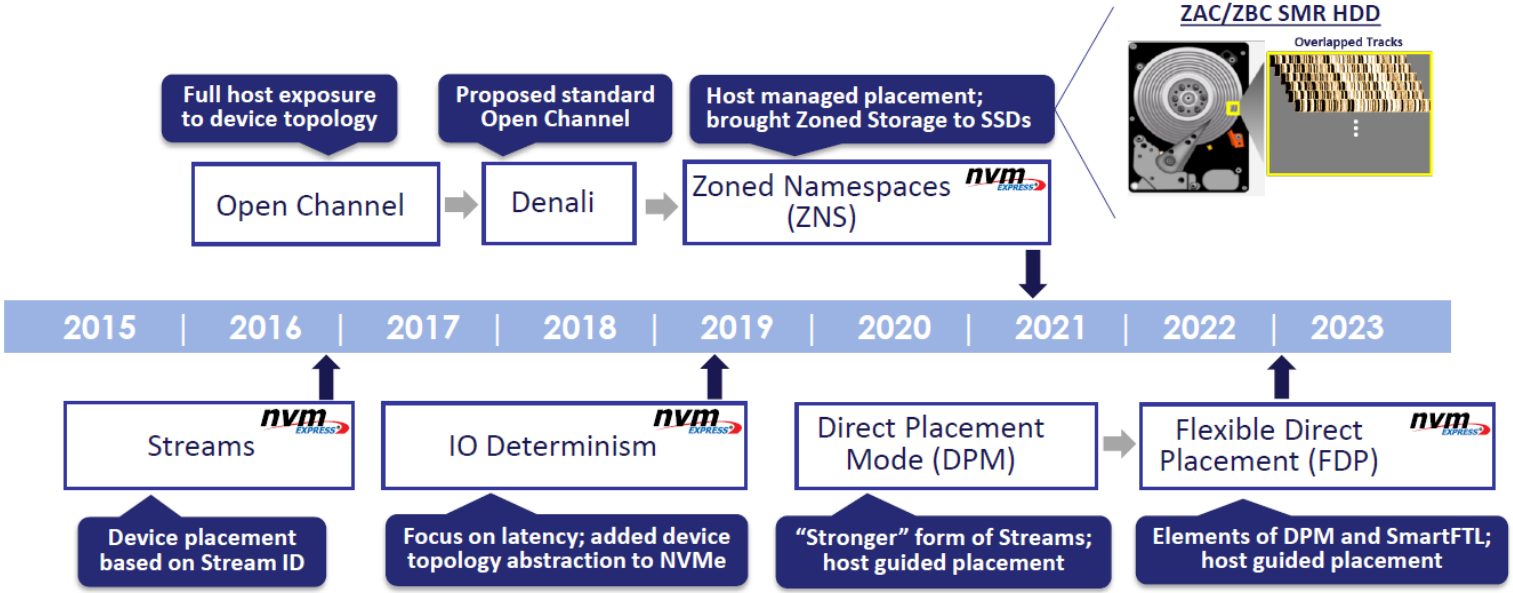
FDP와 Eco System

2023/07/04

Overview

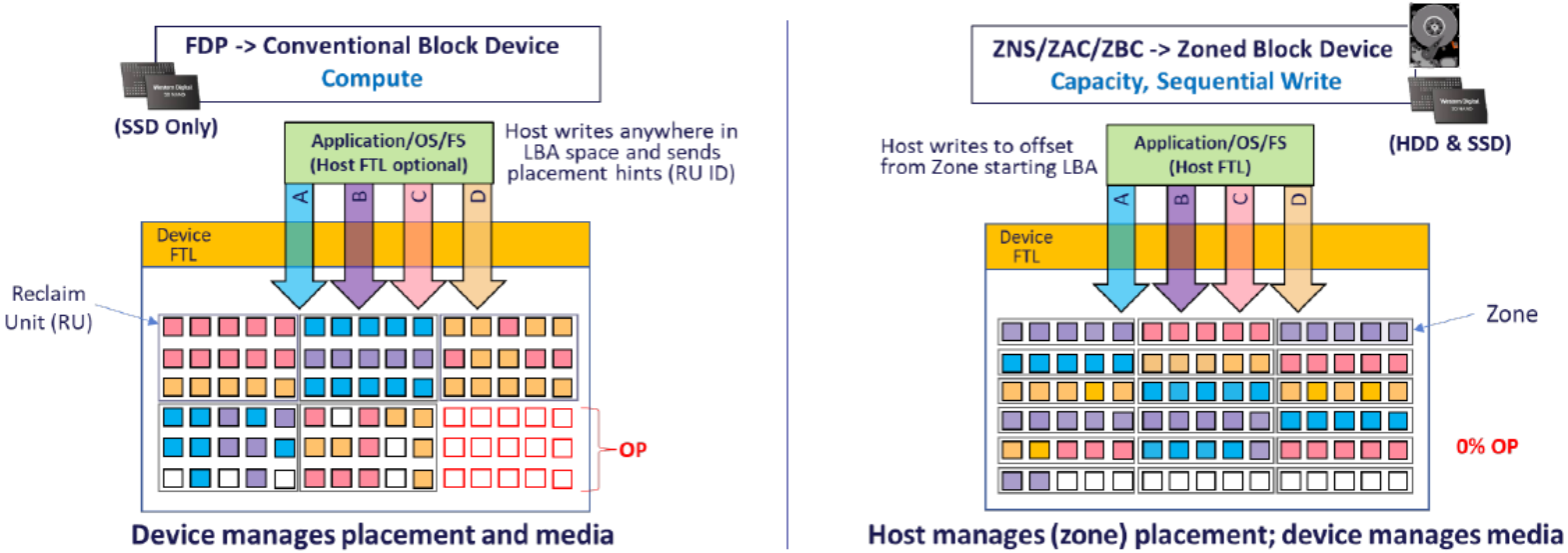
A Brief History of NVMe Data Placement Debates (OCP'22)



Overview

Two Data Placement Approaches (OCP'22)

Common Goals: Low write amp (endurance, predictable performance)



Two interfaces, two use cases ➔ Ecosystem now needs stability

ZNS

GC Overhead in SSDs

GC Overhead in traditional SSD

- SSD는 page 단위로 write / block 단위로 erase
- 새로운 공간 할당을 위해 valid data를 copy 수행하여야 함(GC)
- WAF(Write Amplification Factor): 클 수록 GC로 인한 overhead 증가

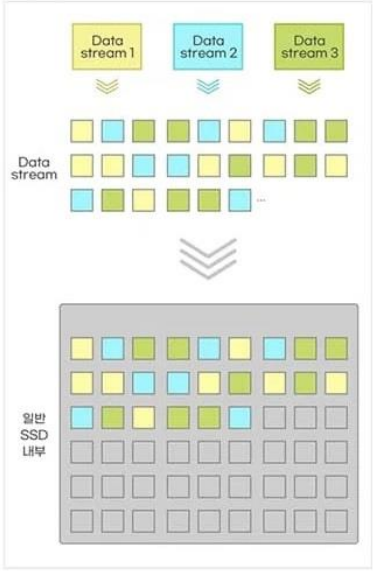
WAF를 최소화하기 위한 방법

- block 내 invalid page의 파편화를 방지하여 valid page copy를 최소화
- **GC 동작이 필요하지 않도록 Host level에서 I/O를 수행 -> ZNS SSD**

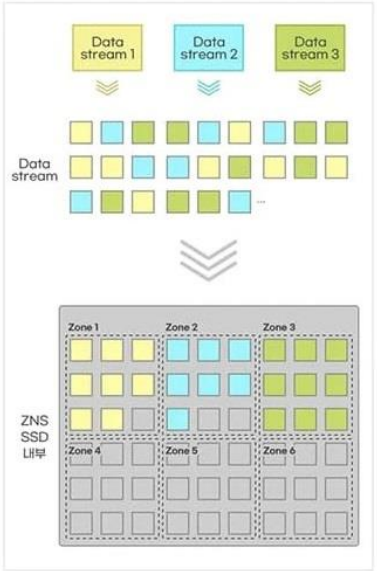
Zoned Namespace SSD

Zoned Namespace SSD(ZNS SSD)

- Zoned Storage Device 모델을 따르는 NVMe interface specification
- NVMe2.0 Spec 일부로 정의된 NVMe ZNS Command Set 구현
- 기존 SMR HDD 지원에 더해, NAND 특화된 추가적인 기능 지원



일반 SSD



ZNS SSD

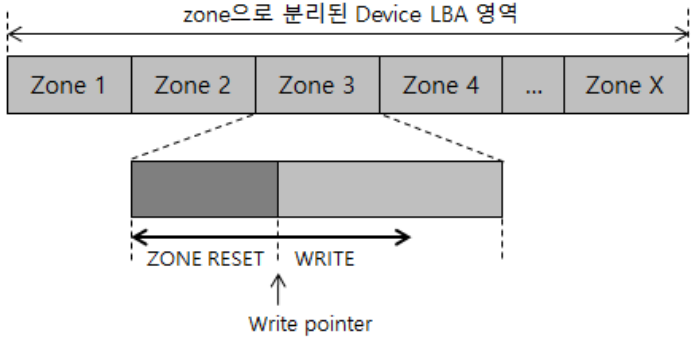
Zoned Storage Device

Zoned Storage Device 모델

- 일정 크기의 zone으로 나뉘어진 LBA 영역을 가진 저장 장치
- 각각의 zone은 특정한 write 제한을 가짐

Zone

- sequential write만을 지원
- 다음 write가 수행될 write pointer 위치를 저장
- 이미 쓰여진 곳은 overwrite 불가능
- 재사용을 위해 zone reset을 통해 zone 전체를 초기화



Zoned interface SSD 적용

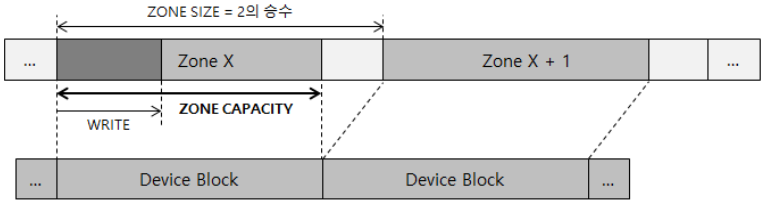
- 기존 SMR HDD 지원 위한 모델
- 해당 모델의 zone 제한은 NAND 플래시 및 SSD 제한과 유사
- Zoned interface 사용 시, **Host level(응용)**에서 각 zone 및 data I/O 관리 수행
- SSD에 Zoned Storage Device model 적용 -> **Device 내 GC 동작 제거 가능**

Reference: "NVM Express Zoned Namespace Command Set Specification 1.1"

Zoned Namespace SSD

Zone Capacity

- ZNS zone size = 2의 승수
- 하위 미디어 특성에 맞춘 실제 저장 용량



zone capacity

Active Zones

- Zone state
  - empty: 최초 상태, 아무것도 쓰여지지 않음
  - open: 현재 데이터가 쓰여질 수 있는 상태(리소스 할당)
  - closed: 현재 데이터가 쓰여질 수 없는 상태(리소스 반환)
  - full: 내부 데이터가 가득 찬 상태

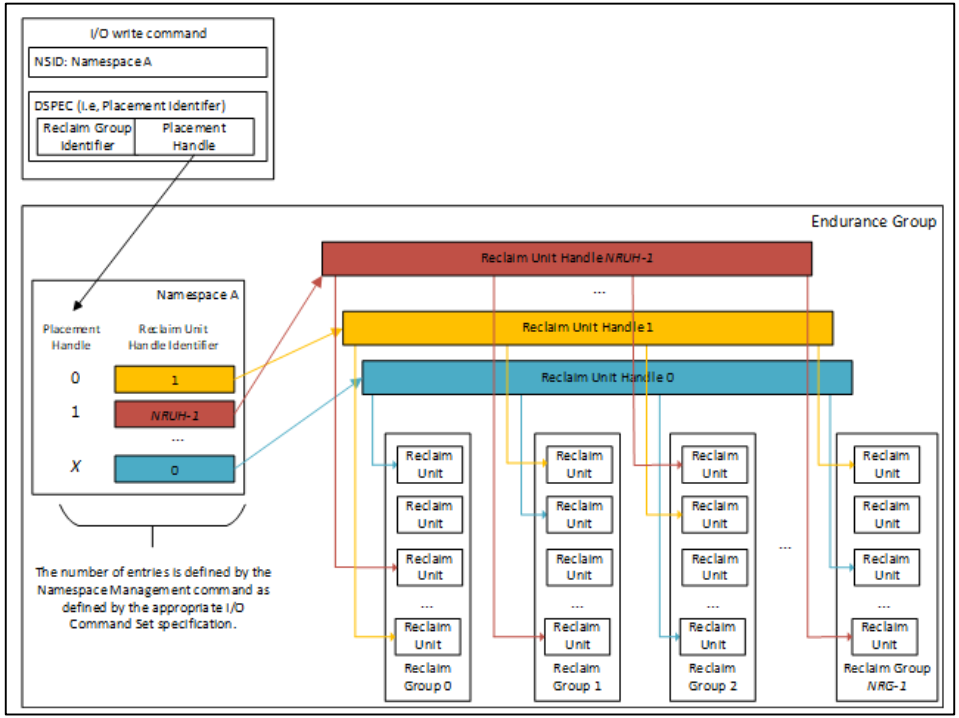
- ZNS Active Zones
  - 기존 Open zone 개수 제한 존재
  - Open zone + Closed zone = Active zones
  - 데이터를 저장하는 총 Zone 수를 제한함

ZNS zone states

State	Zone Characteristics		
	Valid Write Pointer	Active Resources	Open Resources
Empty	Yes	No	No
Implicit Open	Yes	Yes	Yes
Explicit Open	Yes	Yes	Yes
Close	Yes	Yes	No
Full	No	No	No
Read Only	No	No	No
Offline	No	No	No

FDP

FDP Entities (1)



Reference: "TP4146a Flexible Data Placement Specification"

Reclaim Unit(RU)

- 다른 RU를 간섭하지 않고 Controller에 의해 물리적으로 erase될 수 있는 비 휘발성 저장장치의 논리적 표현. 하나의 RU는 하나의 Reclaim Group (RG) 에 속함

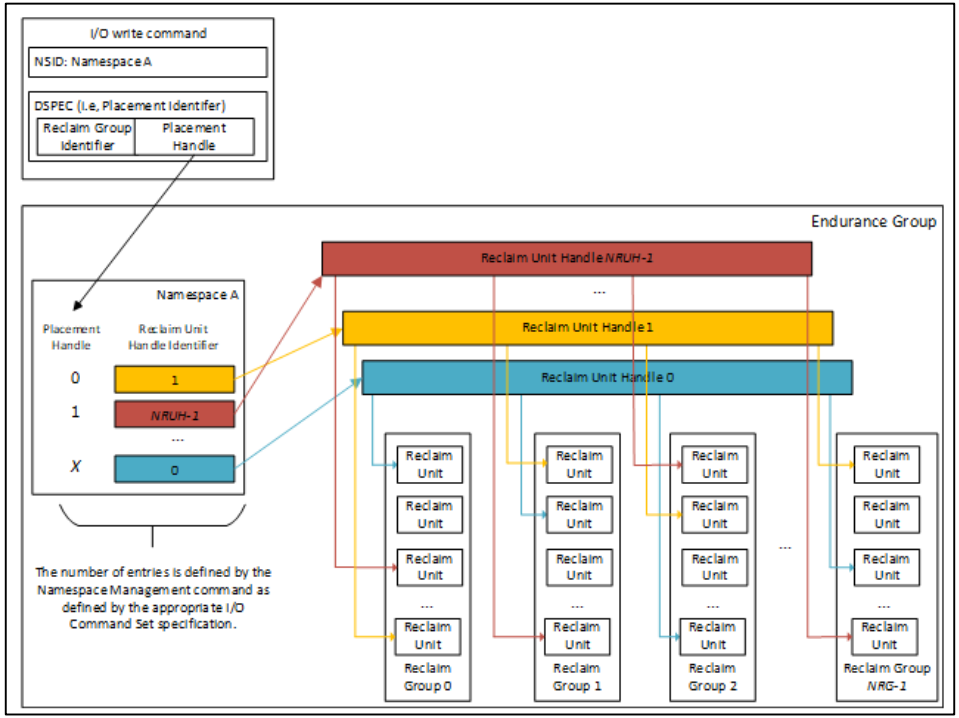
Reclaim Group(RG)

- 한 개 혹은 그 이상의 Reclaim을 포함하는 단위.

Reclaim Unit Handle(RUH)

- 하나의 RG와 그 RG에 속한 RU 하나를 가리키는 ID. Device 관점에서 Write시 RUH로 Target RU에 접근 (multi-stream에서의 Stream ID와 유사)

FDP Entities (2)



Reference: "TP4146a Flexible Data Placement Specification"

Placement Handle(PHNDL)

- Host 관점에서 Namespace 범위에서 Endurance Group 범위의 RU Handle에 Mapping되어 RG의 특정 RU를 가리키는 Identifier.

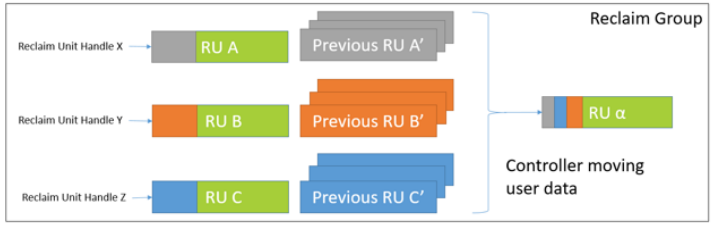
Placement Identifier

- 하나의 RG와 그 RG에 속한 RUH 하나를 명세하는 자료구조 {RG + PHNDL}

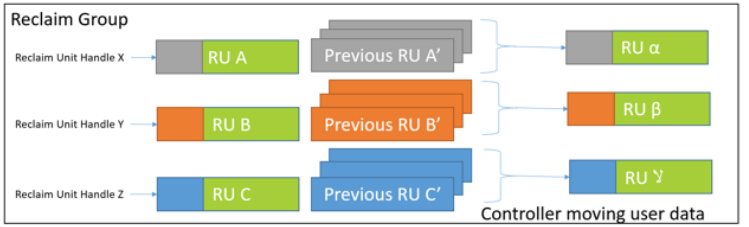
FDP Isolated

Extended Capabilities

- RU의 User data는 GC에 의해 다른 RU로 이동 가능
- 설정에 따라 이동 범위 다름 (RU내에서)
  - Initially Isolated : RG내의 모든 RU로 이동 가능
  - Persistently Isolated : RG내에서 동일 RUH에 의해 쓰여졌던 RU 내에서 이동 가능



< Initially Isolated >



< Persistently Isolated >

FDP Command

Extended Capabilities (Controller는 아래 Command를 support)

- 5.16 Get Log Page Command
  - ✓ Feature Identifiers Supported and Effects log page
  - ✓ FDP Configurations log page
  - ✓ RUH Usage log page
  - ✓ FDP Statistics log page
  - ✓ FDP Events log page

- 5.27 Set Feature Command
  - ✓ Get/Set Feature

- 7 I/O Command
  - ✓ I/O Management Send/Receive

- 8.7 Directives

Comparison ZNS vs FDP

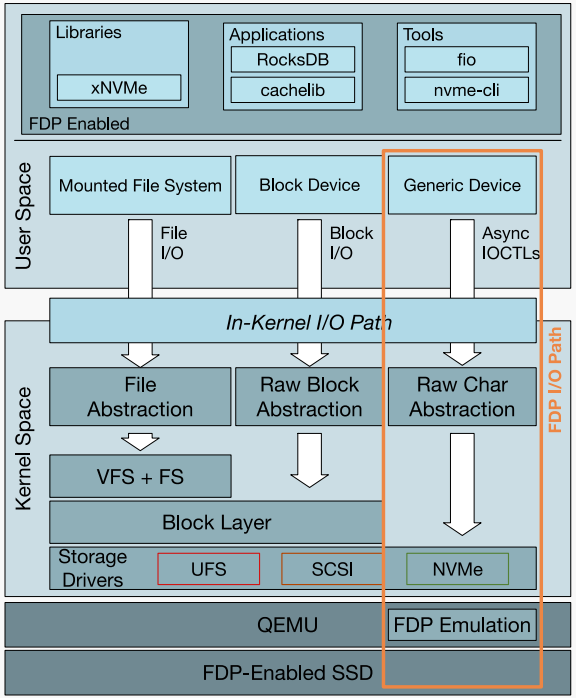
Zoned Namespaces	Flexible Data Placement
WAF = 1 guaranteed WAF 개선 최적화	WAF = 1 achievable with feedback WAF 개선 최적화
기존 SSD 및 Infra와 호환성 이슈 NVMe 외 별도 Spec.	기존 SSD 및 Infra와 호환성 가능 NVMe Spec.
Zones	Reclaim Units
Sequential Write만 허용, Zone Append command	Sequential, Random Write 허용
QD > 1: LBA known at Write Completion	QD > 1: LBA known at Write Submission
Zone written by a single namespace	Reclaim Unit written by one or more namespaces
Zone Open 시 Device가 지정하고, Write LBA로 Zone 번호를 계산해서 Zone Block에 저장	Host가 Write마다 RG(Die) x RUH(Stream) 지정
OP 절약을 통한 저장공간 우위	기존 SSD와 동일하게 OP 존재

FDP: Eco System

FDP enablement easy to integrate in existing storage stacks

- Hint-based, Implicit (no write errors), 하위 호환성 지원

Upstream Linux I/O Paths



Ecosystem Ready

Support in Linux Kernel through I/O Passthru

- User-space에서 직접 커널 내 NVMe 장치 사용
- SPDK와 유사하게, PCIe 장치를 커널에서 분리할 필요 없음
- Mainline 5.13 부터 upstream 진행
- hyperscalers와 enterprise에서 채택 증가
- io_uring을 통한 Scalability 및 Block I/O 대비 높은 성능

Application enablement through xNVMe

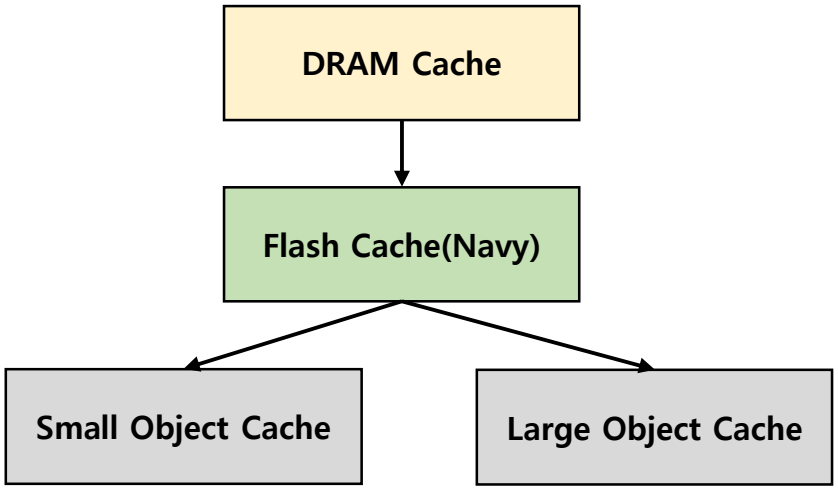
- Application 통합을 위해 서로 다른 I/O Path 지원 (block, I/O Passthru, SPDK)
- SPDK bdev를 사용할 수 있도록 SPDK (native & xNVMe)에 upstream 지원
- Samsung 지원, 커뮤니티 주도
- Cachelib: 커뮤니티와 개발 및 upstream 진행
- RocksDB: POC 실행

Tools

- nvme-cli와 fio 지원
- Emulation QEMU v8.0 이후 지원

FDP: Eco System (Cachelib)

Cachelib with FDP



Hybrid Cache in CacheLib

Reference: "The CacheLib Caching Engine: Design and Experiences at Scale."(USENIX 2020)

- 대규모 Web 서비스를 제공하는 업체는 Caching System을 사용하여 높은 성능과 효율을 달성
- Flash cache는 2KB 미만의 BigHash(Small Object Cache)와 2KB 이상의 BlockCache(Large Object Cache)로 Logical 하게 분리,일반적으로 mix되어 physical하게 같은 flash block에 존재
- SW을 주로 사용하는 BlockCache와 Small RW가 많은 BigHash는 I/O 패턴과 Lifetime이 다름
- 서로 다른 Lifetime을 가진 Data들이 섞여 WAF가 증가하고 GC로 인한 Overhead가 증가
- 두 Cache의 Data를 분리하기 위해 FDP를 사용하고, 서로 다른 RUH를 사용

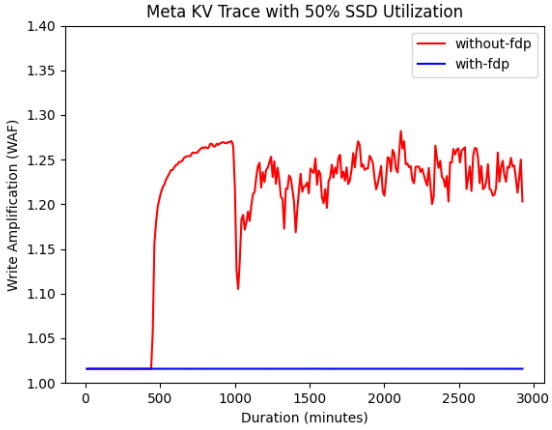
FDP: Eco System (Cachelib)

Initial results show a clear benefit without major Cachelib optimizations

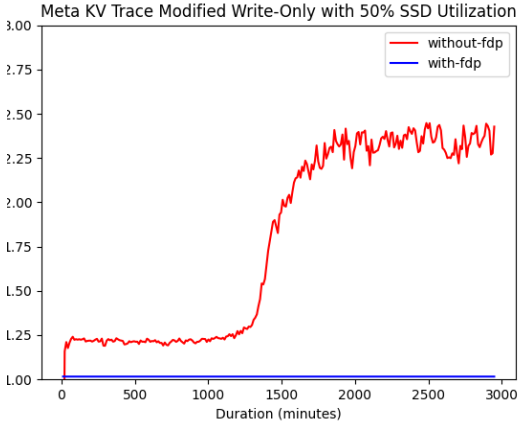
- Hyperscaler의 실제 워크로드에서 WAF 감소 확인
- Utilization 증가 시에도 WAF 유지
- Cache hit-rate에 영향 없음

Next steps

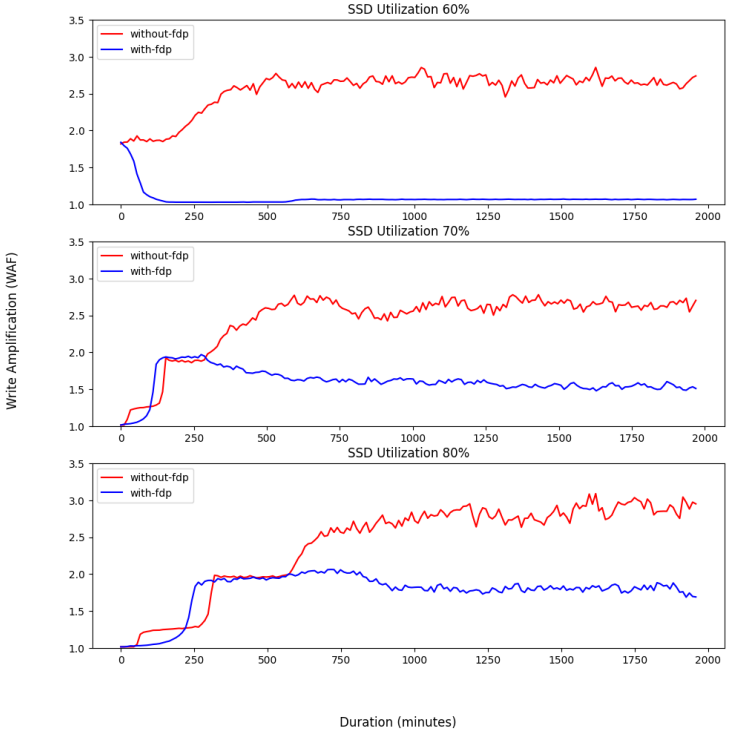
- Cachelib 지원을 위해 upstream (maintainers와 협업)
- 더 나은 데이터 분리를 위해 최적화 진행
- 다른 고객이 허용하는 WAF 수준 확인



1. Raw Meta read-heavy workload.



2. Altered Meta write-heavy workload

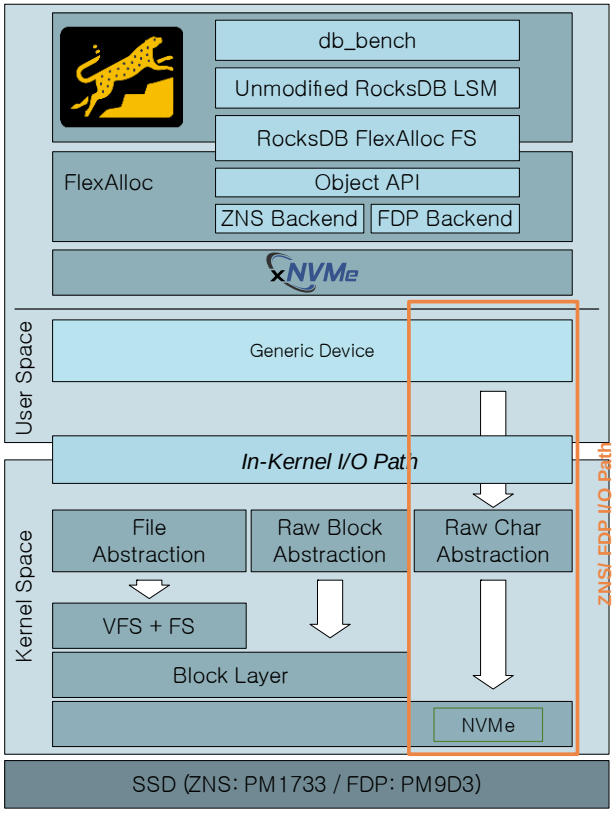


3. Increased utilization on Meta write-heavy workload

FDP: Eco System (xNVMe)

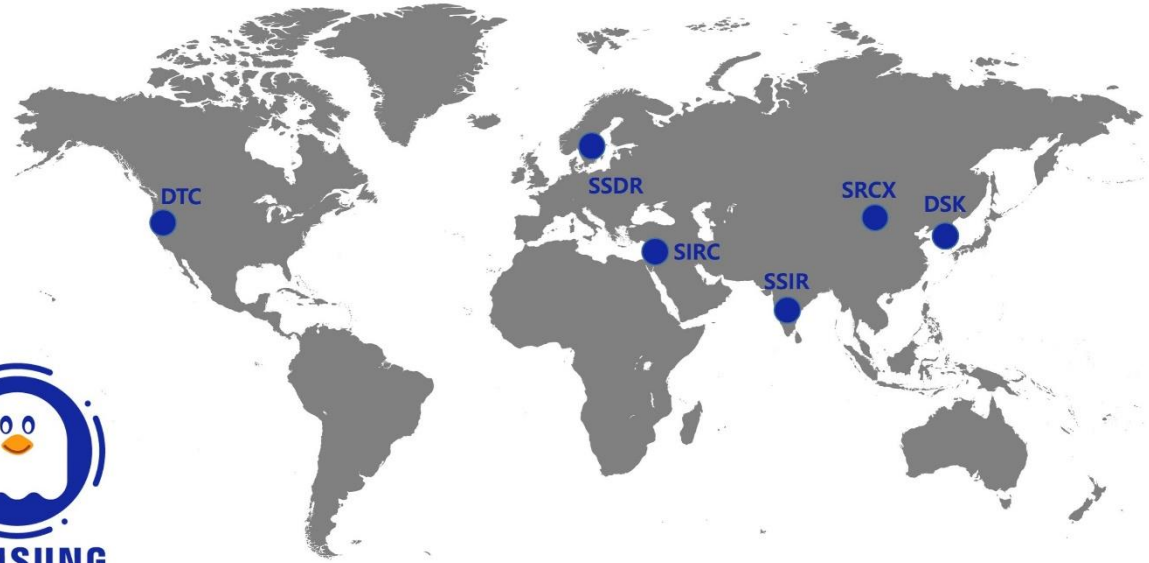
- **xNVMe: Library to enable non-block applications across I/O backends**
  - Common API 지원
  - psync, libaio, io_uring 지원 (Linux Kernel path)
  - SPDK: Linux와 FreeBSD User-space 지원
  - 향후 새로운 Backend 필요

- **Benefits**
  - I/O backend 및 OS 전반에 걸쳐 적용하기 용이
  - 신규 I/O 인터페이스에 적용이 용이
  - Raw device에서 file system적인 의미
  - 성능 영향 없음



Global Open Source Team

Distributed team across all Samsung Memory sites around the world



SAMSUNG

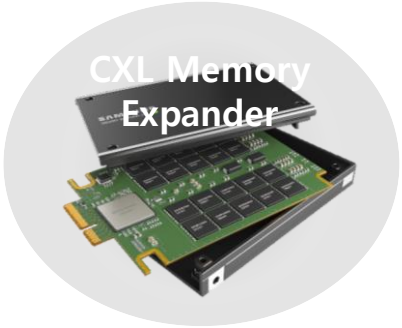
- Thanks to GOST Team Members for reference materials

Samsung Memory Research Center

- 외부 고객/파트너들과 협업하기 위한 인프라 운영

- SMRC는…
  삼성 메모리와 고객, IT 시스템 파트너가 함께 최적의 솔루션을 찾을 수 있는 협업 공간

■ 향후 메모리 솔루션 제품을 위해 고객 및 파트너와 SMRC에서 협업 가능



CXL Memory Expander

CXL 메모리 서브 프로토콜을 지원하는 CXL Type 3 장치로, 데이터 전송을 위해 PCI-Express 버스를 사용하는 바이트 어드레싱 가능한 장치로 사용

사용처 : In- Memory DB, EDA, Memory 가상화

파트너 : 구글 클라우드, SAP, VMWare, Red Hat

High-function SSD

Memory Semantic SSD: 표준 블록 장치로 다루어지는 대신 CXL을 활용

스마트 SSD : 자체 처리 기술을 탑재해 CPU 활용도를 낮추고 데이터 전송 병목 현상을 감소

사용처 : Greenplum DB & vSphere8 ref. architecture 등

파트너 : 구글 클라우드, Weka VMWare, Red Hat, Supermicro

QLC based SSD (ZNS, FDP)

ZNS(Zoned Namespaces)는 NVMe™에 새로운 Command Set으로, 호스트와 SSD 사이의 Zoned Block Storage Interface를 노출하여 데이터를 자신의 미디어에 완벽하게 정렬 가능

사용처: Object storage , Cold storage

파트너 : Red Hat, NetApp, Weka.io

Ultra High Density Storage

페타바이트 스케일 스토리지를 위한 새로운 스토리지 솔루션으로 분리된 스토리지 인프라를 위한 rack-scale 공간 효율성 증가

사용처 : 빅데이터, AI 교육

파트너 : Red Hat, Weka.io, Lightbits, Supermicro

THE NEXT CREATION STARTS HERE

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