is filled up to buffer size (actual value size is specified in CQ entry)**

Bytes			Description	
			key size ≤ 16B	Key size > 16B
63:56	CDW14-15	8 Bytes	key[8:15]	Key PRP2
55:48	CDW12-13	8 Bytes	key[0:7]	Key PRP1
47:44	CDW11	4 Bytes	[16:31] Reserved, [8:15] Retrieve option, [0:7] Key length (0's based)	
43:40	CDW10	4 Bytes	Host buffer size in DWORD	
39:24	DPTR	16 Bytes	Value PRP1/2	
23:20	MPTR	4 Bytes	Not used	
19:16		4 Bytes	Not used (It would be used for offset later)	
15:08	Reserved	8 Bytes	Not used	
07:04	NSID	4 Bytes	[8:31] Reserved [0:7] Keyspace ID	

Bit	Description	
31:16	Command Identifier (CID)	As specified in the NVMe 1.2 specification
15:14	PRP or SGL for Data Transfer (PSDT)	As specified in the NVMe 1.2 specification
13:10	Reserved	Not used
09:08	Fused Operation (FUSE)	Always 00b, Normal operation
07:00	Opcode	0x90

KV Command Format : Delete

- **Delete option : 0x00 (Default), 0x02 (Forced_Delete)**
- **Status Code : NotExistKey is set if key does not exist when Forced_Delete option is set**

Bytes			Description	
			key size ≤ 16B	Key size > 16B
63:56	CDW14-15	8 Bytes	key[8:15]	Key PRP2
55:48	CDW12-13	8 Bytes	key[0:7]	Key PRP1
47:44	CDW11	4 Bytes	[16:31] Reserved, [8:15] Delete option, [0:7] Key length (0's based)	
43:40	CDW10	4 Bytes	0x0 (should be zero)	
39:24	DPTR	16 Bytes	Not used	
23:20	MPTR	4 Bytes	Not used	
19:16		4 Bytes	Not used	
15:08	Reserved	8 Bytes	Not used	
07:04	NSID	4 Bytes	[8:31] Reserved [0:7] Keyspace ID	

Bit	Description	
31:16	Command Identifier (CID)	As specified in the NVMe 1.2 specification
15:14	PRP or SGL for Data Transfer (PSDT)	As specified in the NVMe 1.2 specification
13:10	Reserved	Not used
09:08	Fused Operation (FUSE)	Always 00b, Normal operation
07:00	Opcode	0xA1

KV Command Format : Exist

- **Check if key exists or not (No value part is returned)**

Bytes			Description	
			key size ≤ 16B	Key size > 16B
63:56	CDW14-15	8 Bytes	key[8:15]	Key PRP2
55:48	CDW12-13	8 Bytes	key[0:7]	Key PRP1
47:44	CDW11	4 Bytes	[16:31] Reserved, [8:15] Exist option, [0:7] Key length (0's based)	
43:40	CDW10	4 Bytes	0x0 (should be zero)	
39:24	DPTR	16 Bytes	Not used	
23:20	MPTR	4 Bytes	Not used	
19:16		4 Bytes	Not used	
15:08	Reserved	8 Bytes	Not used	
07:04	NSID	4 Bytes	[8:31] Reserved [0:7] Keyspace ID	

Bit	Description	
31:16	Command Identifier (CID)	As specified in the NVMe 1.2 specification
15:14	PRP or SGL for Data Transfer (PSDT)	As specified in the NVMe 1.2 specification
13:10	Reserved	Not used
09:08	Fused Operation (FUSE)	Always 00b, Normal operation
07:00	Opcode	0xB3

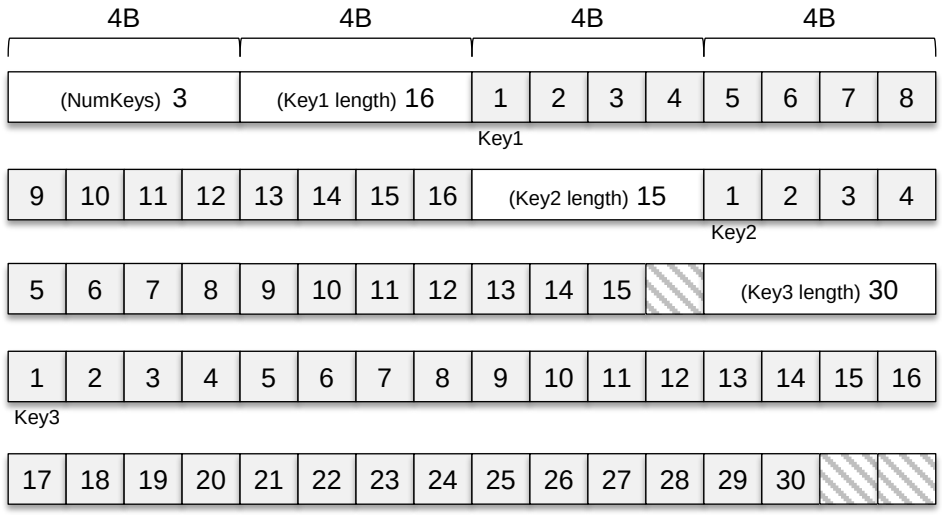
KV Command Format : Iterate

- **Iterate option : 0x01 (Iterate Open), 0x02 (Iterate Close) 0x04 (Key Iterate), 0x10 (Key Iterate with delete)**
- **Response**
  - Command Specific (for iterate open) : Iterate handle ID
  - Status Code: InvalidIterateHandle, NoAvailableIterateHandle, DuplicateIterateOpenRequest, IterateScanFinished

Bytes			Description	
			Iterate open (Option 0x01)	Iterate close (Option 0x02)
63:56	CDW14-15	8 Bytes	Not used	
55:52	CDW13	4 Bytes	Iterate bitmask	Not used
51:48	CDW12	4 Bytes	Iterate value	Not used
47:44	CDW11	4 Bytes	[16:31] Reserved [08:15] Iterate Request option	
			[00:07] Reserved	[00:07] Iterate handle
43:40	CDW10	4 Bytes	0x0 (should be zero)	
39:24	DPTR	16 Bytes	Not used	
23:16	MPTR	8 Bytes	Not used	
15:08	Reserved	8 Bytes	Not used	
07:04	NSID	4 Bytes	[8:31] Reserved [0:7] Keyspace ID	Not used
03:00	CDW0	4 Byte	See the below table	

Iterate Response Format

- **Memory buffer (32KB) includes a list of matched keys**



Who does what?

Host S/W vs. KV-SSD

	[Host S/W need to care]		[Host S/W simply calls KV command]
Idempotent Store	`if (!exist(key))` `  kv_store(key,value);`	→	`kv_store(key, value, idempotent_on);`

Iterate with Delete	`kv_iterate(&keys, option=key_read);`
	`for (key: keys)`
	`  kv_delete(key);`
		→	`kv_iterate(key, option=delete);`
			`while (!finished)`
			`  check_status();`

Large Value Store	`for (i=0; i<num_chunks; i++)` `  kv_store(key,value[i],i,num_chunks);`	→	`kv_store(key,value_2MB);`

Large Value Retrieve	`get_value_size(key, &value_size);`
	`num_chunks = value_size / chunk_size;`
	`for (i=0; i<num_chunks; i++)`
	`  kv_retrieve(key,value[i],i);`
		→	`kv_retrieve(key, &value, &value_2MB);`

How to update the firmware

- **Nvme-cli : https://www.mankier.com/1/nvme**
  - root privilege needed
- **F/W Types**
  1. Format F/W: Device is automatically formatted after F/W Download. Downloading Format F/W can revive the device under Error Mode
     - EGA50KOO_YYYYMMDD_ENC.bin
  2. Non-Format F/W: Update F/W only. All user data is preserved
     - EGA50KOO_YYYYMMDD_NF_ENC.bin

```
root@linux:/home# nvme fw-download /dev/nvme0 --fw=EGA50K01_20181022_ENC.bin --xfer=0x20000

Firmware download success

root@linux:/home# sleep 5

root@linux:/home# nvme fw-activate /dev/nvme0 --slot=2 --action=1

Success activating firmware action:1 slot:2, but a conventional reset is required
```

PCIe slot No. into which the SSD is inserted

FW Binary File

THE NEXT CREATION STARTS HERE

Placing **memory** at the forefront of future innovation and creative IT life