Flash Memory Summit Session:

Benefits of ZNS in Datacenter Storage Systems

Woosuk Chung, Director, Memory Systems R&D

Legal Disclaimer

The information contained in this document is claimed as property of SK hynix. It is provided with the understanding that SK hynix assumes no liability, and the contents are provided under strict confidentiality.

This document is for general guidance on matters of interest only. Accordingly, the information herein should not be used as a substitute for consultation or any other professional advice and services.

SK hynix may have copyrights and intellectual property right. The furnishing of document and information disclosure should be strictly prohibited.

SK hynix has right to make changes to dates, product descriptions, figures, and plans referenced in this document at any time. Therefore the information herein is subject to change without notice.

CONTENTS

Introduction

Performance Evaluation

RocksDB, F2FS with ZNS SSD

Summary

Introduction: Zoned Namespaces Proposal

- **Previous proposals are not a complete solution for data center storage system**
- **New proposal, Zoned Namespaces (ZNS), appears to be an optimal solution**

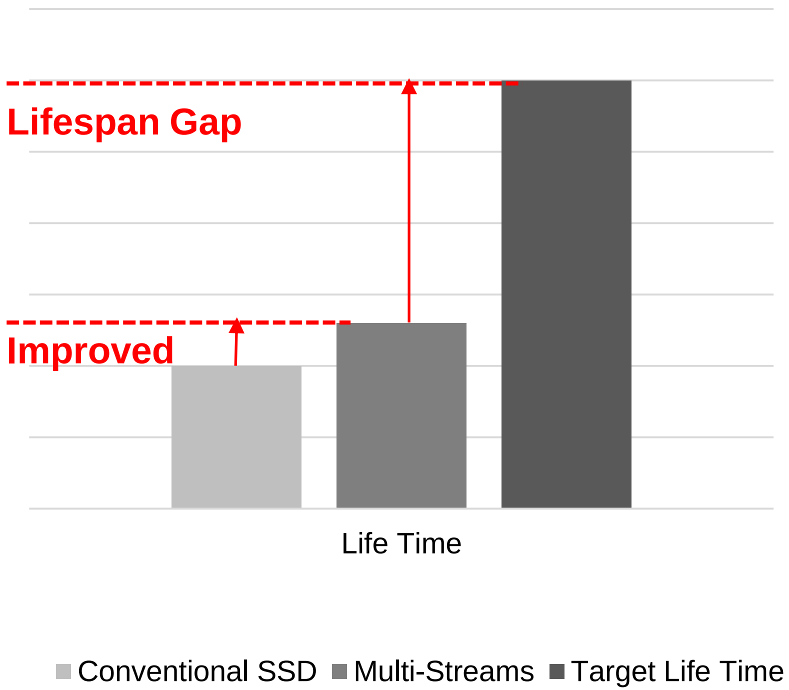
✓ Available
● Incomplete
✗ Not planned

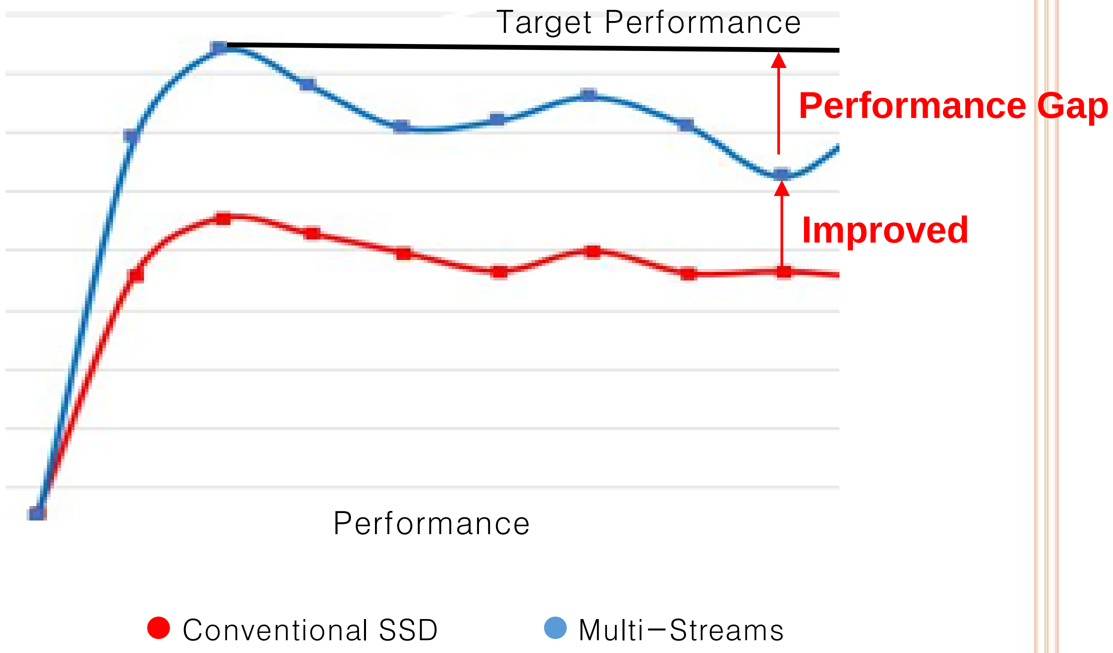
	Log Abstraction	In-Host Placement Policy	In-Drive Reliability
Multi-Streams SSD (HotStor `14)	✗	●	✓
OCSSD 1.2 (ASPLOS `16)	✓	✓	✗
IO Determinism (Fall `16)	✗	●	✓
OCSSD 2.0 (FAST `17)	✓	✓	✓
Zoned Namespaces (NVMe Spec. `19)	✓	✓	✓

Reference MSFT, `17 Storage Developer Conference

Introduction: Case of Multi-Streams

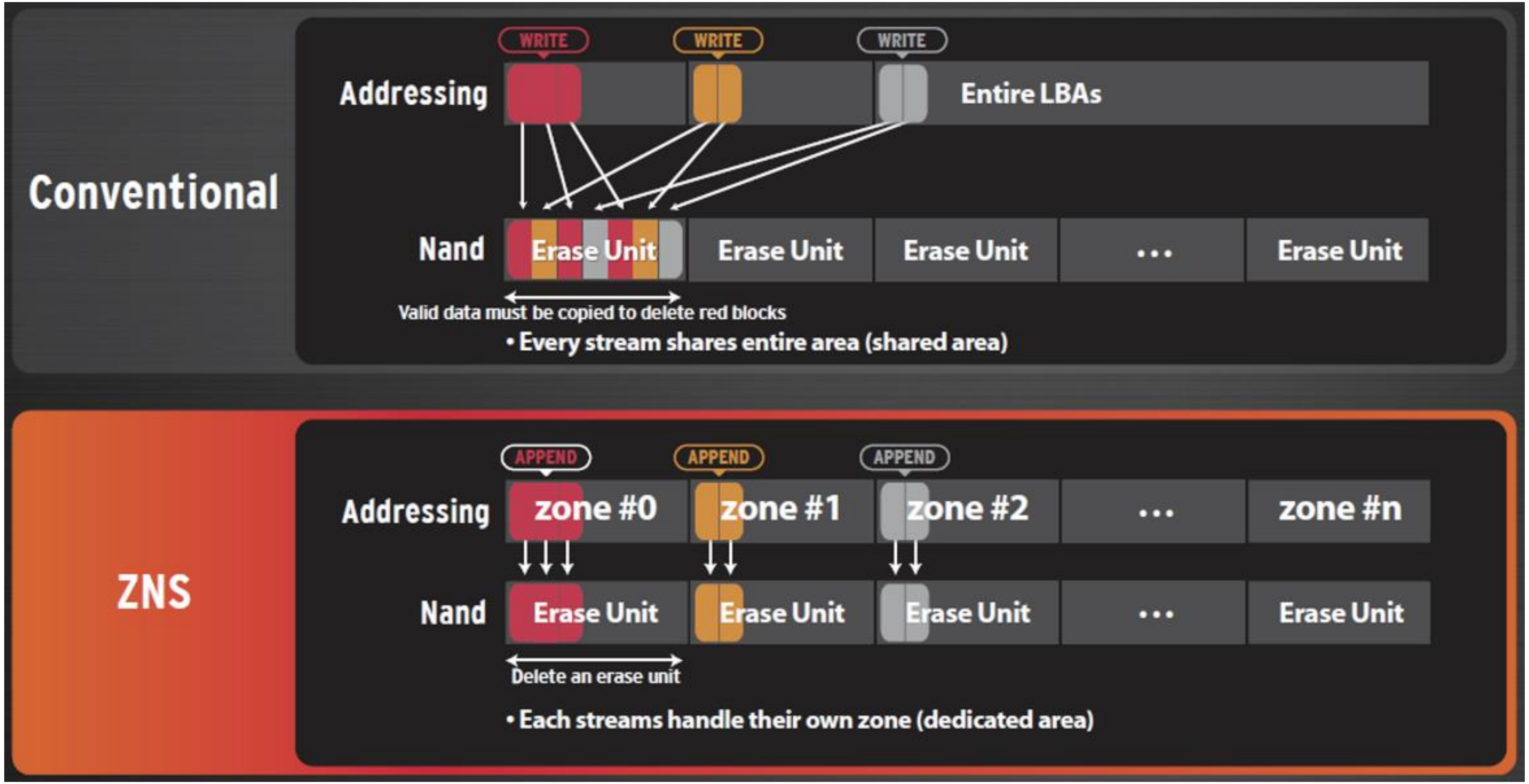
- Garbage Collection is reduced but not completely removed
- Lifespan & Performance can be enhanced but not to the optimal level





Introduction: ZNS Concept

- Only sequential write is accepted in each Zone (random write is not allowed)
- Zones are erased by the host issuing a special command, Zone Reset (no GC)



Introduction: SK hynix‘s ZNS SSD Prototype

- **Prototype available on two SK hynix SSD products**

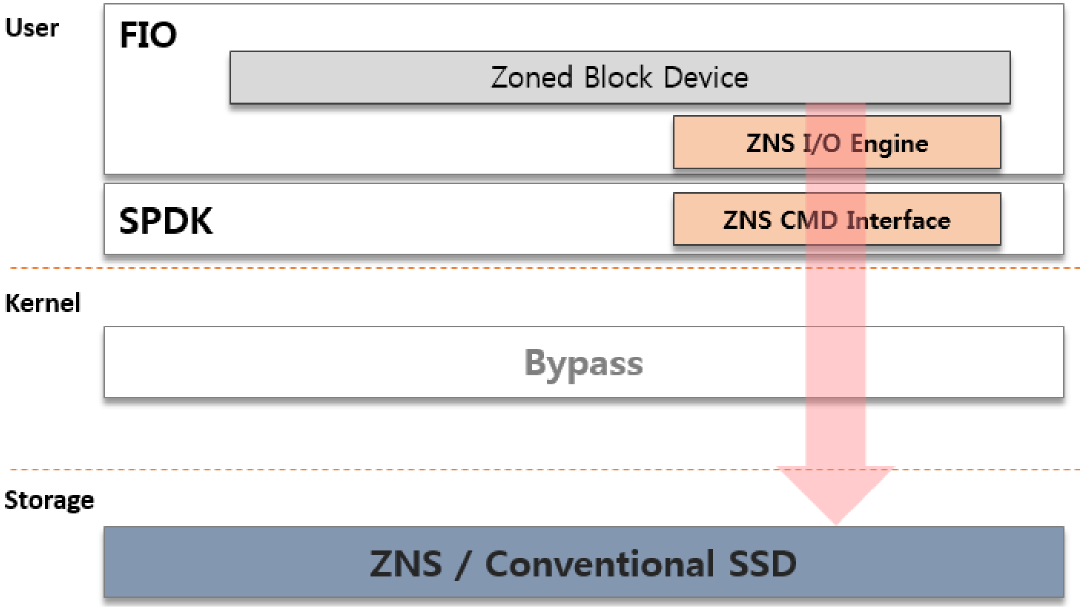
Item		PE4011
Interface		PCIe Gen3 x 4
Protocol		NVMe 1.2.1
Form Factor		M.2 22110
Capacity		1920GB
NAND	Density	512 Gb
	Type	3D **TLC**

Item		PE6011
Interface		PCIe Gen3 x 4
Protocol		NVMe 1.3
Form Factor		U.2 7mm
Capacity		3840TB
NAND	Density	512 Gb
	Type	3D **TLC**

Performance Evaluation: Configuration & Environment

- New ZNS commands added in SPDK
- Emulated workload generated and run by FIO
- Kernel S/W stack is bypassed to remove overhead

Test Framework



Environment

- Hardware:
  - CPU: Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz
  - Memory: 251GB
  - SSD(ZNS/Conventional): PE4011, PE6011
- Software:
  - Ubuntu 15.04
  - Linux 4.20.0 x86_64
  - FIO-3.12, SPDK

Performance Evaluation: Expected Benefits

SK hynix

I. Extend SSD Lifespan

II. Reduce Read Tail Latency (QoS)

III. Improve I/O Performance

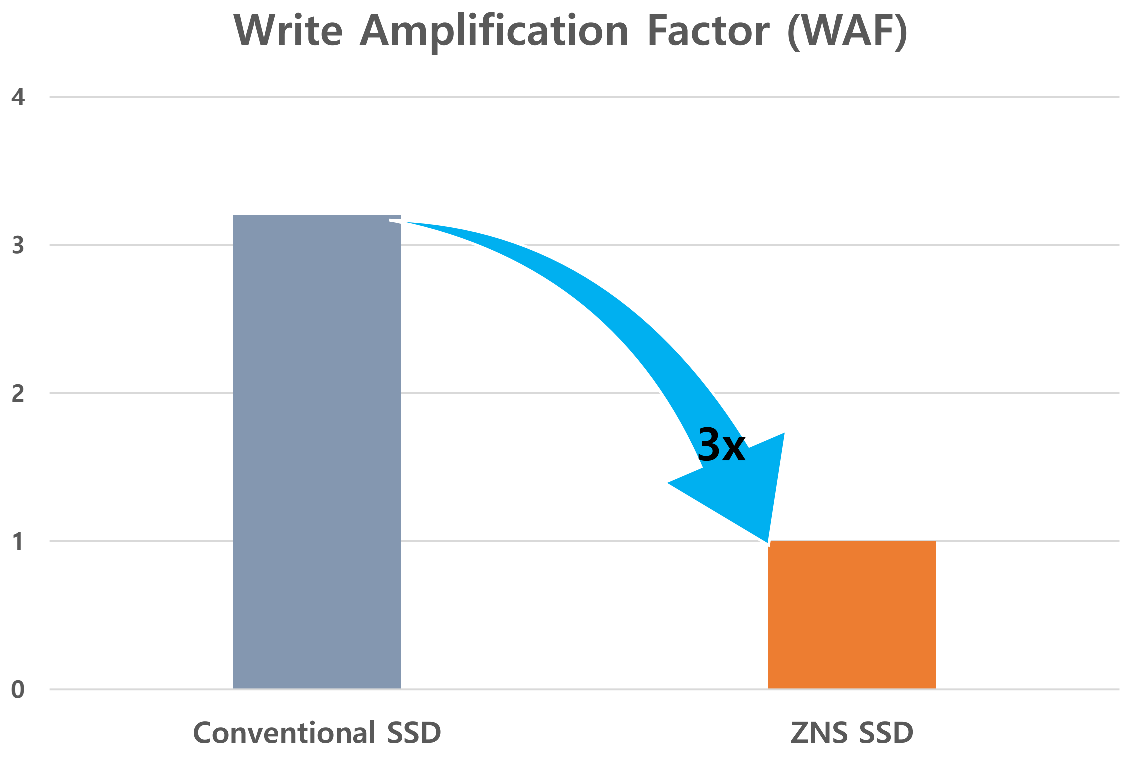
IV. Reduce Overprovisioning

V. Reduce DRAM in SSD

Performance Evaluation: Expected Benefits (1/5)

- Extend SSD Lifespan
  - Increase lifespan 3x for the case of 8-writes
  - No Garbage Collection is required

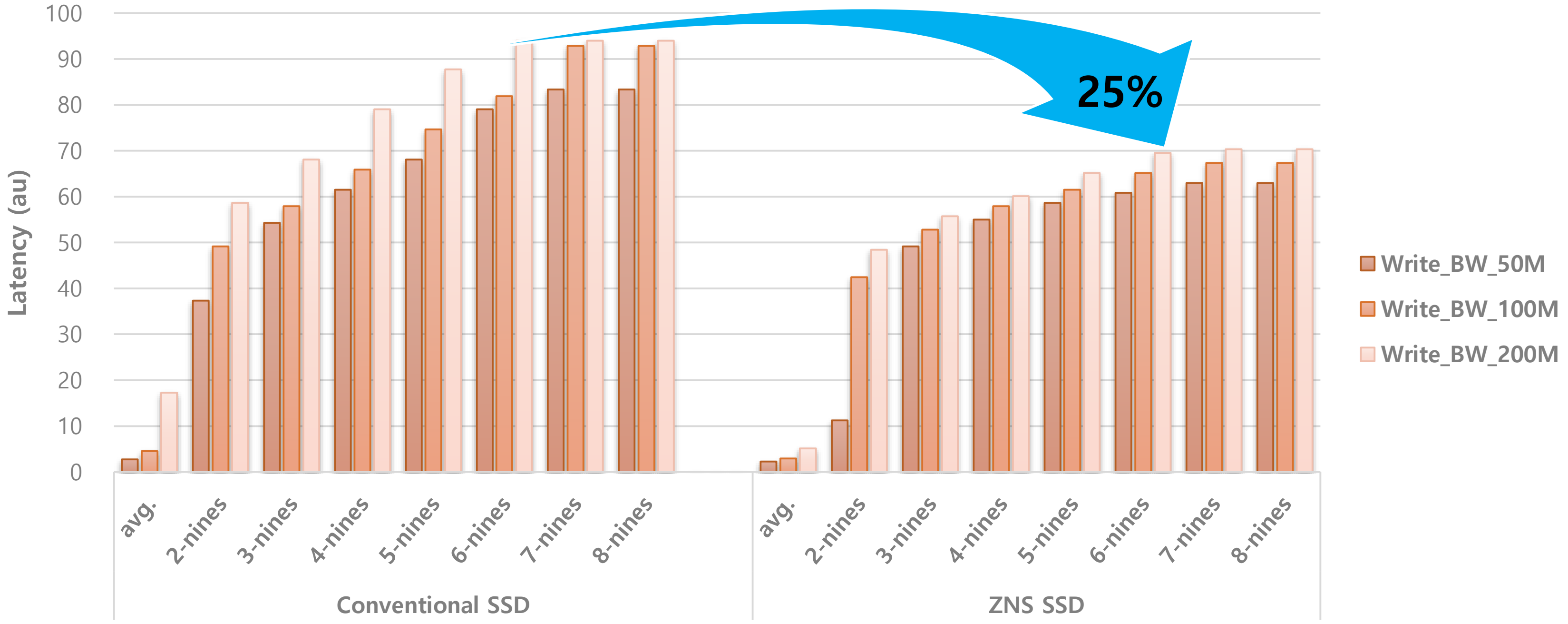
$$WAF = \frac{Bytes\ written\ to\ NAND}{Bytes\ written\ from\ Host}$$



128KB(Block Size), 8-writes

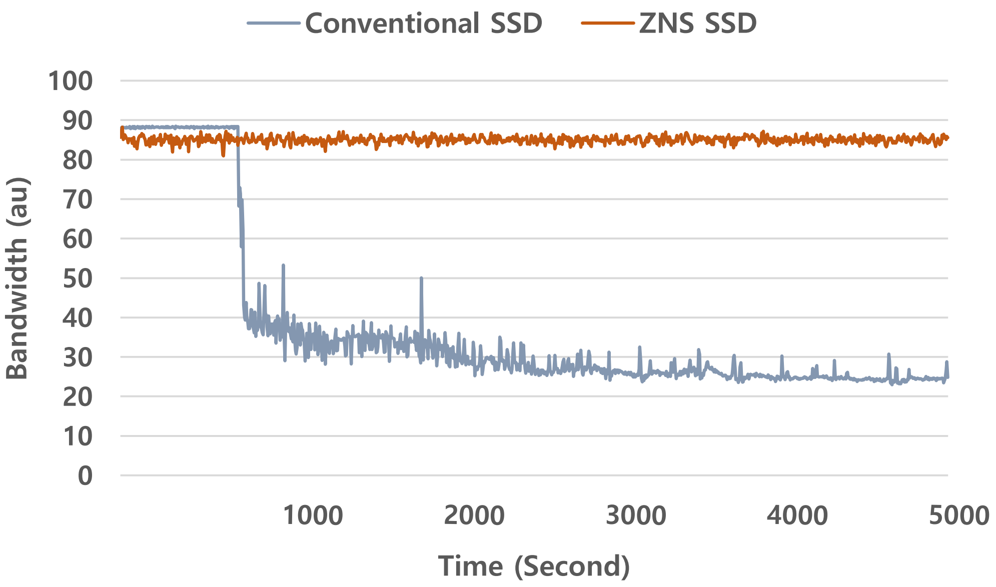
Performance Evaluation: Expected Benefits (2/5)

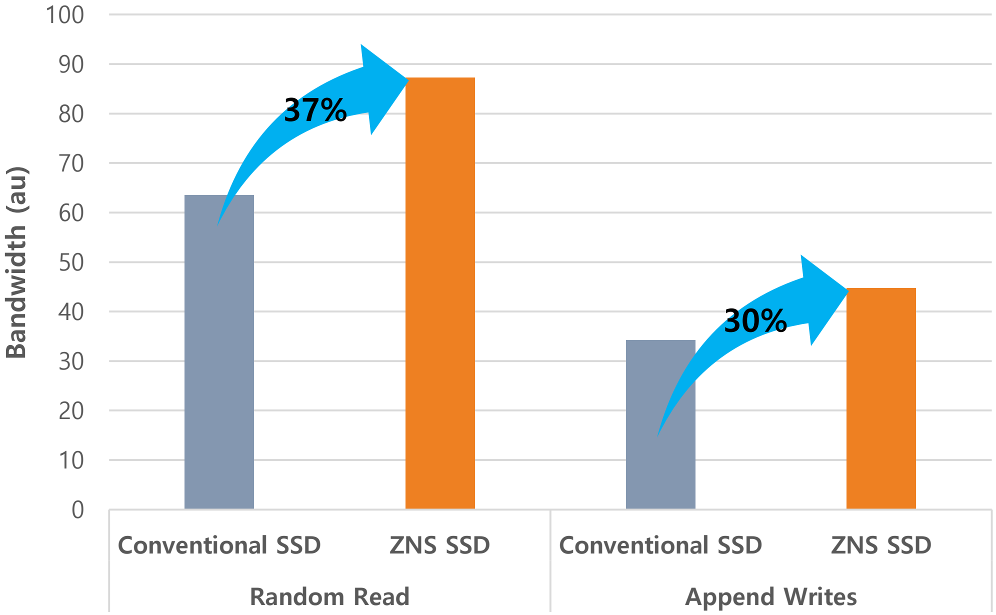
- Improve Read Tail Latency
  - Reduce IO interference by SSD’s internal background operations
  - Improve read response for mixed workloads



Performance Evaluation: Expected Benefits (3/5)

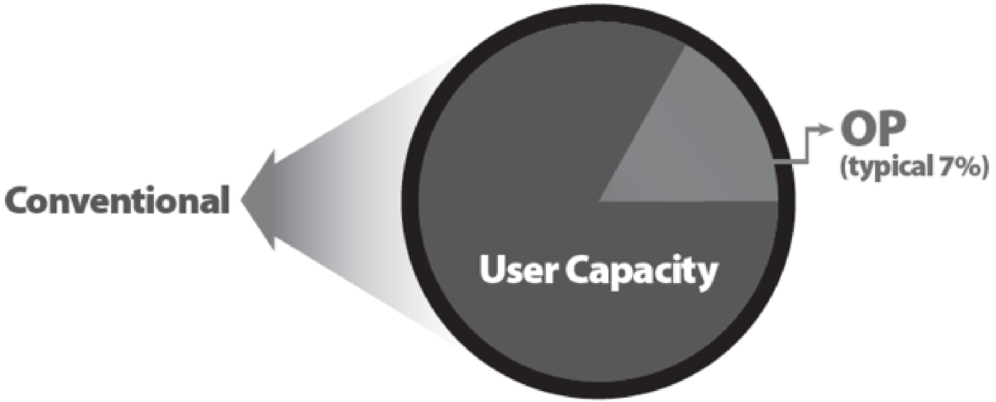
- Improve I/O Performance
  - Getting consistent throughput
  - Higher bandwidth for mixed workloads (1 Random Read and 5 Writes)

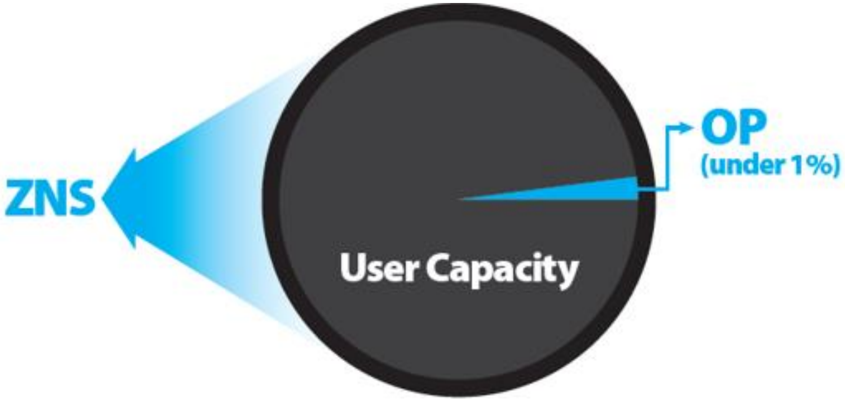




Performance Evaluation: Expected Benefits (4/5)

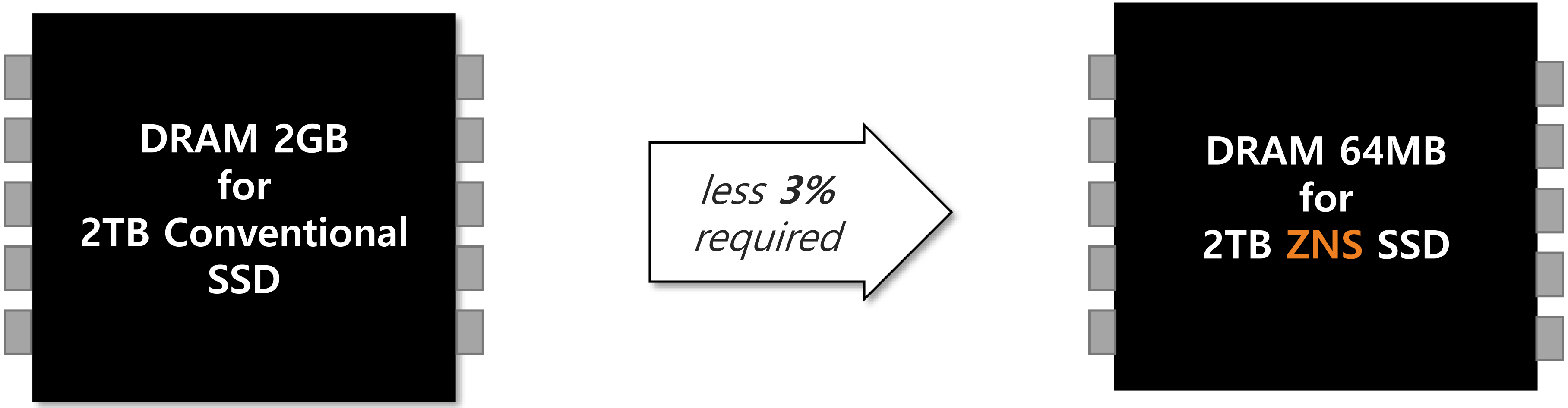
- Less Overprovisioning(OP)
  - Reduce OP area for removing Garbage Collection
  - Eventually, user capacity is increased





Performance Evaluation: Expected Benefits (5/5)

- Reduce DRAM size requirement for SSD
  - Less DRAM is required per 1TB capacity
  - Make rooms for more critical DRAM use



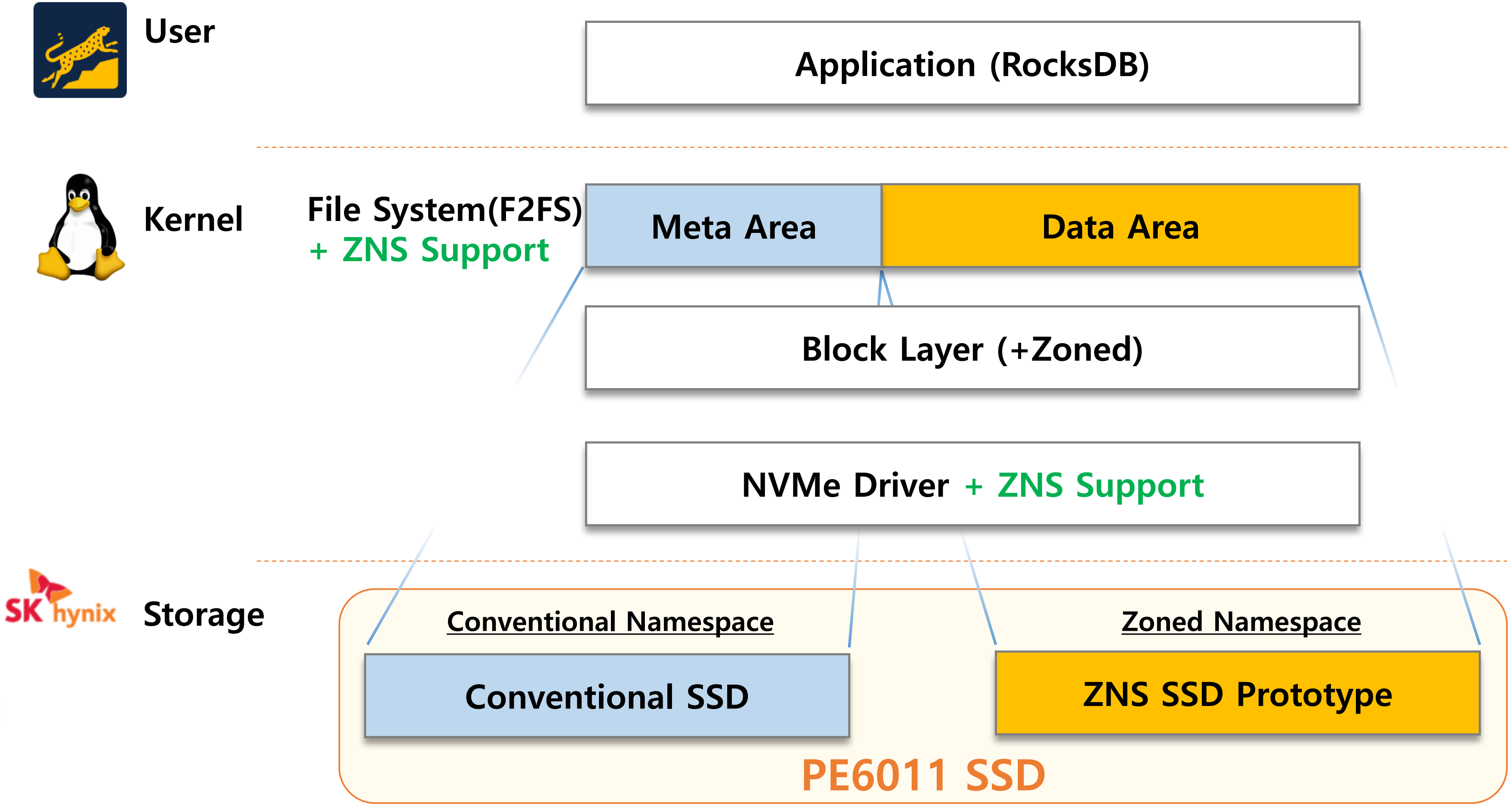
DRAM 2GB
for
2TB Conventional
SSD

less ***3%***
required

DRAM 64MB
for
2TB ZNS SSD

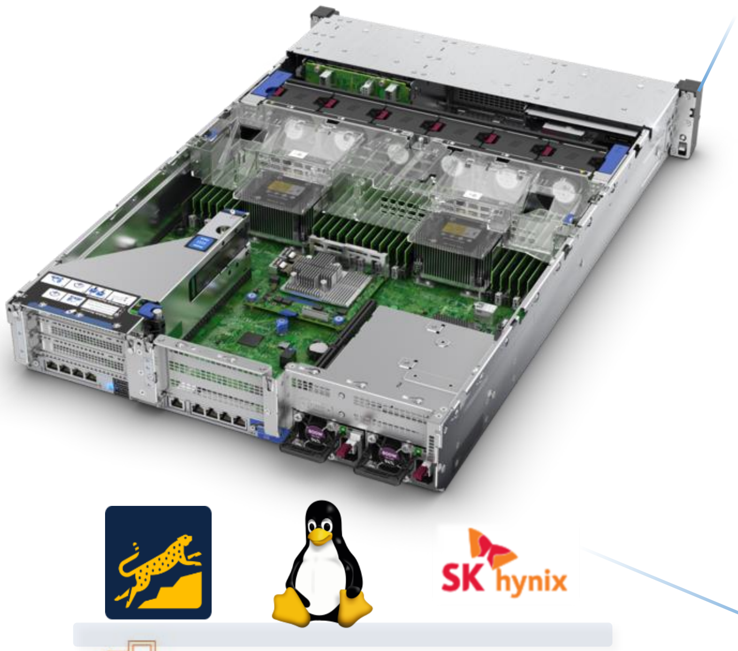
SW Enablement: RocksDB, F2FS with ZNS SSD

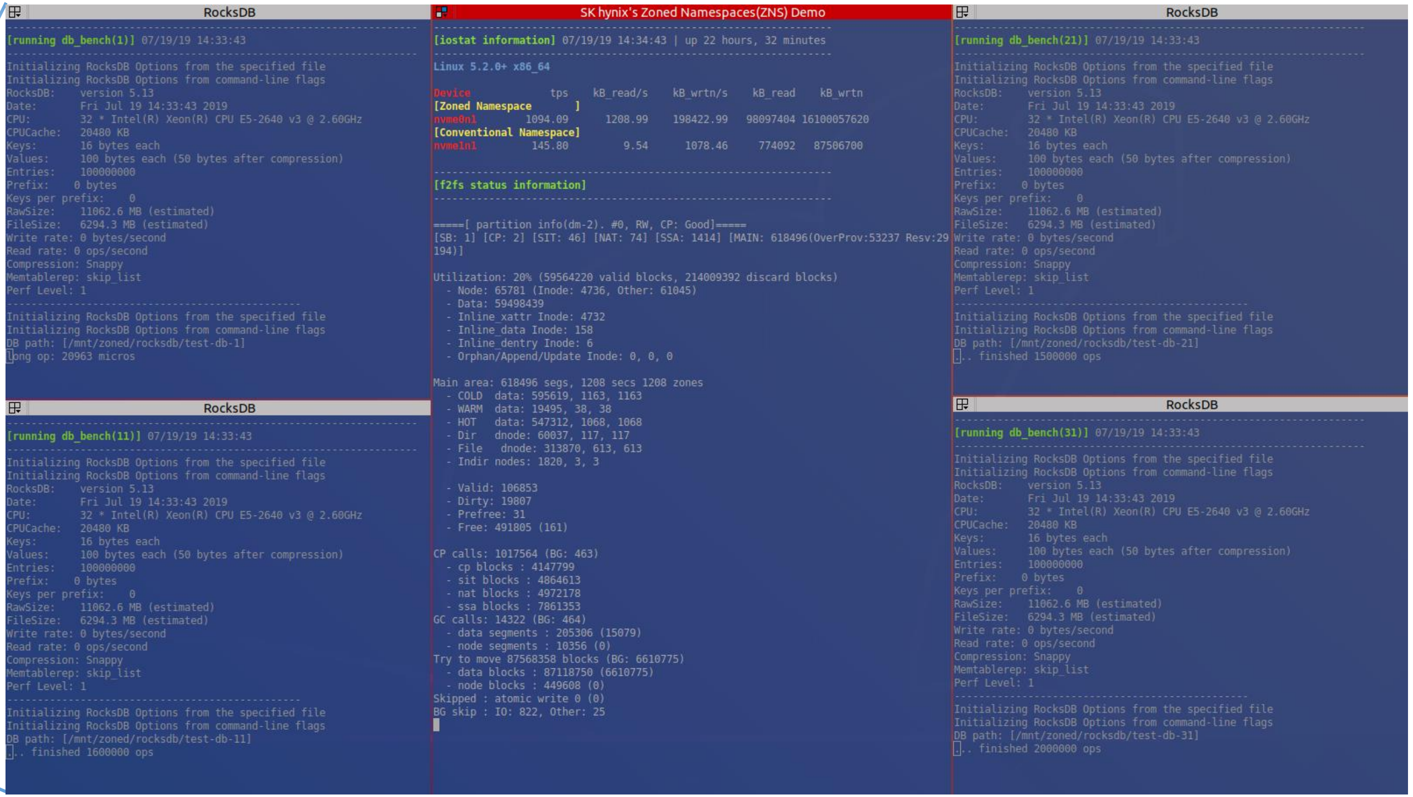
- "Linux kernel that supports Zoned Device" + "Two types of PE6011 SSD"
  - PE6011-based ZNS SSD for append-only write in F2FS
  - Conventional SSD for random-write in F2FS



SW Enablement: Live Demo

- Running RocksDB with db_bench on ZNS-configured Linux Host System
- Key-value data is stored into PE6011 ZNS SSD





```
------------------------------------------------------------------
[running db_bench(1)] 07/19/19 14:33:43
------------------------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 5.13
Date:       Fri Jul 19 14:33:43 2019
CPU:        32 * Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
CPUCache:   20480 KB
Keys:       16 bytes each
Values:     100 bytes each (50 bytes after compression)
Entries:    100000000
Prefix:    0 bytes
Keys per prefix:    0
RawSize:    11062.6 MB (estimated)
FileSize:   6294.3 MB (estimated)
Write rate: 0 bytes/second
Read rate: 0 ops/second
Compression: Snappy
Memtablerep: skip_list
Perf Level: 1
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/mnt/zoned/rocksdb/test-db-1]
long op: 20963 micros
```

```
------------------------------------------------------------------
[running db_bench(11)] 07/19/19 14:33:43
------------------------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 5.13
Date:       Fri Jul 19 14:33:43 2019
CPU:        32 * Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
CPUCache:   20480 KB
Keys:       16 bytes each
Values:     100 bytes each (50 bytes after compression)
Entries:    100000000
Prefix:    0 bytes
Keys per prefix:    0
RawSize:    11062.6 MB (estimated)
FileSize:   6294.3 MB (estimated)
Write rate: 0 bytes/second
Read rate: 0 ops/second
Compression: Snappy
Memtablerep: skip_list
Perf Level: 1
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/mnt/zoned/rocksdb/test-db-11]
... finished 1600000 ops
```

```
------------------------------------------------------------------
[iostat information] 07/19/19 14:34:43 | up 22 hours, 32 minutes
------------------------------------------------------------------
Linux 5.2.0+ x86_64

Device            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
[Zoned Namespace       ]
nvme0n1        1094.09      1208.99    198422.99   98097404 16100057620
[Conventional Namespace]
nvme1n1         145.80         9.54      1078.46     774092   87506700

------------------------------------------------------------------
[f2fs status information]
------------------------------------------------------------------

=====[ partition info(dm-2). #0, RW, CP: Good]=====
[SB: 1] [CP: 2] [SIT: 46] [NAT: 74] [SSA: 1414] [MAIN: 618496(OverProv:53237 Resv:29
194)]

Utilization: 20% (59564220 valid blocks, 214009392 discard blocks)
  - Node: 65781 (Inode: 4736, Other: 61045)
  - Data: 59498439
  - Inline_xattr Inode: 4732
  - Inline_data Inode: 158
  - Inline_dentry Inode: 6
  - Orphan/Append/Update Inode: 0, 0, 0

Main area: 618496 segs, 1208 secs 1208 zones
  - COLD  data: 595619, 1163, 1163
  - WARM  data: 19495, 38, 38
  - HOT   data: 547312, 1068, 1068
  - Dir   dnode: 60037, 117, 117
  - File   dnode: 313870, 613, 613
  - Indir nodes: 1820, 3, 3

  - Valid: 106853
  - Dirty: 19807
  - Prefree: 31
  - Free: 491805 (161)

CP calls: 1017564 (BG: 463)
  - cp blocks : 4147799
  - sit blocks : 4864613
  - nat blocks : 4972178
  - ssa blocks : 7861353
GC calls: 14322 (BG: 464)
  - data segments : 205306 (15079)
  - node segments : 10356 (0)
Try to move 87568358 blocks (BG: 6610775)
  - data blocks : 87118750 (6610775)
  - node blocks : 449608 (0)
Skipped : atomic write 0 (0)
BG skip : IO: 822, Other: 25
```

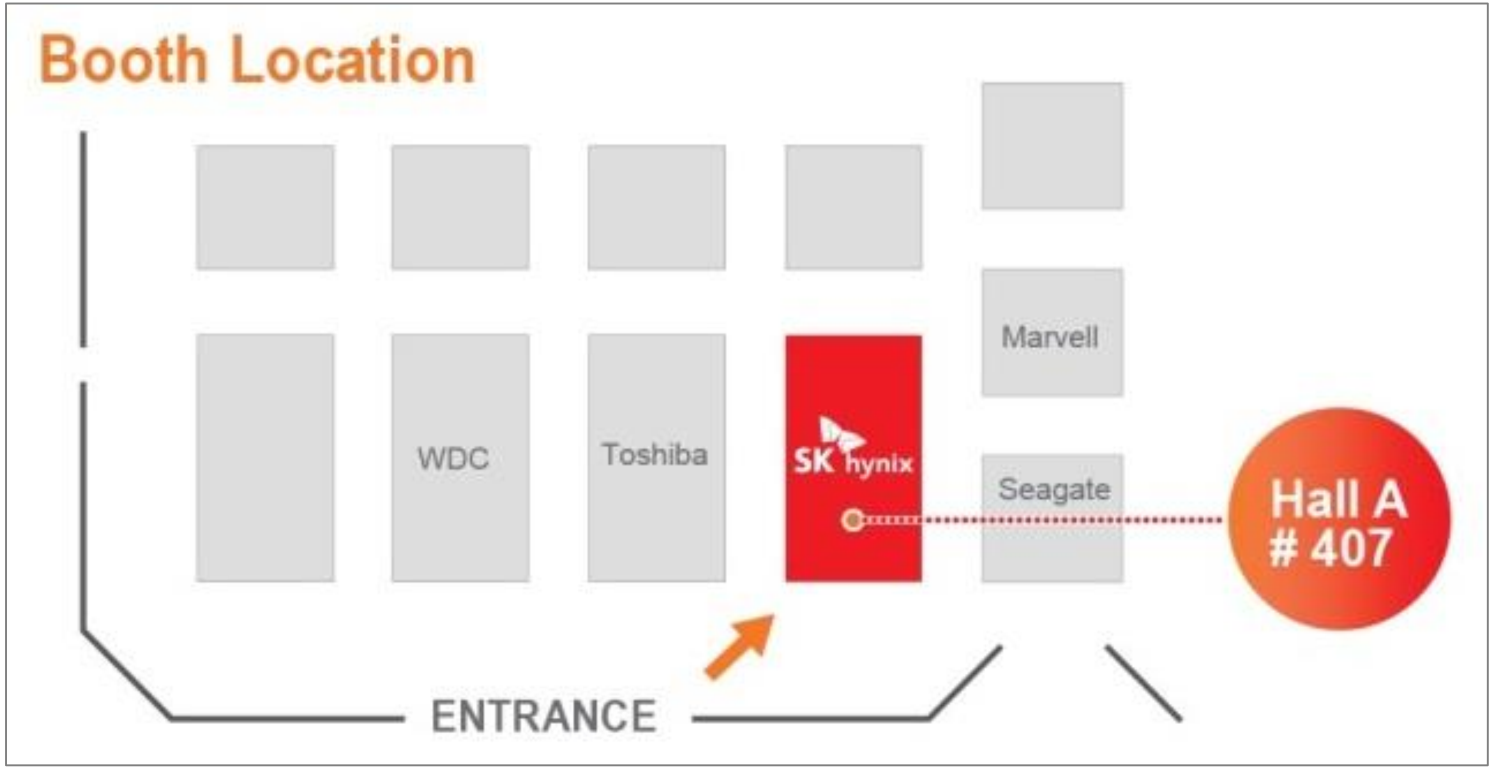
```
------------------------------------------------------------------
[running db_bench(21)] 07/19/19 14:33:43
------------------------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 5.13
Date:       Fri Jul 19 14:33:43 2019
CPU:        32 * Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
CPUCache:   20480 KB
Keys:       16 bytes each
Values:     100 bytes each (50 bytes after compression)
Entries:    100000000
Prefix:    0 bytes
Keys per prefix:    0
RawSize:    11062.6 MB (estimated)
FileSize:   6294.3 MB (estimated)
Write rate: 0 bytes/second
Read rate: 0 ops/second
Compression: Snappy
Memtablerep: skip_list
Perf Level: 1
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/mnt/zoned/rocksdb/test-db-21]
... finished 1500000 ops
```

```
------------------------------------------------------------------
[running db_bench(31)] 07/19/19 14:33:43
------------------------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
RocksDB:    version 5.13
Date:       Fri Jul 19 14:33:43 2019
CPU:        32 * Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
CPUCache:   20480 KB
Keys:       16 bytes each
Values:     100 bytes each (50 bytes after compression)
Entries:    100000000
Prefix:    0 bytes
Keys per prefix:    0
RawSize:    11062.6 MB (estimated)
FileSize:   6294.3 MB (estimated)
Write rate: 0 bytes/second
Read rate: 0 ops/second
Compression: Snappy
Memtablerep: skip_list
Perf Level: 1
------------------------------------------------
Initializing RocksDB Options from the specified file
Initializing RocksDB Options from command-line flags
DB path: [/mnt/zoned/rocksdb/test-db-31]
... finished 2000000 ops
```

Summary

1. **Two SK hynix's NVMe ZNS Prototypes**
   - PE4011
   - PE6011
2. **Expected Benefits with ZNS**
   - Extend SSD Lifespan
   - Improve Performance & QoS
   - Reduce SSD's memory resource requirement (OP, DRAM)
3. **Linux Host enabled with ZNS Prototype**

Learn more about SK hynix



Visit SK hynix
@ booth #407

Experience SK hynix products and demos & get a free giveaway!

SK hynix

Thank you

Growing together

for better tomorrow