

# Characterizing and Emulating FDP SSDs with WARP

Inho Song, Shoaib Asif Qazi, Javier Gonzalez, Matias Bjorling, Sam H. Noh, Huaicheng Li  
Virginia Tech, Samsung Electronics, Western Digital

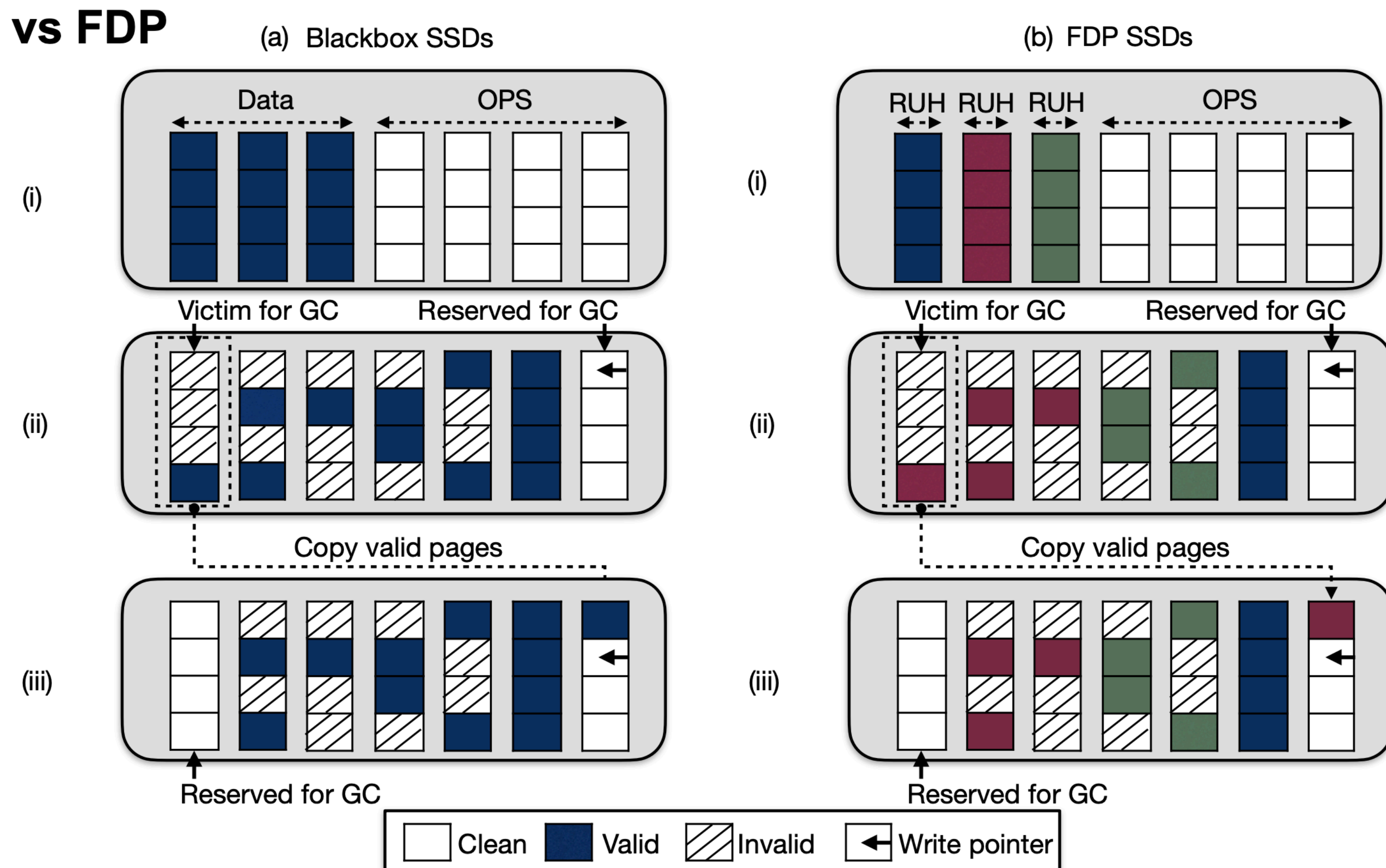


## Introduction to FDP & Key Characteristics

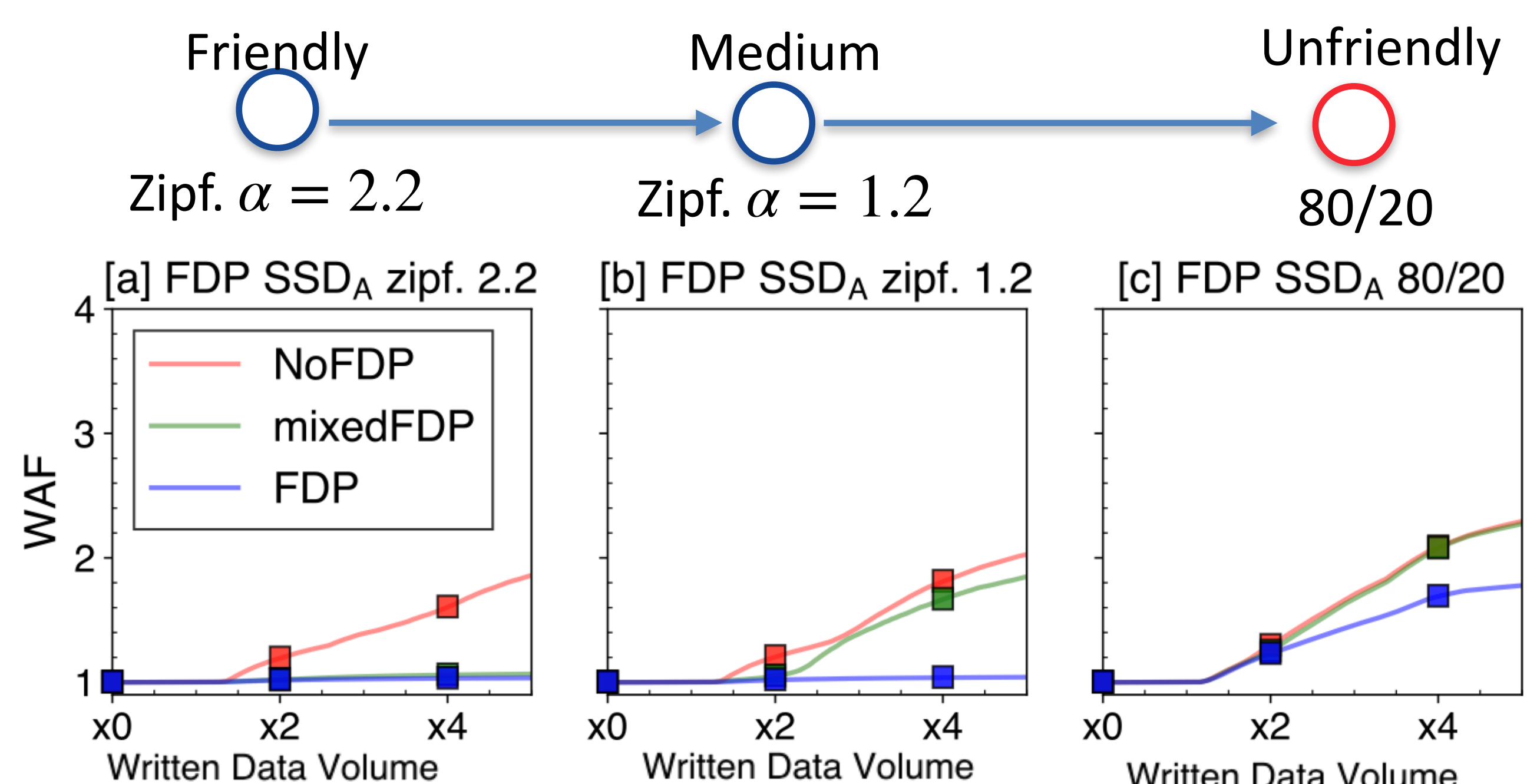
### What is Flexible Data Placement?

- Write amplification (WAF) is a fundamental challenge in flash storage. Garbage collection reclaims pages, causing excess writes that shorten SSD lifetime and inflate cost.
- FDP lets hosts steer writes into Reclaim Unit Handles (RUHs), isolating data by lifetime. When lifetimes align with RUH assignment, WAF approaches the ideal 1.0
- Ratified as NVMe TP4146. Adopted by Google, Meta, Samsung. Preserves backward-compatible block interface (unlike ZNS/OpenChannel).
- But FDP is a best-effort interface — GC remains device-managed and opaque. Vendor firmware choices are hidden from the host.

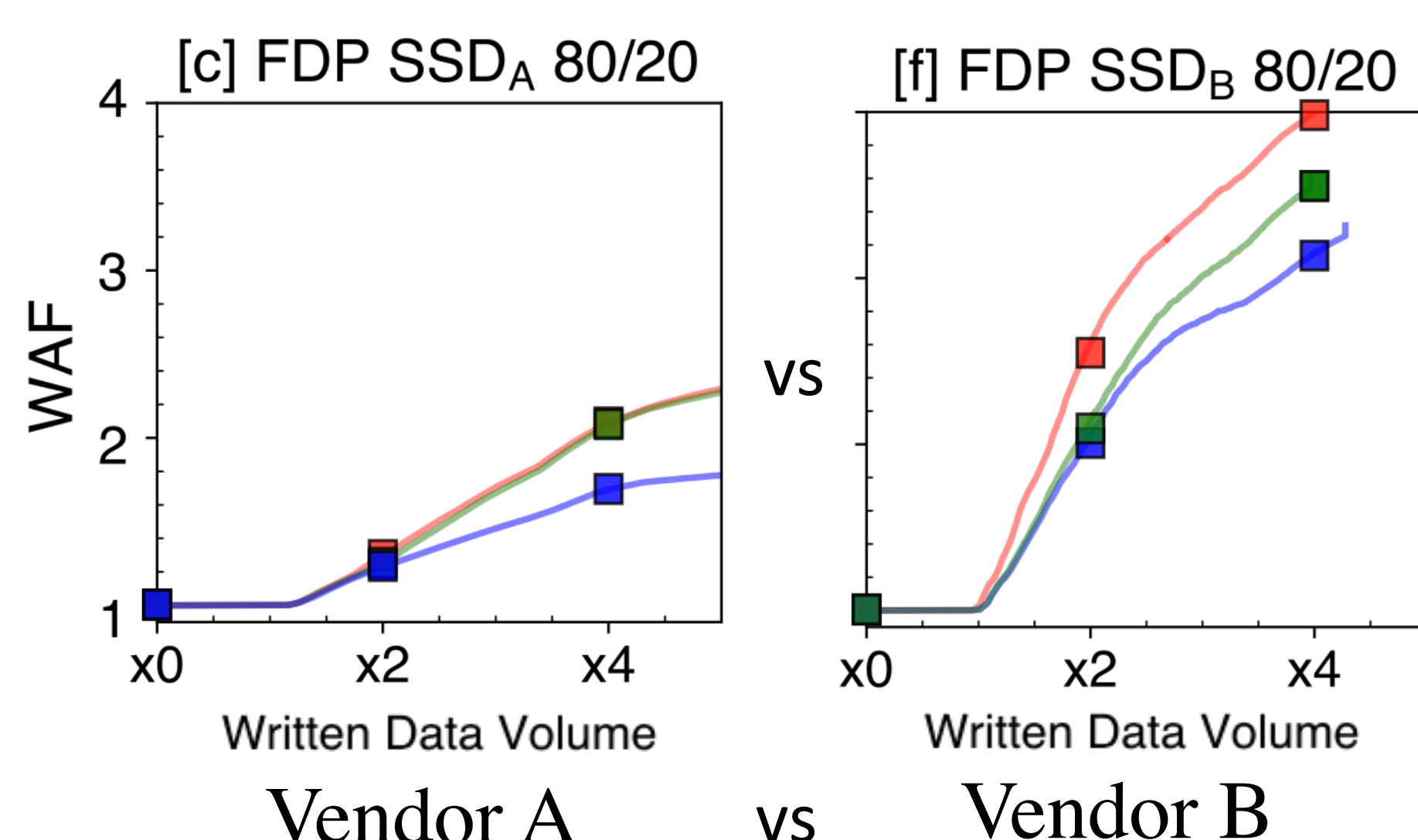
### BB vs FDP



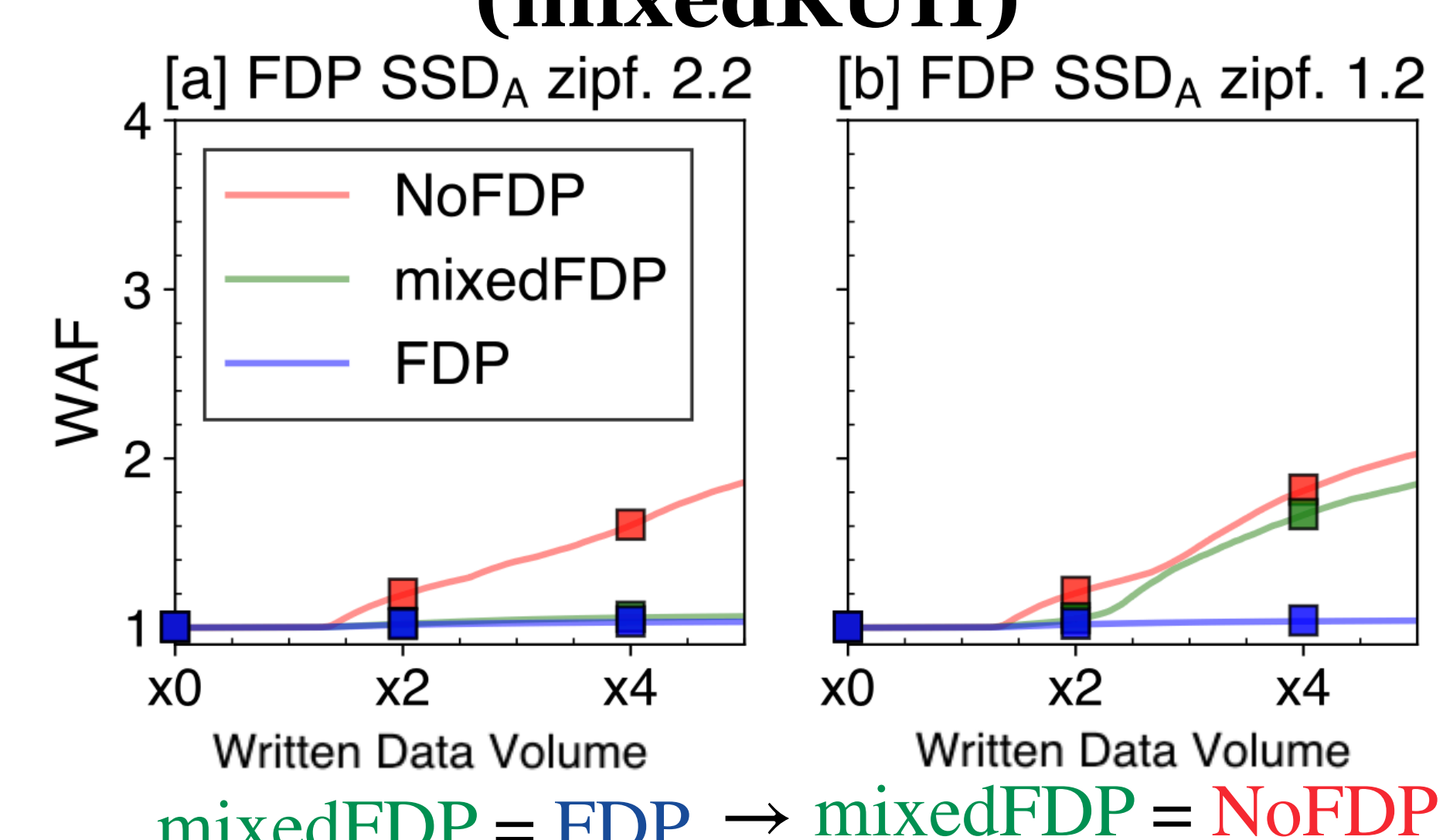
### 1. The workload skewness



### 2. The vendor design



### 3. The classification (mixedRUH)

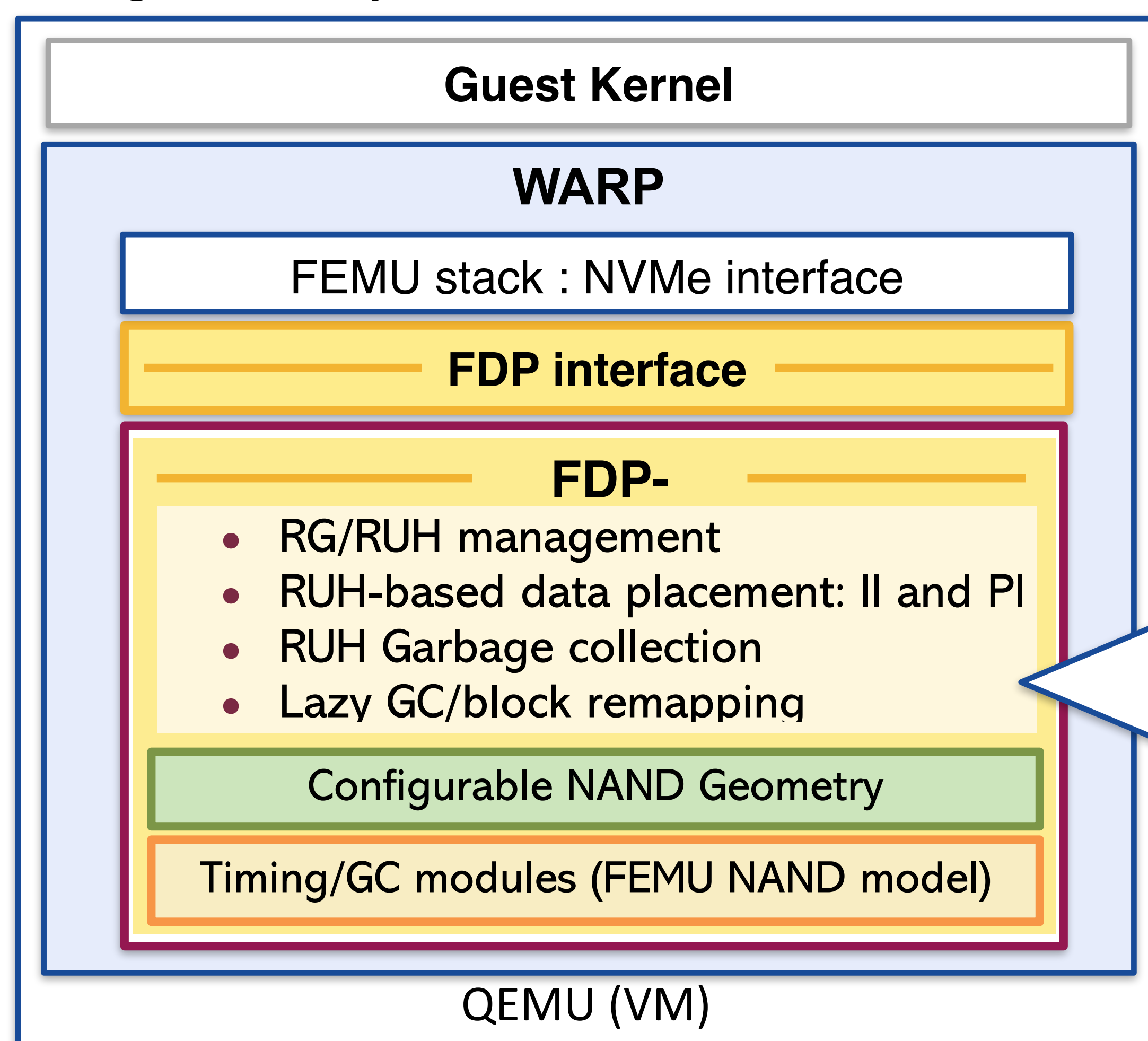


[ FDP is quite fragile, easy to fail! ]

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

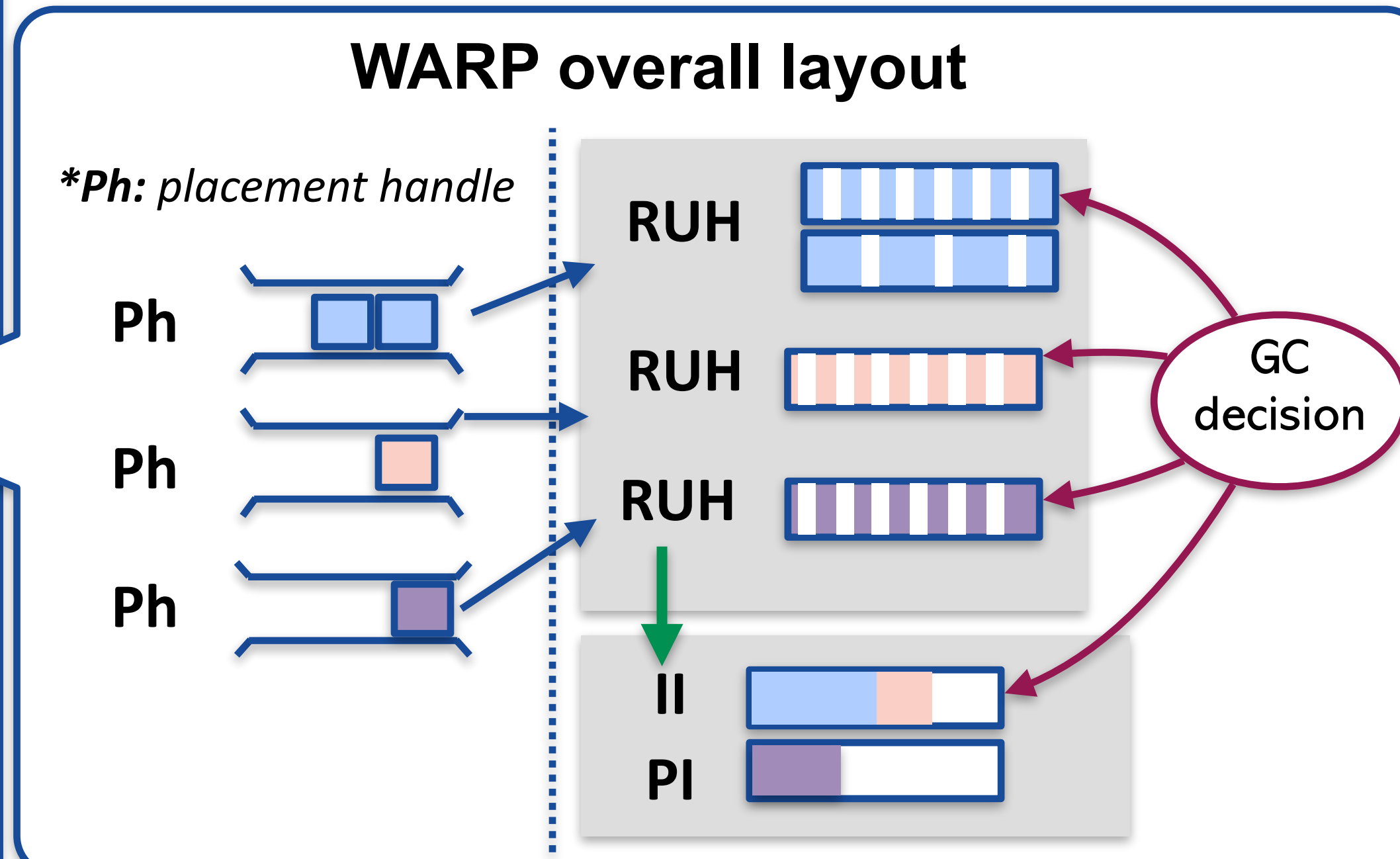
## WARP: A Versatile FDP Emulator Platform

### Design and Implementation of WARP



### Three Key Capabilities

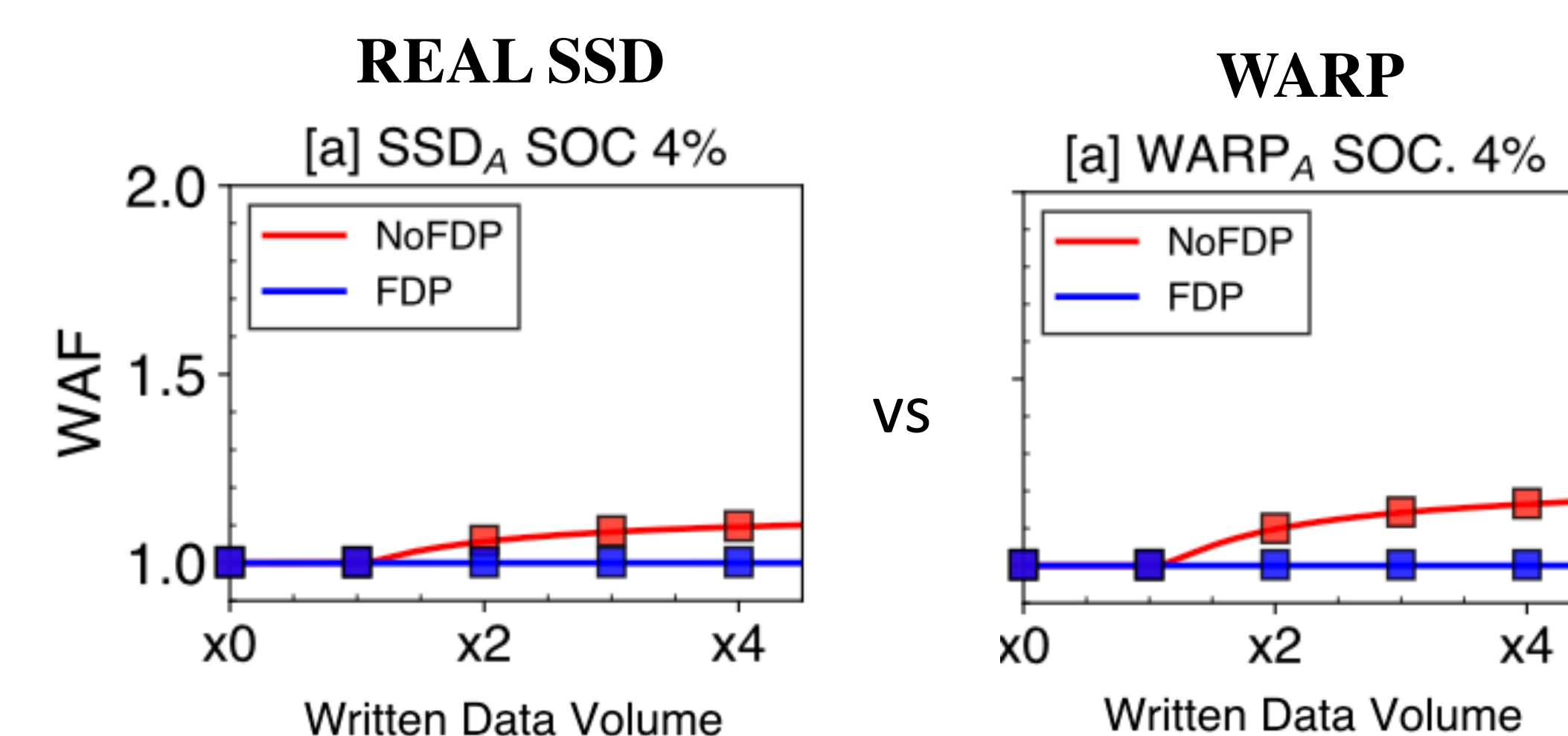
- Isolation Semantics**
  - Both Initially Isolated (II) and Persistently Isolated (PI) modes
  - II: GC copies go to shared GC-RUH | PI: strict per-RUH isolation after
- Configurable Geometry & GC**
  - Tunable RU size (128/256/512 MB), OP ratio (1–28%), RUH count
  - Greedy, cost-benefit, pressure-based victim selection; lazy GC, block remapping



### (3) Per-RUH Observability

- Per-RUH counters (host bytes, GC bytes, WAF) and structured event logs
- Exposes Noisy RUH and Save Sequential — invisible on real hardware

### [CacheLib kvcache workload]



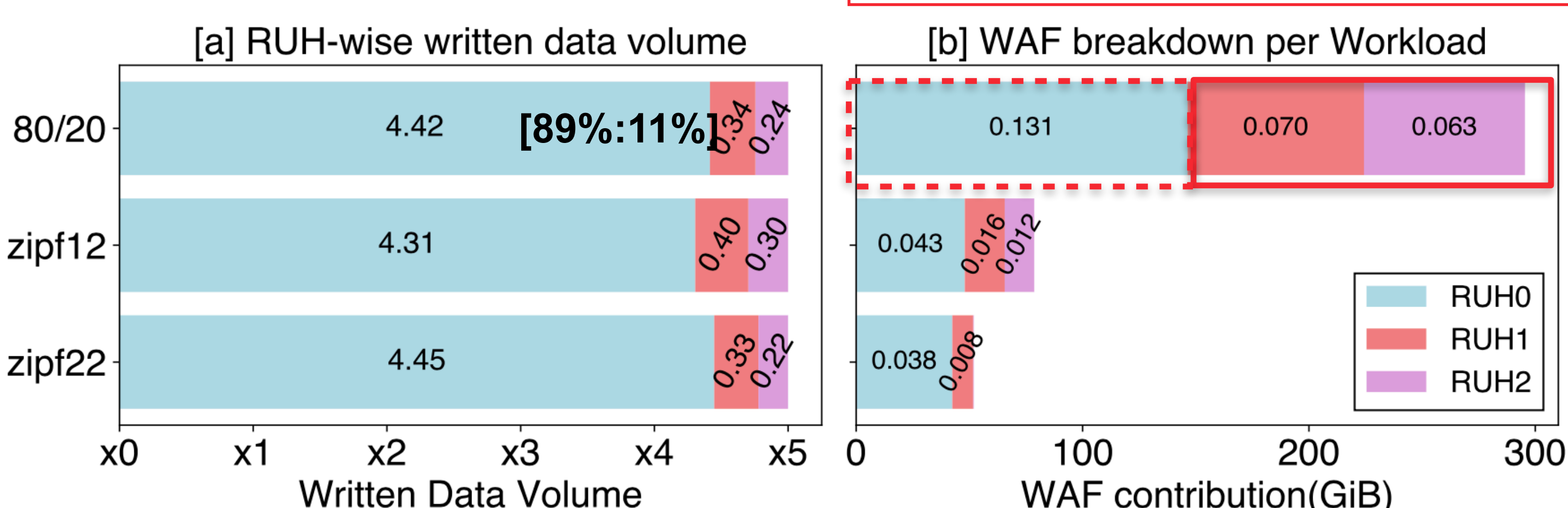
[ WARP vs Real FDP SSD evaluation ]  
THEY ARE SIMILAR!

## Key Findings from WARP

### Finding 1: Noisy RUH

One Handle Inflates Global WAF

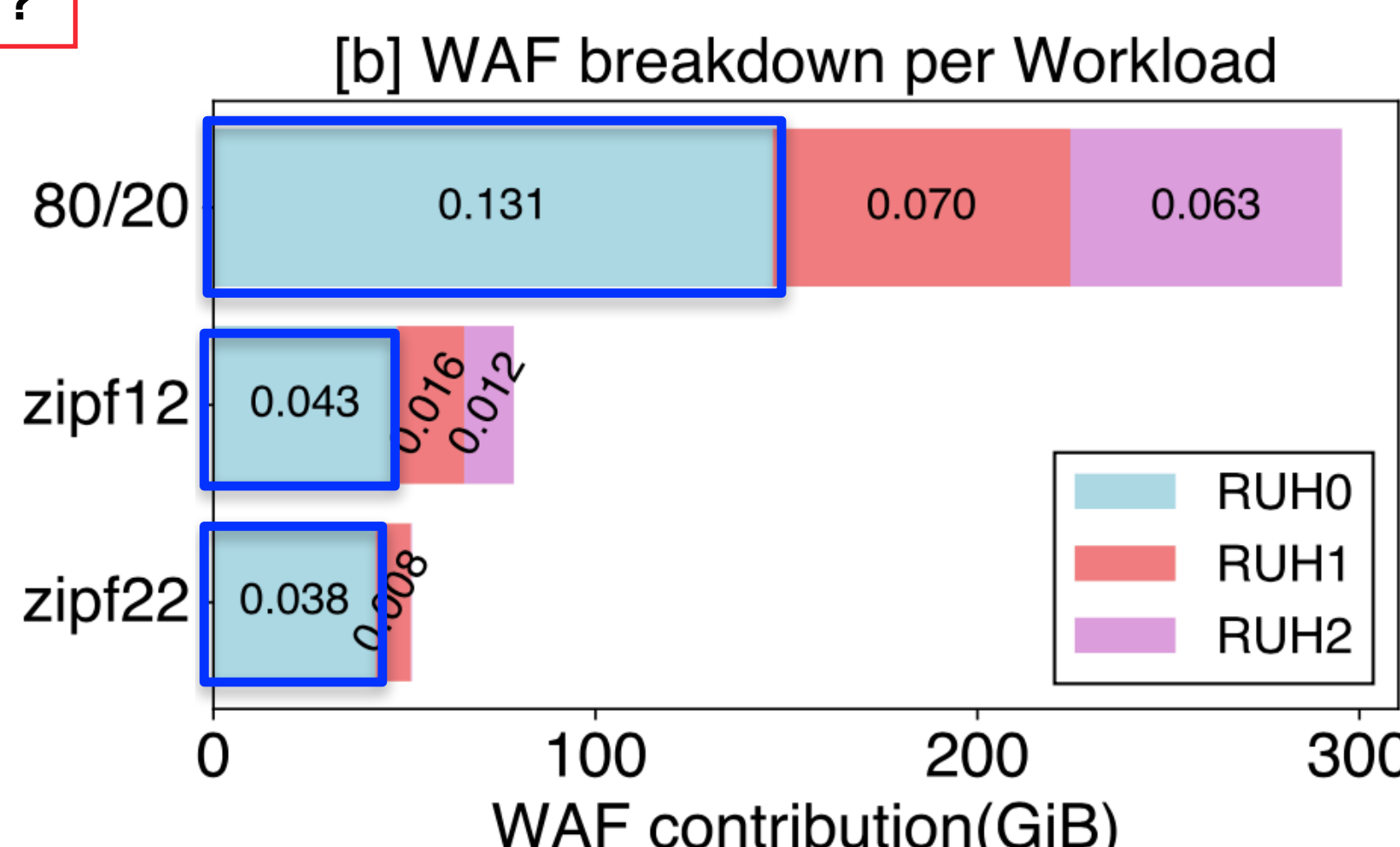
Small portions (~10%) contribute ~50% of WAF?



- A single RUH's invalidation stream forces GC across ALL handles, inflating global WAF.
- Under 80/20 workload: RUH 2 writes only 5–6% of data but contributes 26% of total WAF (0.131 GiB).
- Small portions (~10% of traffic) contribute ~50% of WAF.
- FDP isolation is not airtight — cross-RUH interference is a fundamental pitfall.

### Finding 2: Save Sequential

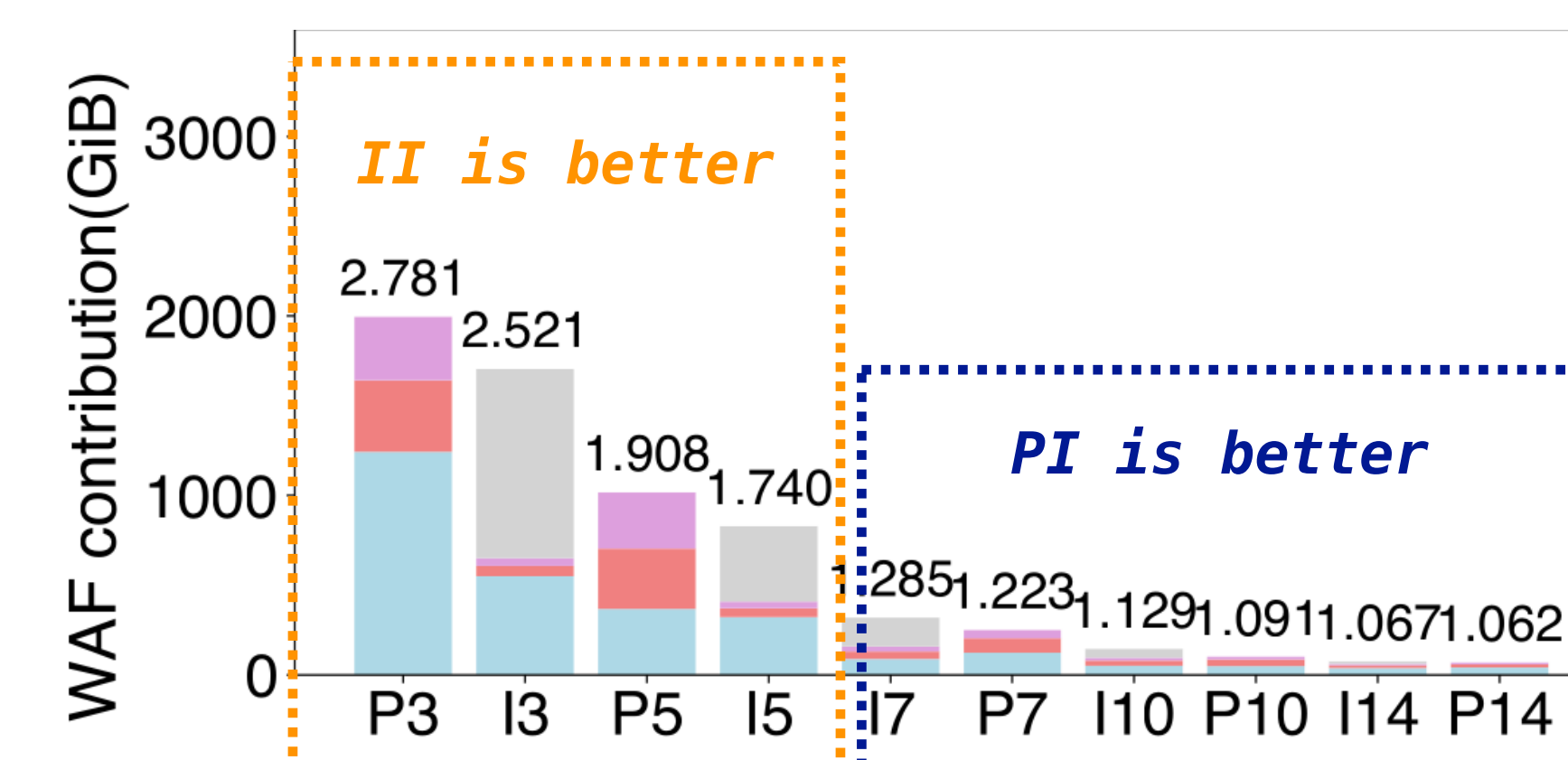
GC Victims Are Not Who You Think



- The apparent "victim" of WAF is the capacity-dominant RUH (sequential stream), not the noisy one.
- GC prematurely reclaims large sequential streams even though their data would self-invalidate soon.
- Even long sequential writes cannot be assumed safe when they collide with device-internal GC policy.
- This is a unique resource-sharing problem specific to FDP SSDs.

### Finding 3: II vs PI Tradeoff

Over-Provisioning Determines the Winner



- II wins when OP is scarce (<7%): pools spare space in a shared GC-RUH, absorbing GC pressure across streams.
- PI wins when OP is abundant (>10%): strict per-RUH isolation minimizes cross-stream interference.
- Crossover at 7–9% OP for RU256MB; shifts to 5–7% for RU128MB.
- II: robust with less host knowledge. PI: powerful but fragile under limited OP or heterogeneous workloads.