Characterizing and Emulating FDP SSD with WARP

Inho Song, *Shoaib Asif Qazi*, *Javier Gonzalez**, *Matias Bjørling***, *Sam H. Noh*, *Huaicheng Li*

VIRGINIA TECH

* SAMSUNG

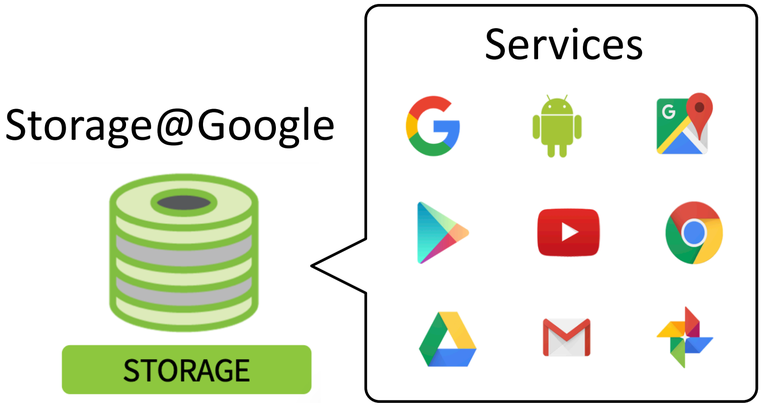
** Western Digital

I. Introduction

- **Emerging storage: Flexible Data Placement (FDP) SSDs**
  - FDP Cloud service providers (Google and Meta), NVMe specification support, and manufacturers (Samsung)

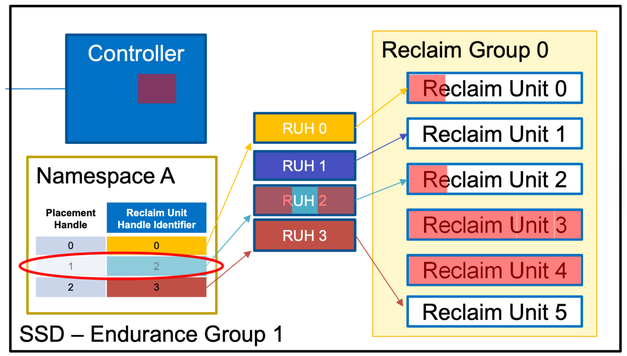
Storage ecosystems favor what FDP SSD can offer :

"Transparent interface, low write amplification, but easy-to-use!"



Google's path to SmartFTL SSD

(2021, Global OCP Summit)



NVMe TP4146 FDP Specification

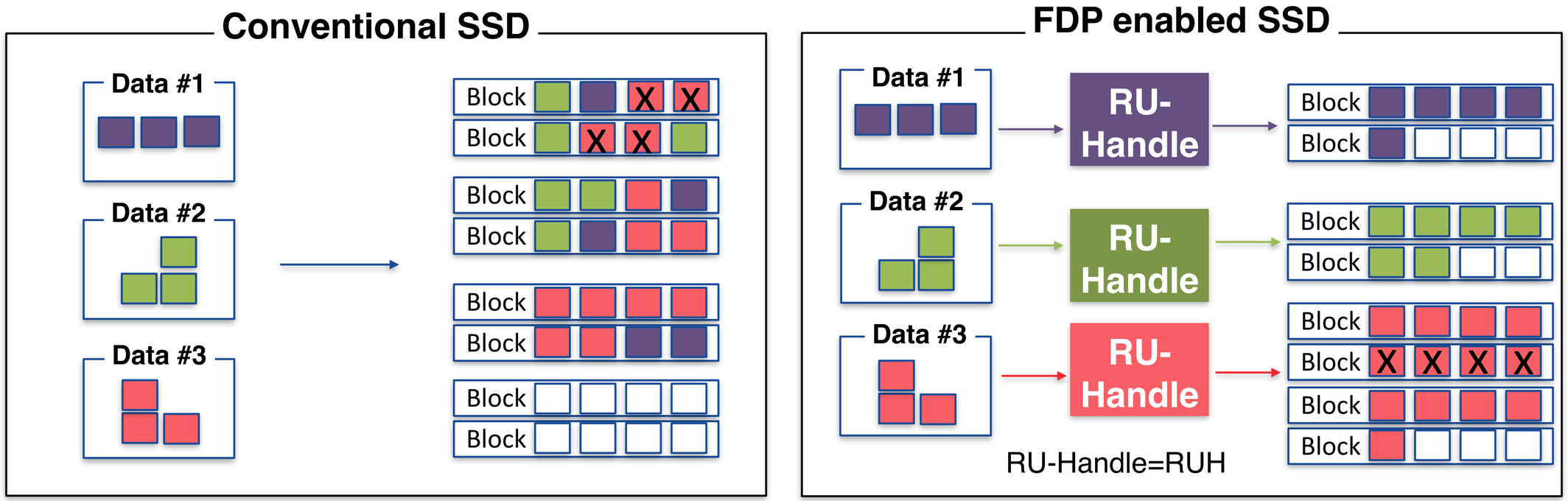
(2023, Hyperscale innovation: FDP , state of the union, NVM express)

Flexible Data Placement(FDP) enabled SSD

(Samsung PM9D3a)

II. Background

- **Conventional SSD vs FDP SSD**



II. Background

- **Terminology in FDP SSD**
  - **RUH: Reclaim Unit Handle**
    - Specified in the NVMe request field
    - Equivalent to stream-id in Multi-stream SSD
  - **RU: Reclaim Unit**
    - Typically, RU = NAND superblock
    - Equivalent to zone in ZNS SSD

Very quick summary with ZNS

- RUH = zone id
- RU = zone

III. Motivation

- **Blackbox FDP : Need for transparent platform for FDP**

Blackbox

User perspective

- (How effective is FDP to lower WAF for production workload?)

Google Meta

Device manufacturer

- (How to design FDP capable SSD controller for lowest WAF?)

SAMSUNG

Three questions we want to address

1. *When does FDP deliver near-1 WAF, and when does it fail?*

2. *Which vendor-level configurations drive these differences?*

3. *What are the design tradeoffs in FDP?*

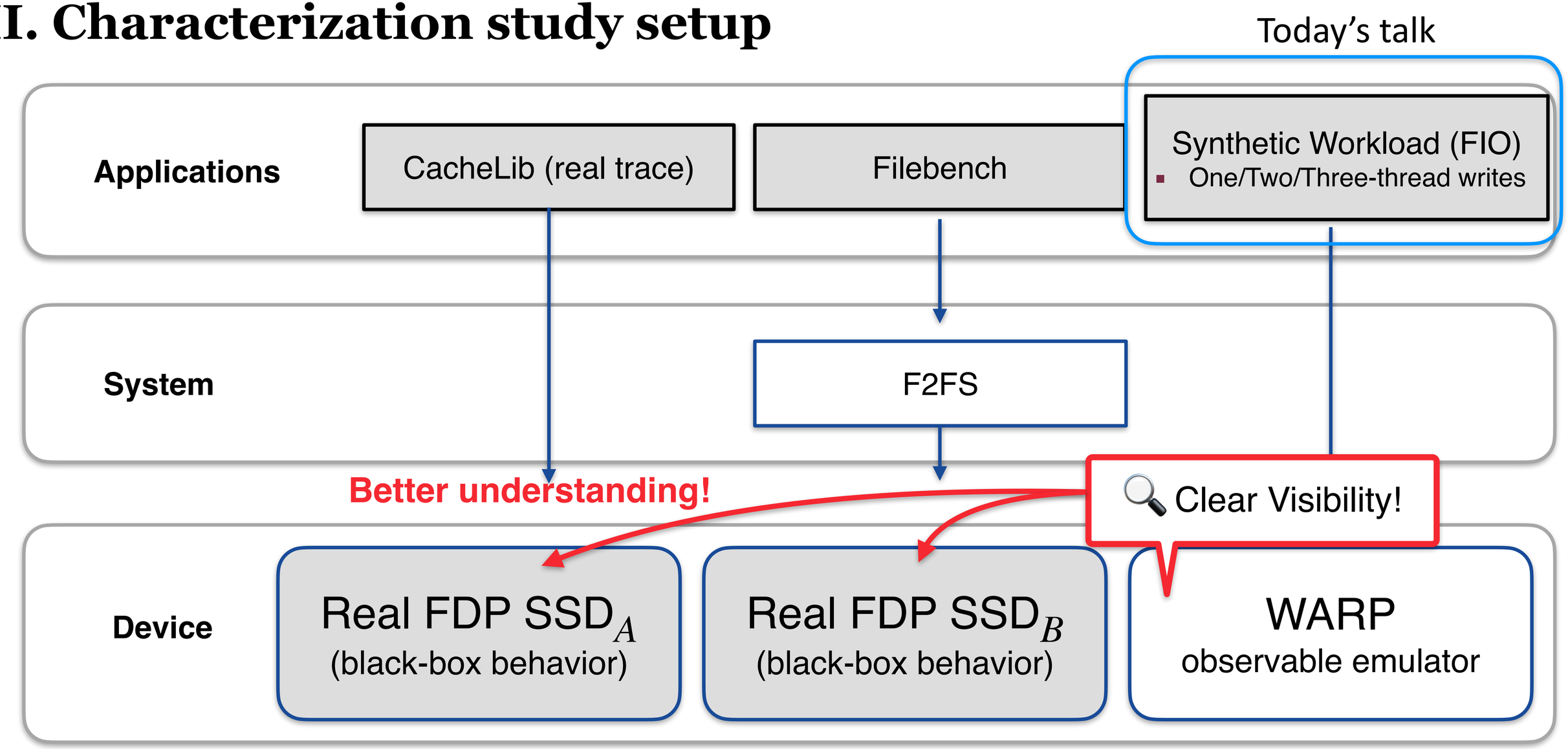
Our Contributions

- We conduct the first systematic study of commercial FDP SSDs, revealing when FDP sustains near-1.0 WAF and when it collapses.
- We identify two previously unreported behaviors that explain how RUH interference and premature reclamation erode FDP's benefits.
- We design and validate WARP(Write Amplification Research Platform), the first open FDP emulator that reproduces hardware trends while exposing GC victim choices and tunable geometry.
- With WARP, we further explore firmware strategies that are hidden in current hardware.

Outline

- Introduction
- Motivation
- **FDP Characterization Study**
- **Design and Implementation of WARP and Evaluation**
- **Findings from WARP**
- **Conclusion**

III. Characterization study setup



We conduct the first systematic study of commercial FDP SSDs across synthetic, trace-driven, and file-system workloads

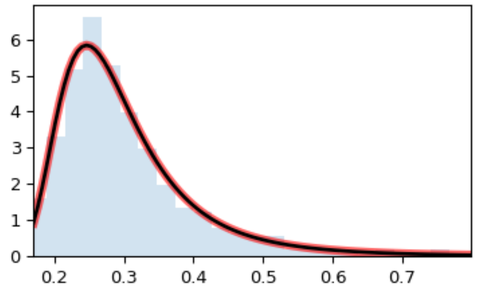
III. Characteristics of real FDP SSDs

Synthetic workload design (FIO)

1. **[Three write threads] : Sequential + Random + Overwrite** ✅*Today's scope*

2. random 1. Sequential thread

Invalidate Distribution

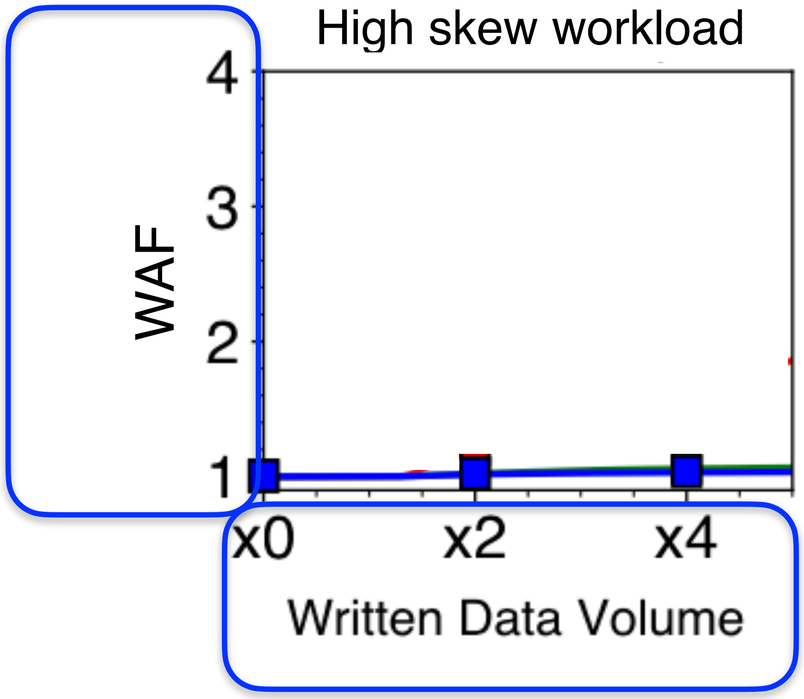


- High skew (zipf 2.2)
- Medium skew (zipf 1.2)
- Low skew (80/20)

3. Overwrite thread (data temperature)

III. Characteristics of real FDP SSDs

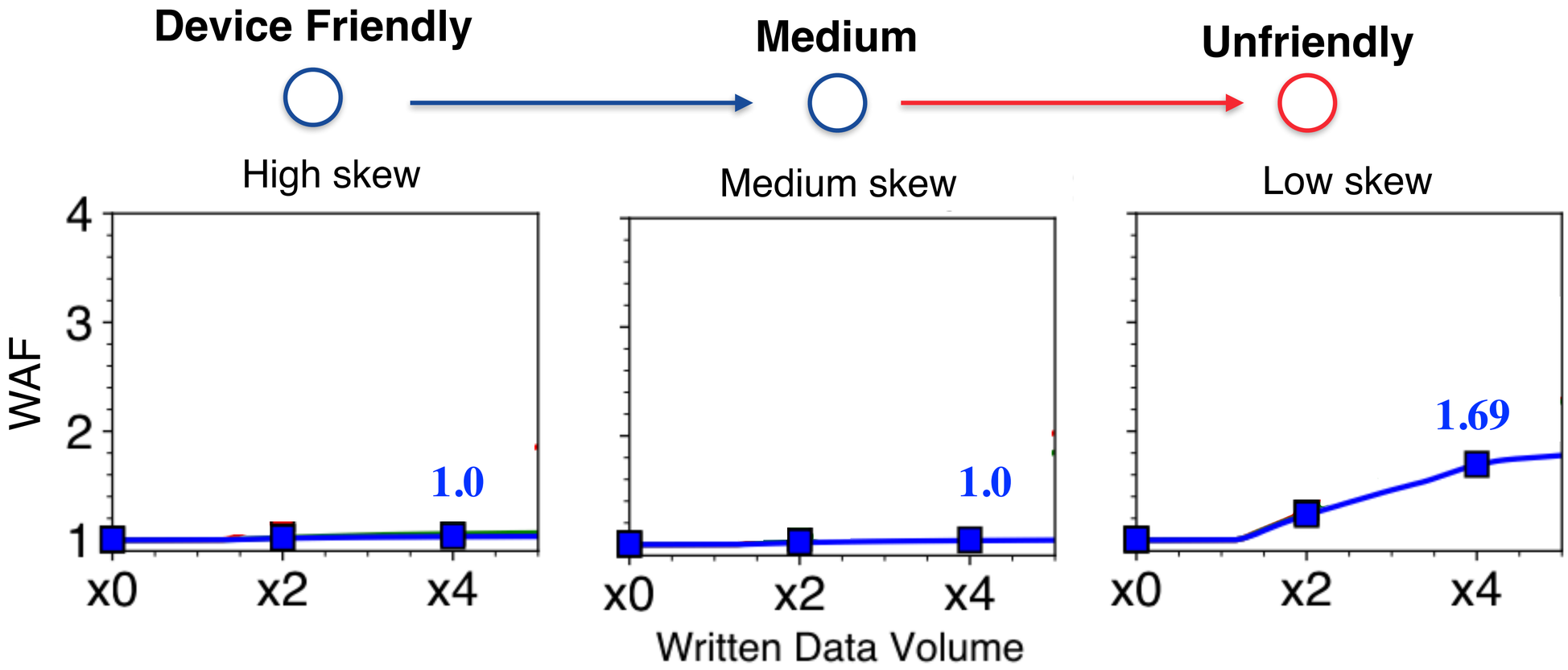
How to read this figure:



- **Y-axis**: Write amplification factor(WAF). Lower is better.
- **X-axis**: Workload size, which is a relative number to device capacity(IO size = x2 capacity, etc)

III. Characteristics of real FDP SSDs

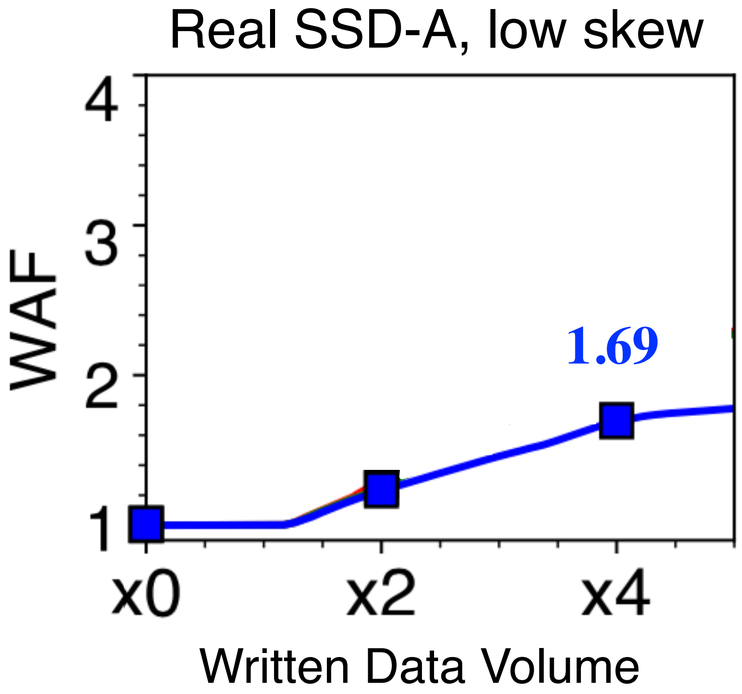
FDP benefit depends on …

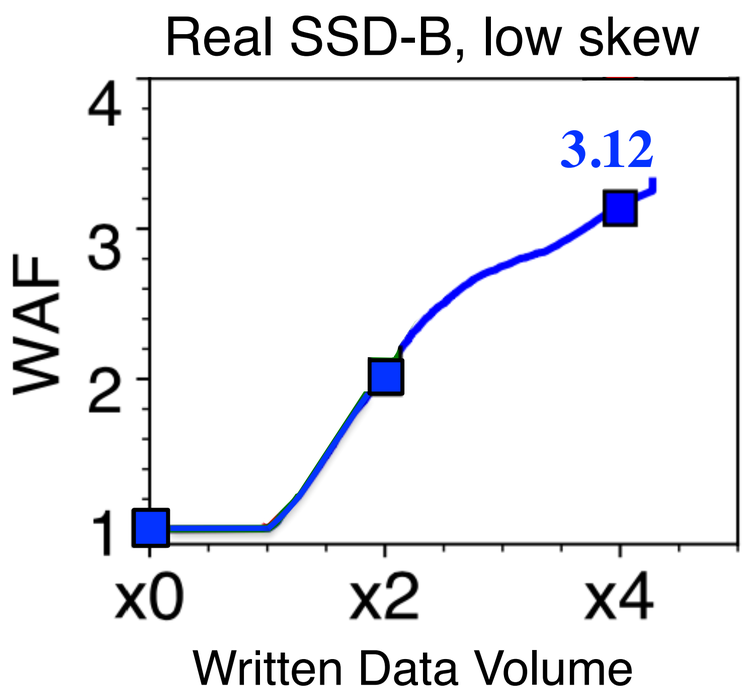


1) Workload skewness

III. Characteristics of real FDP SSDs

FDP benefit depends on …





2) Firmware design

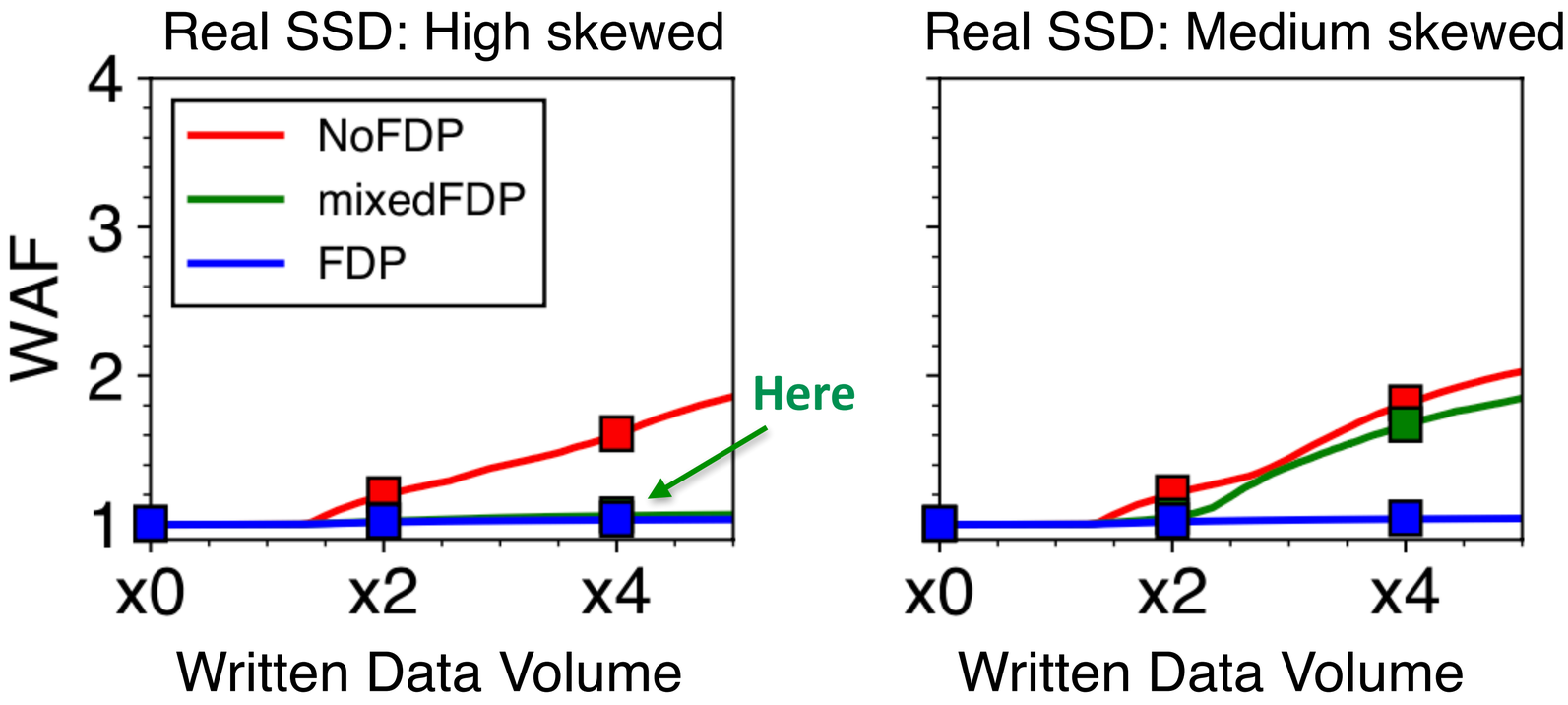
III. Characteristics of real FDP SSDs

NoFDP :
mixure of all three data

mixedFDP :
mixure of sequential overwrite into a single RUH

FDP :
Separates all three data

FDP benefit depends on …



Together, FDP is quite *fragile*, and easy to fail!

WAF in FDP heavily depends on
1) Workload skewness, 2) Firmware design, and **3) Host classification**

IV. WARP, FDP SSD emulator

Commercial FDP SSDs

Blackbox

- Lack of visibility
- Vendor dependent
- Hard to reason

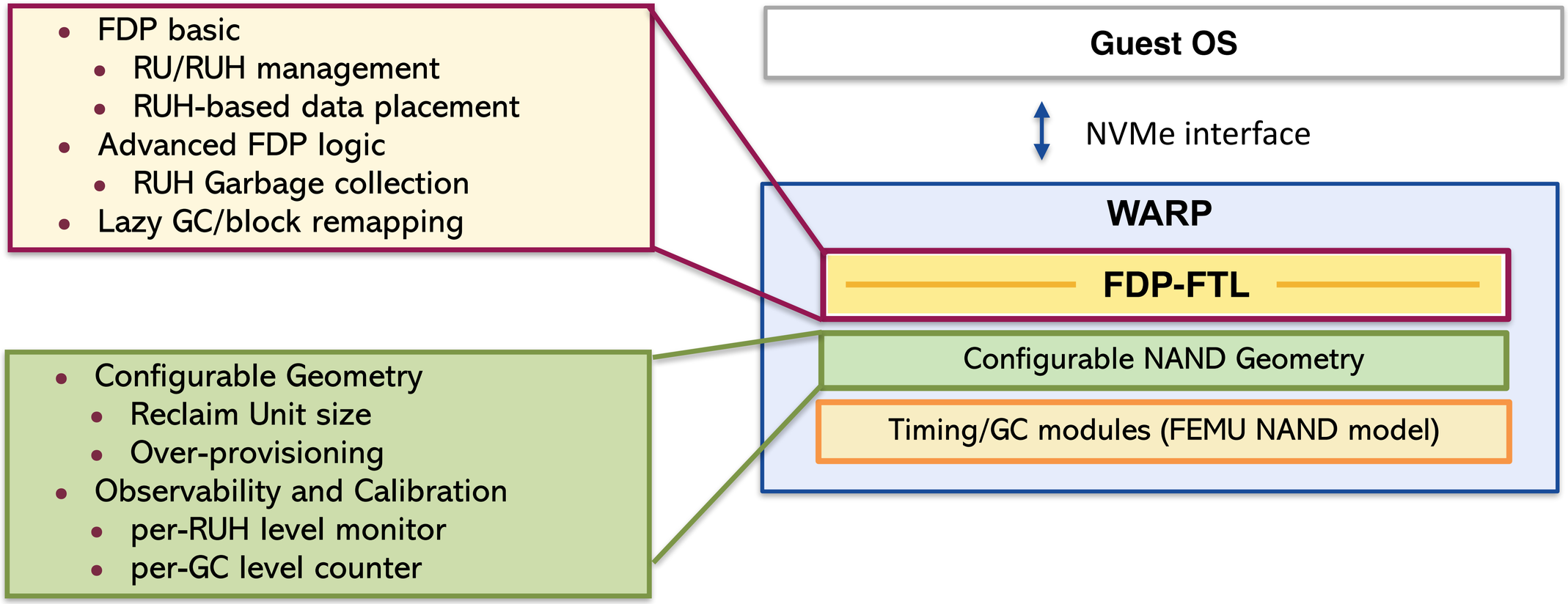
Open FDP SSD Emulator

Whitebox

- Great visibility ✅
- Vendor agnostic ✅
- Great for reasoning ✅

IV. WARP, FDP SSD emulator

- **WARP: FEMU-based FDP SSD emulation**

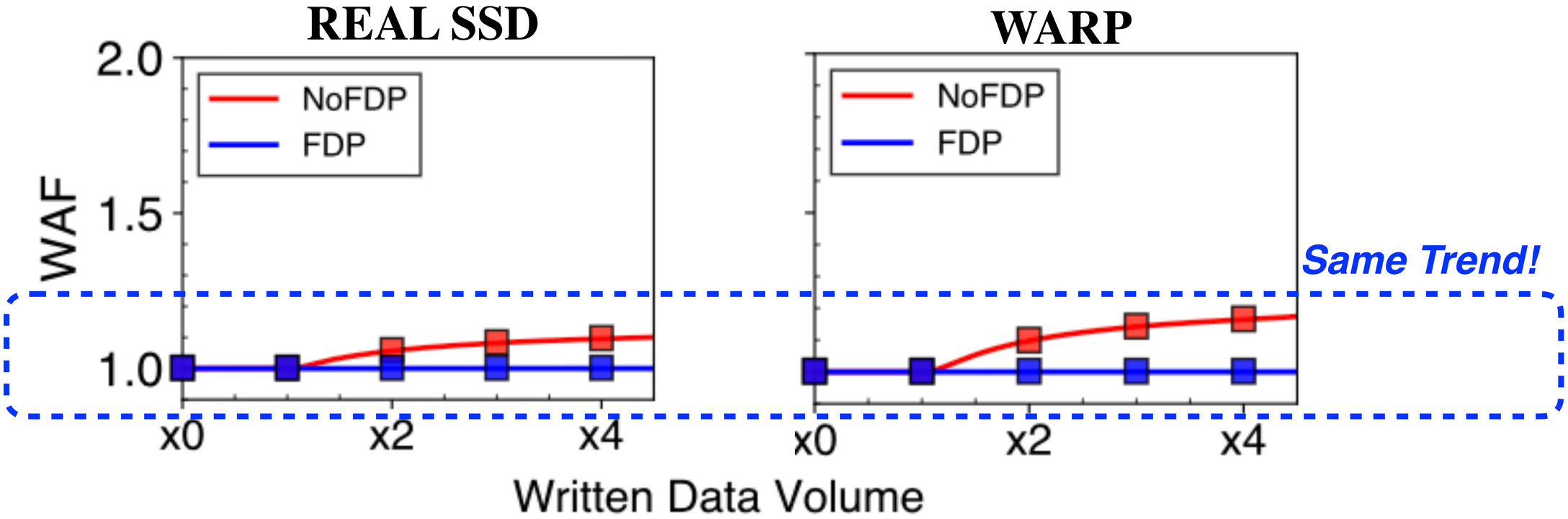


Outline

- Introduction
- Motivation
- FDP Characterization Study
- Design and Implementation of WARP
- **Evaluation**
- **Findings from WARP**
- **Conclusion**

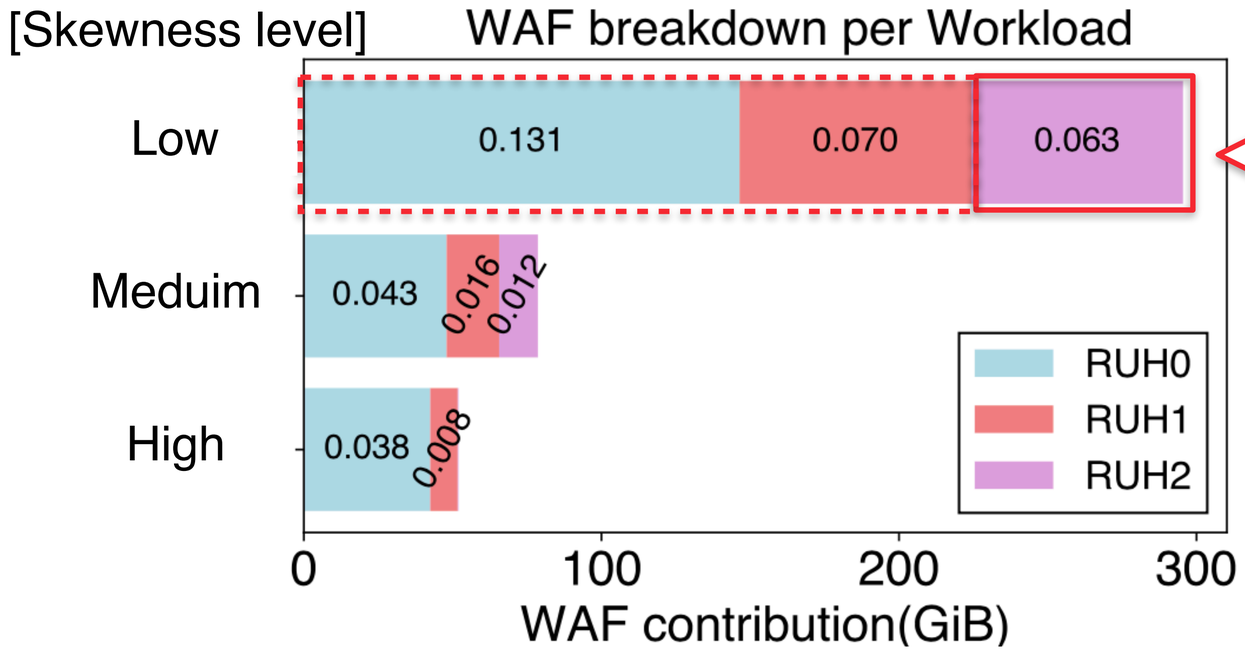
V. Evaluation

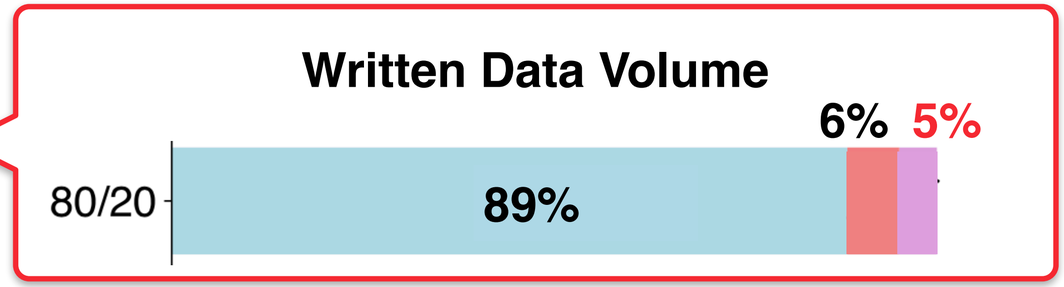
- **Evaluate with real FDP SSDs**



WARP reproduces the real FDP SSD's WAF trend in both synthetic and real application workloads

VI. WARP Findings

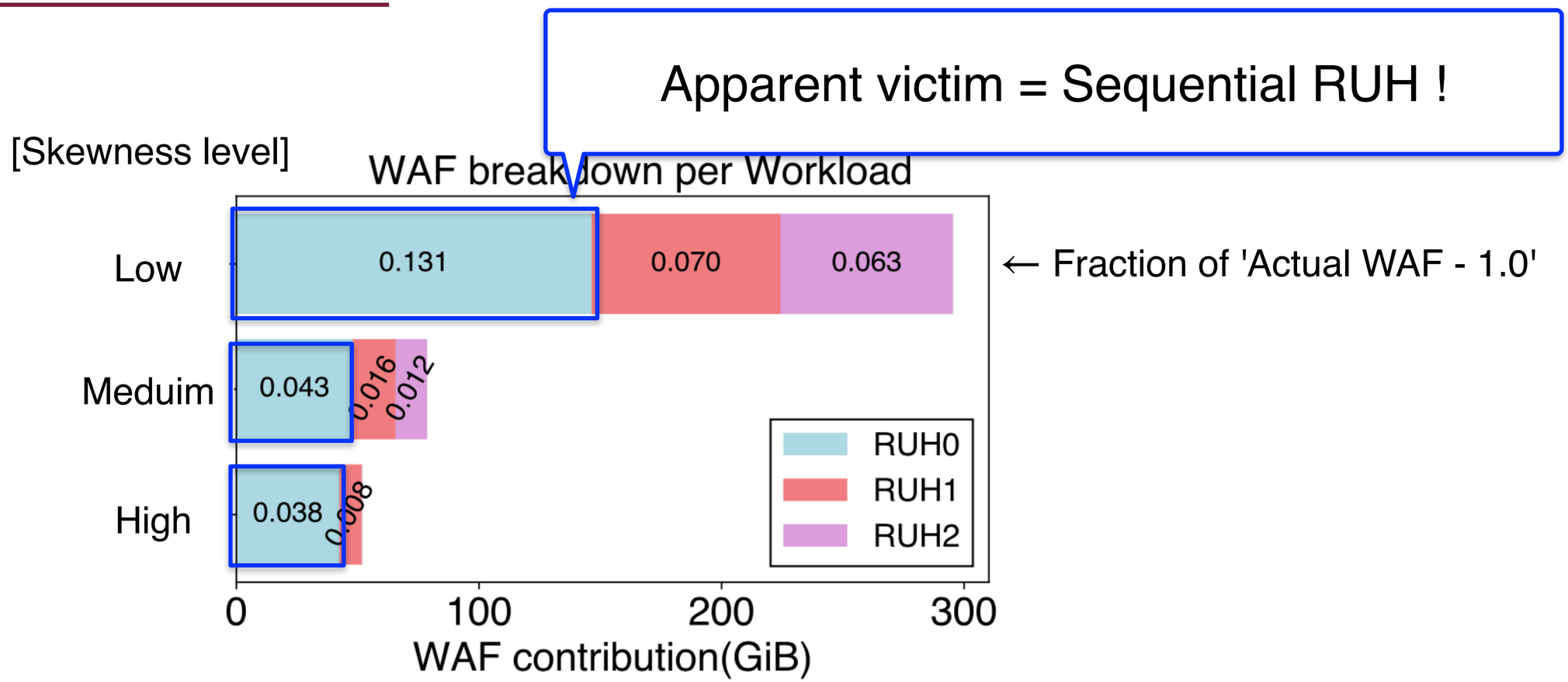




Noisy RUH

Even small portions (~5%) can contribute to ~25% of WAF.

VI. WARP Findings



Save sequential

Sequential data can get reclaimed prematurely

More in the paper

- **More Characterization study**
  - CacheLib + real traces
  - F2FS + filebench results

- **FDP SSD design internal**
  - FDP SSD placement policy tradeoff in over-provisioning tradeoff
  - WARP guided optimization : CacheLib

VII. Conclusion

- **Q1. When does FDP deliver near-1 WAF, and when does it fail?**

When does FDP fail?

  - *1) Vendor & 2) Workload, and 3) Misclassification*

- **Q2. What platform can we test and evaluate?**

Write **A**mplification **R**esearch **P**latform: Open FDP SSD emulator

Findings

  - **Noisy RUH** : One RUH can inflate global WAF
  - **Save Sequential** : reduce over-victimizing capacity-dominant RUH

WARP git

- **Q3. How is the FDP SSD intrinsic related to the FDP benefit?**

  - ***II vs PI OP tradeoff*** : Case of FDP SSD design tradeoff (Check our paper)

Characterizing and Emulating FDP SSD with WARP

Inho Song, *Shoaib Asif Qazi*, *Javier Gonzalez**, *Matias Bjørling***, *Sam H. Noh*, *Huaicheng Li*

VIRGINIA TECH

* SAMSUNG

** Western Digital