






# BCW: Buffer-Controlled Writes to HDDs for SSD-HDD Hybrid Storage Server

Shucheng Wang[1], Ziyi Lu[1], Qiang Cao[1], **Hong Jiang**[3],
Jie Yao[2], Yuanyuan Dong[4] and Puyuan Yang[4]

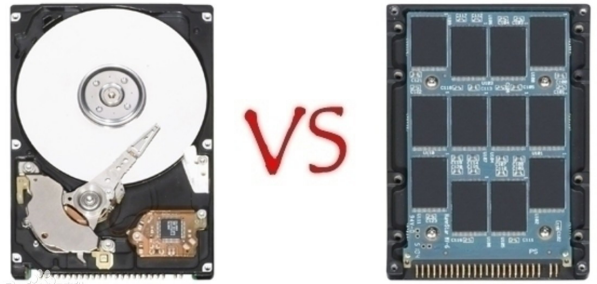
[1,2] Huazhong University of Science and Technology,
[3] University of Texas at Arlington,
[4] Alibaba Group

# Outline

- Background
  - SSD-HDD hybrid storage
- Challenge and Motivation
  - Unbalanced device utilization
  - HDD buffered write behavior
- Design
  - Buffer-Controlled Writes
  - Mixed IO scheduler
- Evaluations

# SSD-HDD hybrid storage

- SSD and HDD
  - SSD (Solid State Drive) has high-speed performance
  - HDD (Hard Disk Drive) has large capacity and low cost




- SSD-HDD hybrid storage
  - ➢ Applications

Microsoft Azure  Google Cloud  aws  Alibaba Cloud

  - ➢ Properties
    - ➢ **Cost-effectiveness**
    - ➢ **High-speed and low latency**

# Unbalanced Device Utilization

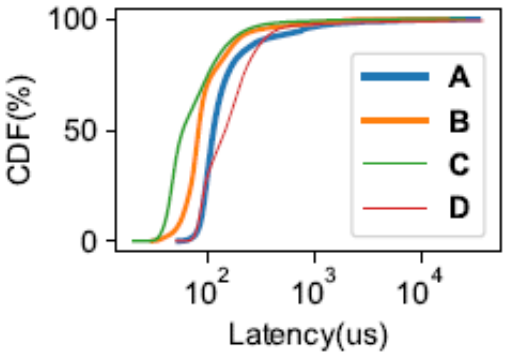
- SSDs suffer from **high write pressure**

**Highly-intensive writes in each SSD**

- Peak write requests per second $> 10KRPS$
- Data written per day $> 3\ TB$

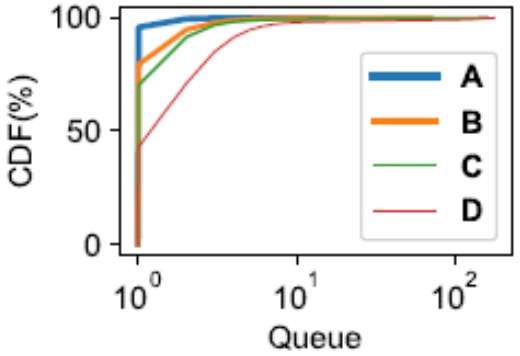
**Long tail latency**

- 99$^{th}$-percentile write latency $> 10ms$
- 99.9$^{th}$-percentile write latency $> 50ms$




**Large queue length**

- More than **100** blocked write requests in the queue

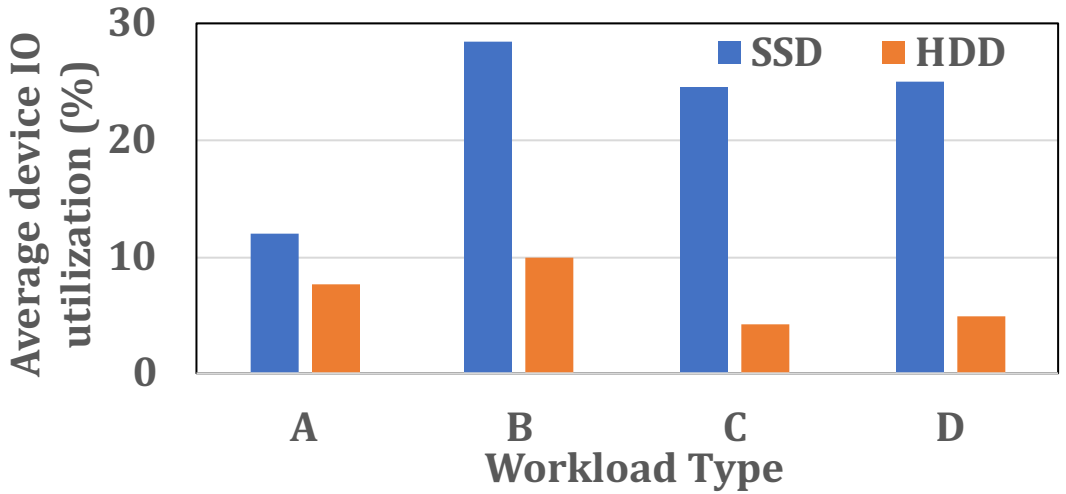


# Unbalanced Device Utilization

- HDDs are **underutilized**

**Low HDD utilization**

- **1/2 – 1/5** of SSD utilizations
- **90% – 95%** of times are in idle state



# Unbalanced Device Utilization

- SSDs suffer high writing pressure

Long tail latency
- 90th-percentile

Large queue length
- More than 100 blocked







We want to exploit the **underutilized HDD** to relieve the pressure of SSDs in hybrid storage nodes

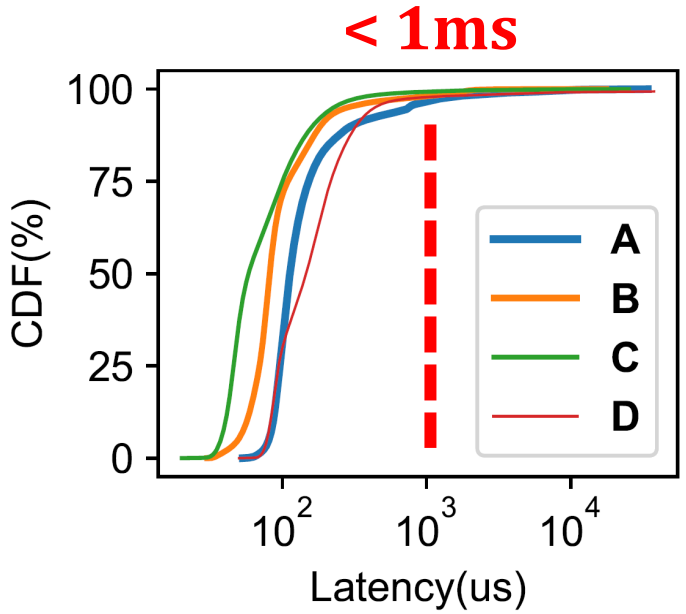
- HDDs are underutilized
  - **90% – 95%** of times are in idle state
  - **1/2 – 1/5** lower than SSDs

# Challenges




**Method is feasible when:**

$$Average\ Latency\ of\ HDD < Tail\ Latency\ of\ SSD$$




?

**Could the HDDs reach to *μs*-level write latency?**

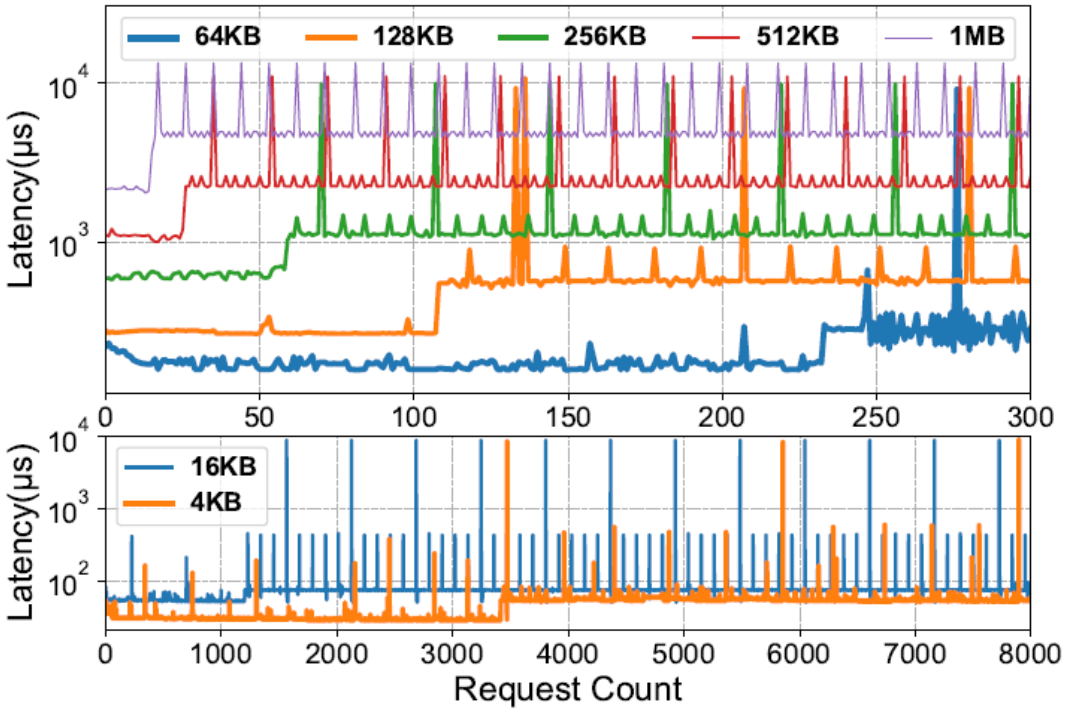
# Motivational Test

- Issue **sequential** and **continuous** writes to HDDs
  *(Append-only)* *(Close-loop)*

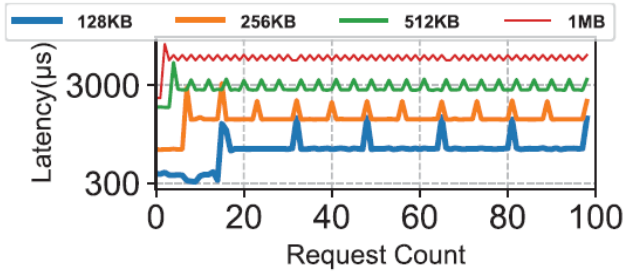
- Tested HDD Devices

| Manufacturer | Capacity | Model | Recording Technology | Sequential Write Bandwidth (MB/s) |
|---|---|---|---|---|
| West Digital | 10TB | WD100EFAX | PMR | 200 |
| | 8TB | WD8004FRYZ | PMR | 180 |
| | 4TB | WD40EZRZ | PMR | 180 |
| Seagate | 8TB | ST8000DM0004 | PMR | 180 |
| | 4TB | ST4000DM004 | SMR | 180 |

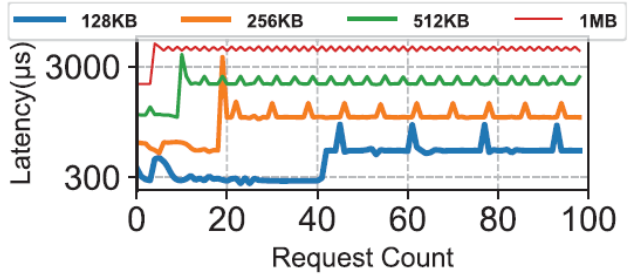
# Results Overview



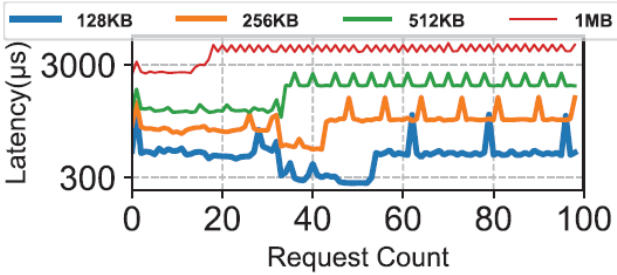

10TB West Digital HDD



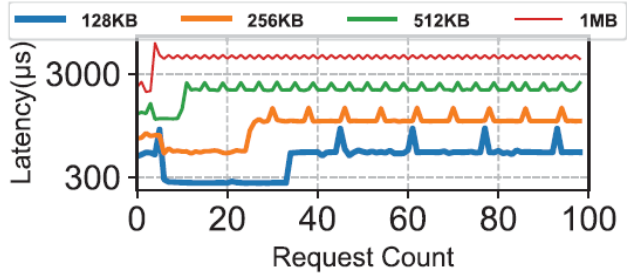

(a) 4TB West Digital HDD




(b) 4TB Seagate HDD




(c) 8TB West Digital HDD

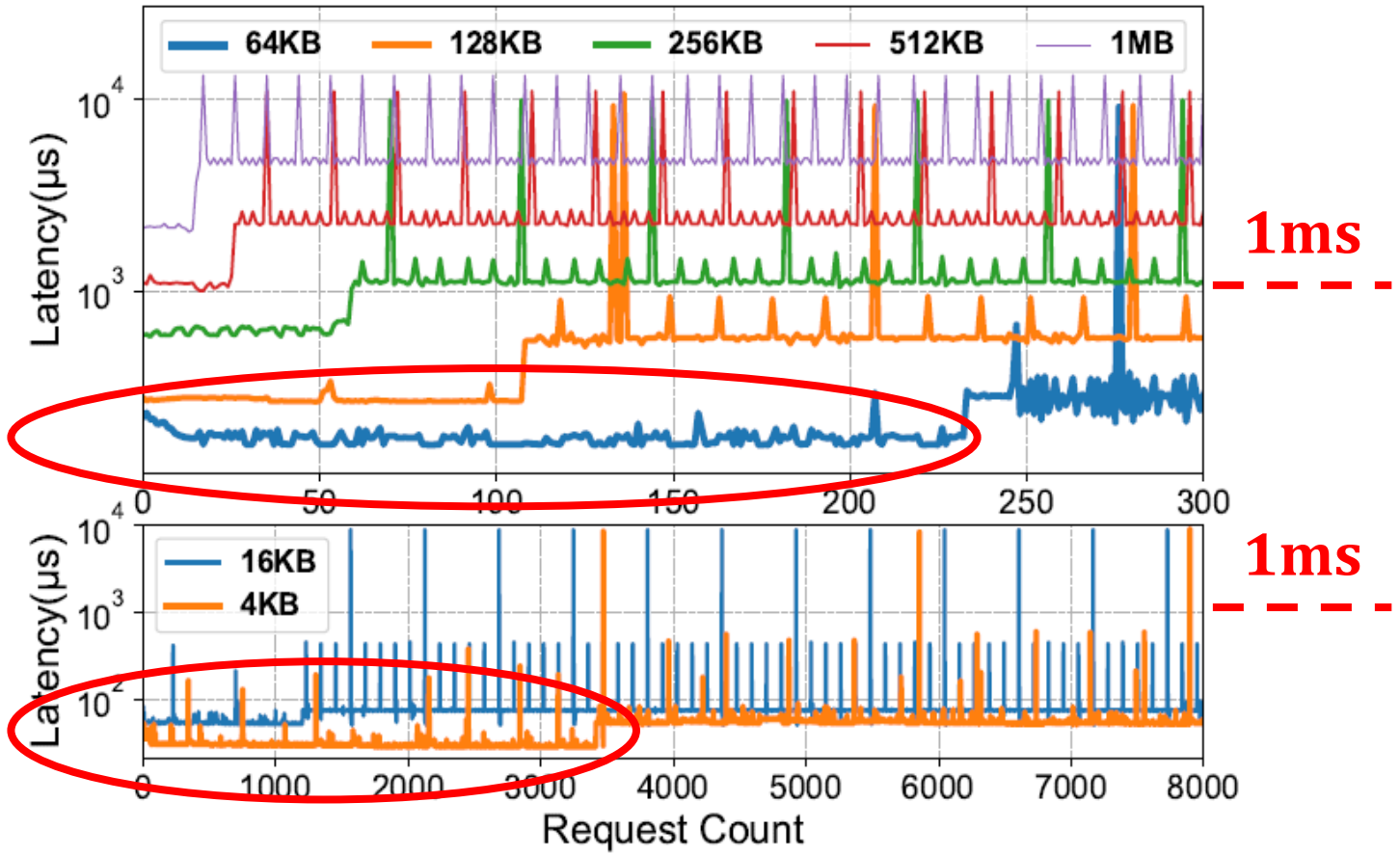



(d) 8TB Seagate HDD

# HDD write behaviors

HDD **can** reach ***μs*-level write latency**, especially for small size requests

***35us*** *for* ***4KB*** *writes*
***66us*** *for* ***16KB*** *writes*
***180us*** *for* ***64KB*** *writes*

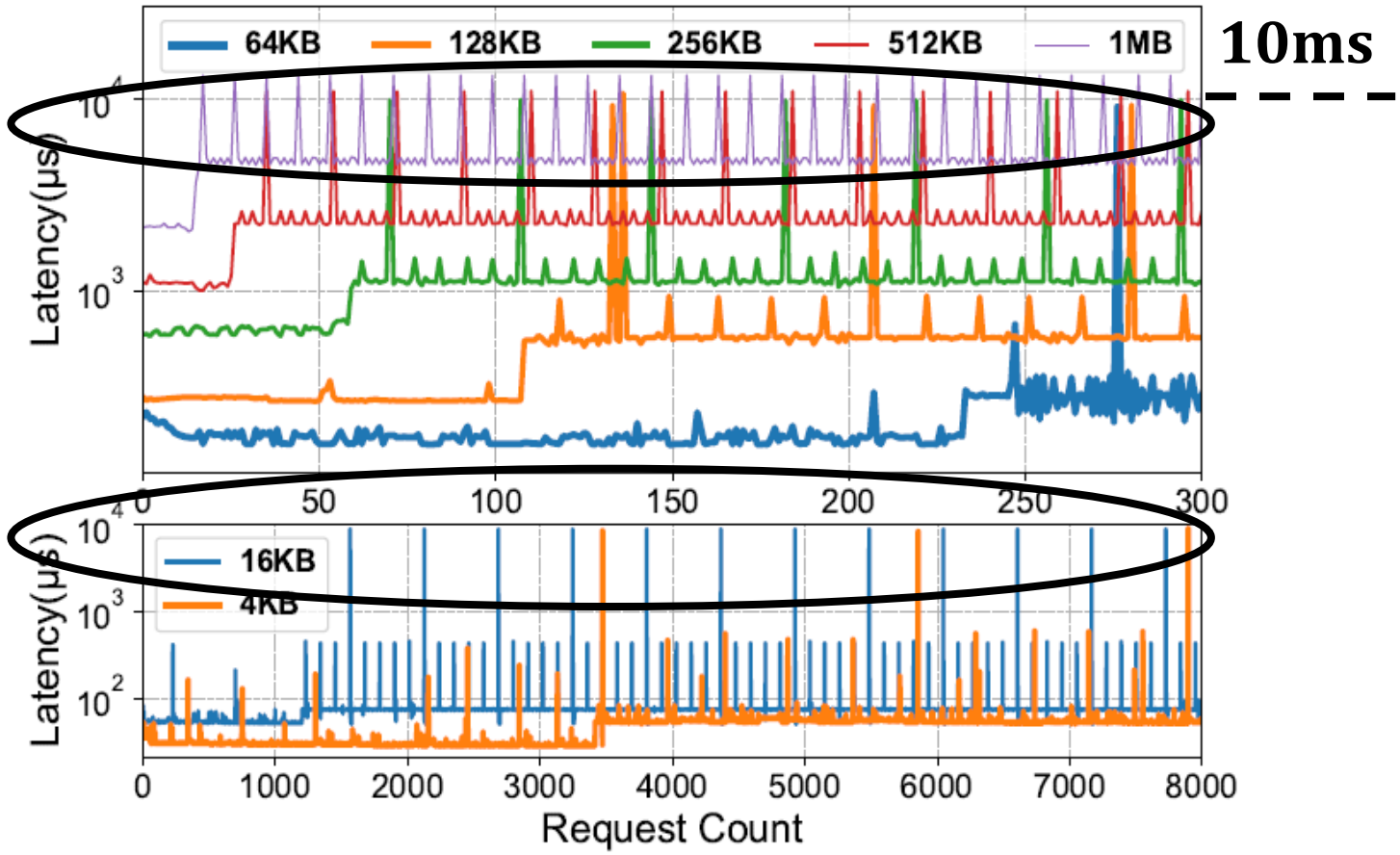


# HDD write behaviors

Reach **us-level** write latency, especially for small size requests

***ms*-level** latency spikes

*Higher than **10ms** for some spikes*

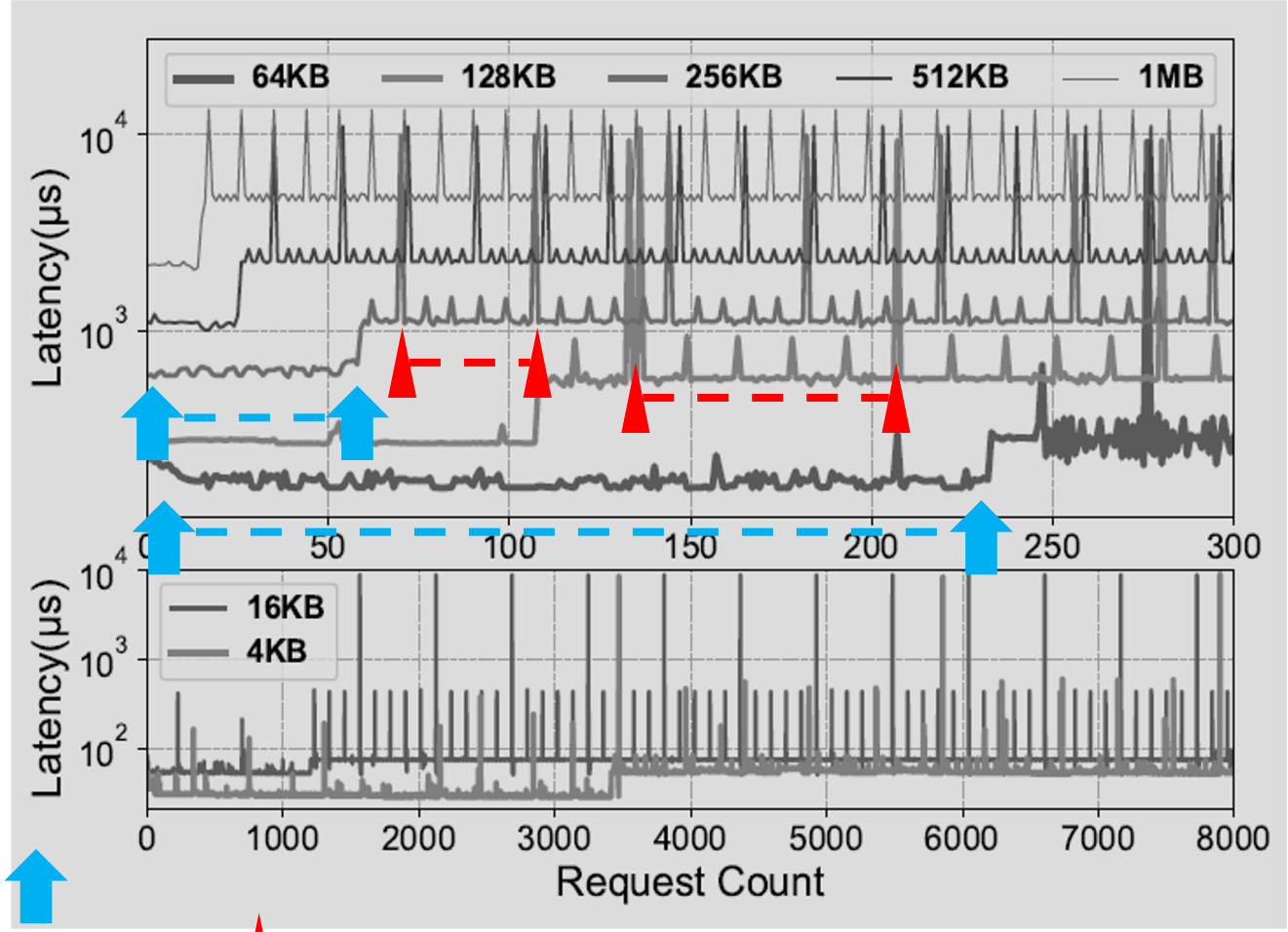


# HDD write behaviors

Reach to **us-level** write latency, especially for small size requests

ms-level latency spikes

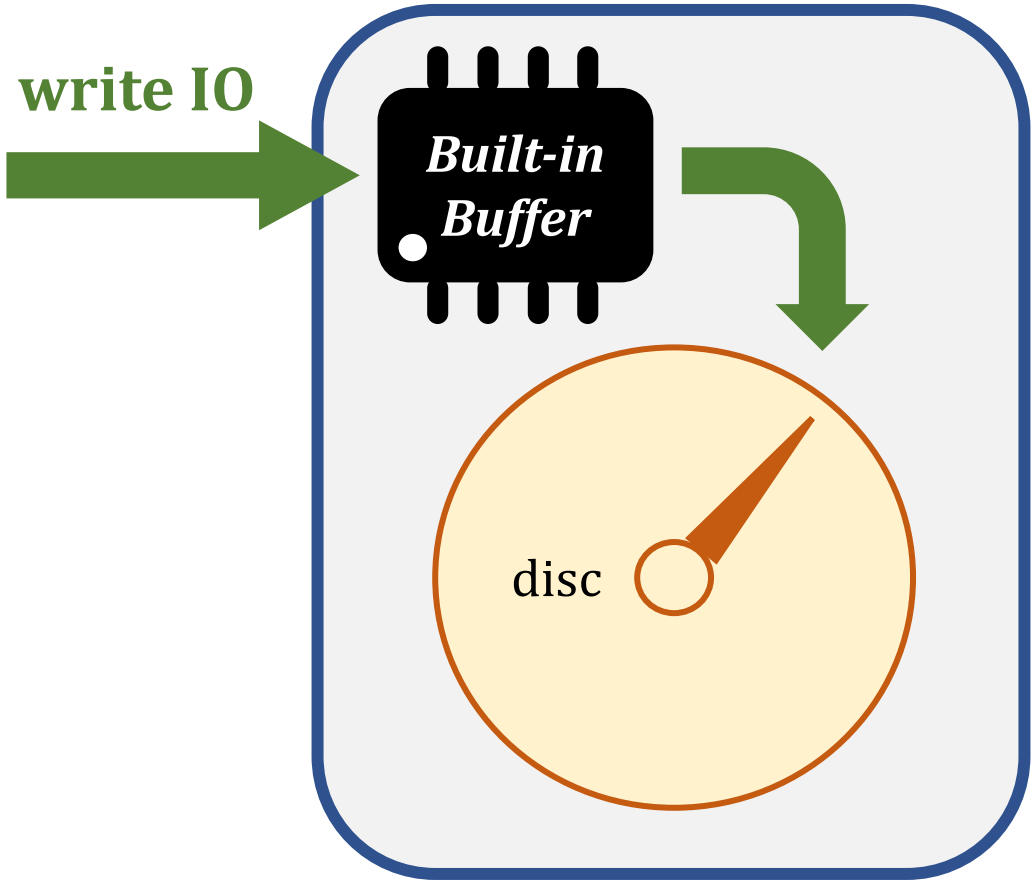
Fixed write latency period




*The length of the fastest write stage is **16MB***

*The interval between two highest latency spikes is **8MB***

# How does this happen?




**HDD Device**

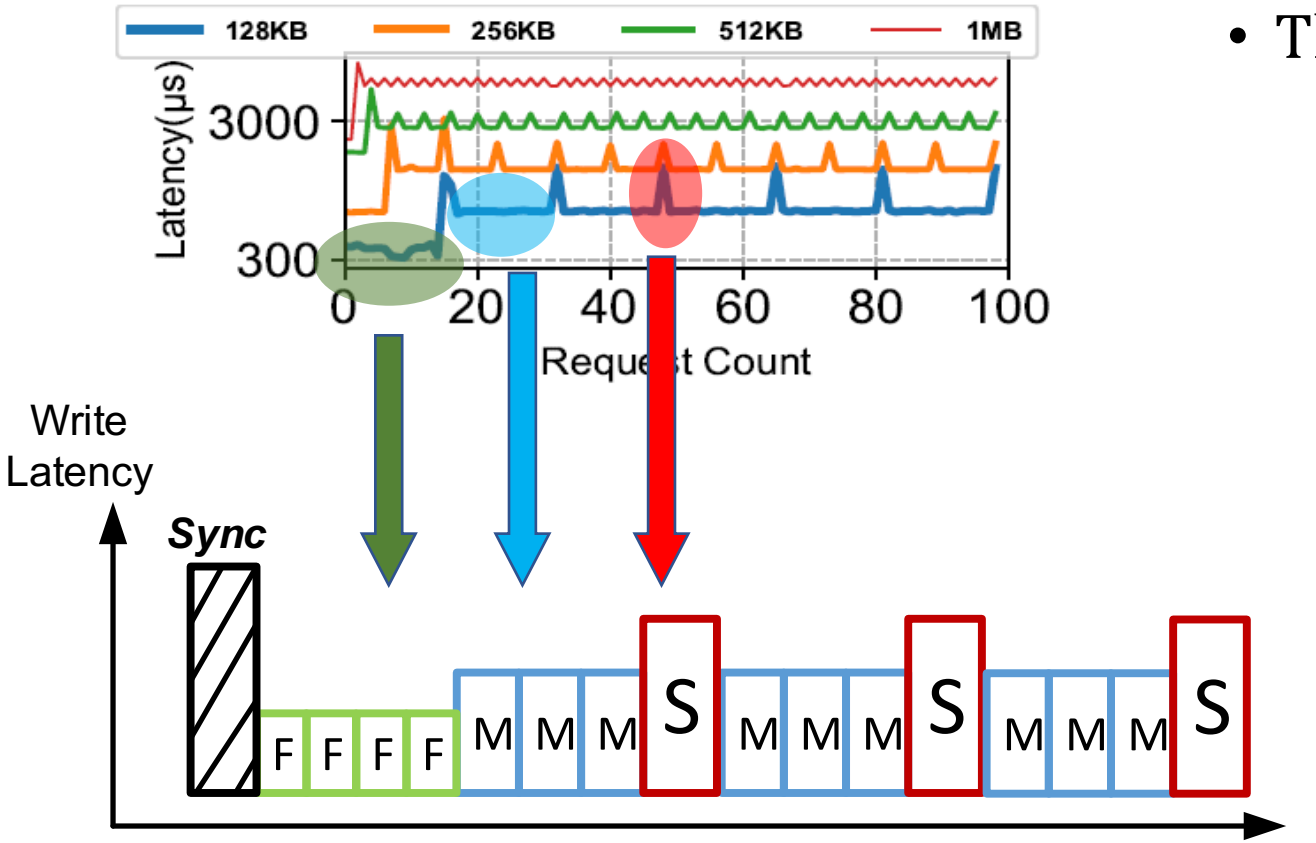
# Challenges




✓ **HDDs can reach $\mu s$-level write latency**

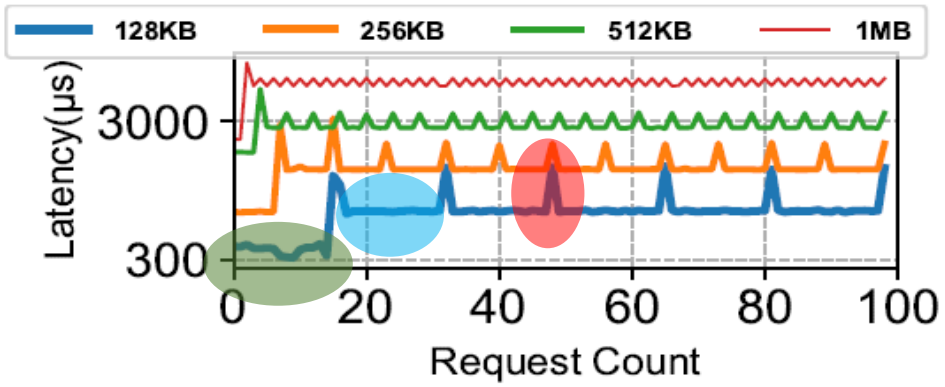
? **How to control these $\mu s$-level latency writes in HDDs?**

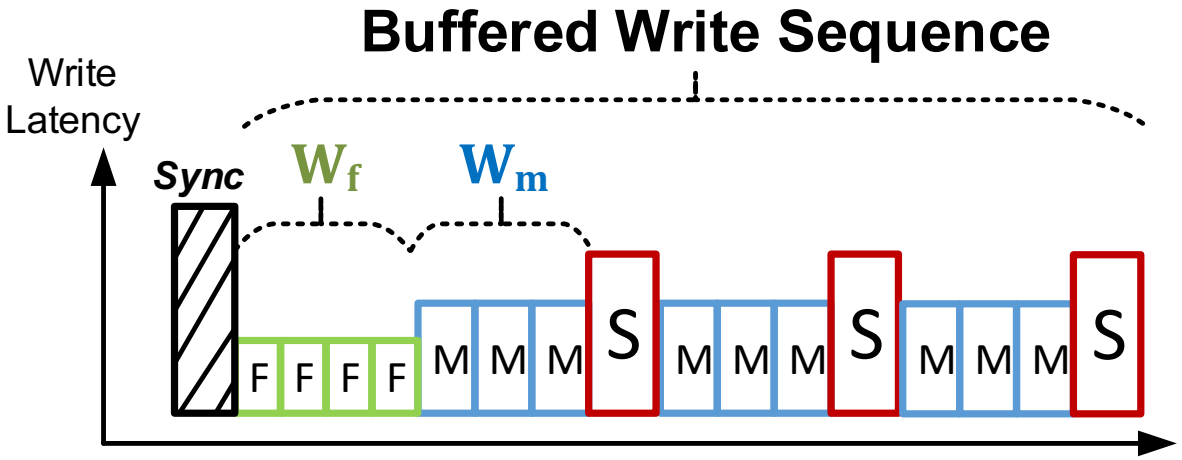
# HDD Buffered-Write Model




- Three types of **HDD buffered writes**
  - ***F***ast write (low-latency)
  - ***M***id write (mid-latency)
  - ***S***low write (high-latency spike)

# HDD Buffered-Write Model



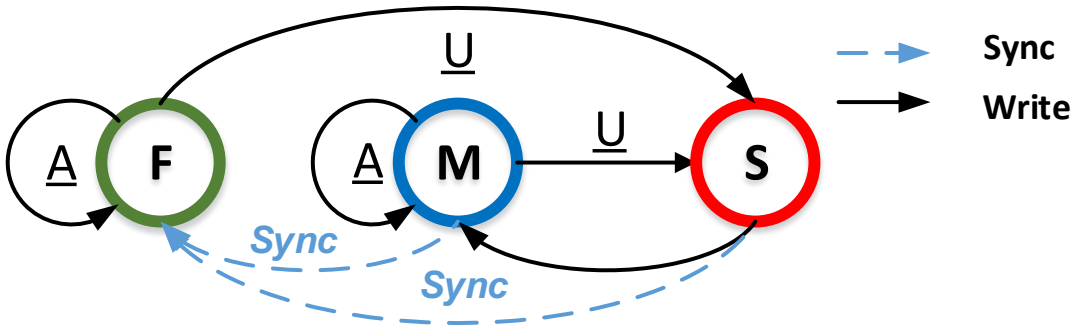

- Three types of HDD buffered writes
  - ***F***ast write (low-latency)
  - ***M***id write (mid-latency)
  - ***S***low write (high-latency spike)

- Buffered Write Sequence (after a Sync)

  Starts with a ***Fast* stage**, followed by one or more ***Slow-and-Mid* stage-pairs**
  - *Fast* stage lasts for $W_f$ data written
  - *Mid* stage lasts for $W_m$ data written
  - *Slow* stage contains an ***S***low write

# Write-state Predictor

- The next HDD write state could be predicted, according to:
  - ① Write state of the current request
    - Each write request can only be one of the ***F***, ***M***, or ***S***
  - ② Current $ADW$, $W_f$ and $W_m$ values
    - **A**: $ADW < W_f$ and $W_m$
    - **U**: $ADW \geq W_f$ and $W_m$
  - ③ $Sync$ operation
    - Takes the next write state back to $F$



# Buffer-Controlled Writes (BCW)

- Goal:
  - Ensures user writes to be in the ***F*** or ***M*** write state, avoids writes in the ***S***low state.
- Steps:
  - Perform **profiling** for all key parameters (if unknown)
  - Invokes ***Sync*** operation when starting BCW
    - To flush all data in the HDD build-in buffer



# Buffer-Controlled Writes (BCW)

- Write user data to HDD when there are user requests in the queue

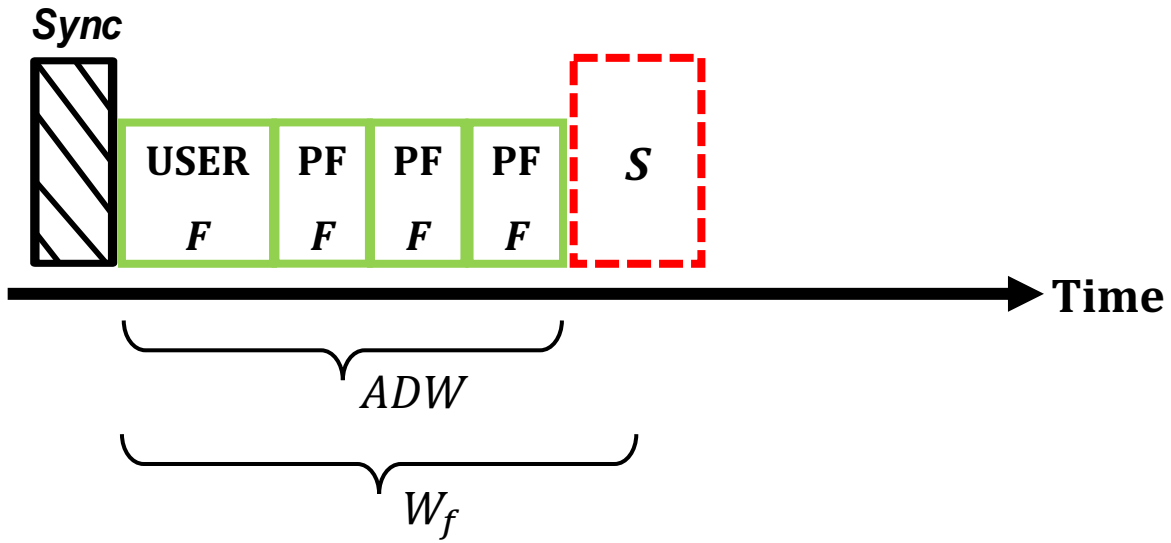



- **Actively pads non-user data** to HDD when there are no user requests
  - **PF**: for padding the $F$ and $M$ stage (i.e., 4KB)
  - **PS**: for padding the $S$ stage (i.e., 64KB)



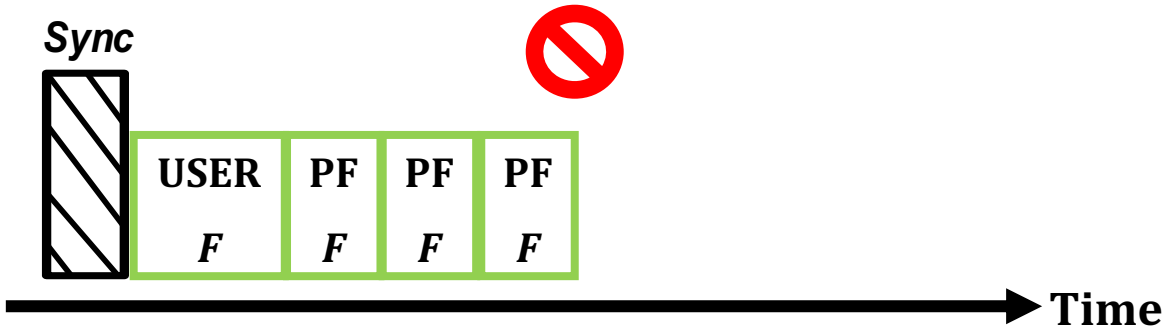
# Buffer-Controlled Writes (BCW)

- When current data written ($ADW$) in the $F$ and $M$ states are close to the $Wf$ and $Wm$ values
  - ① The next HDD write state will be **predicted as *S***



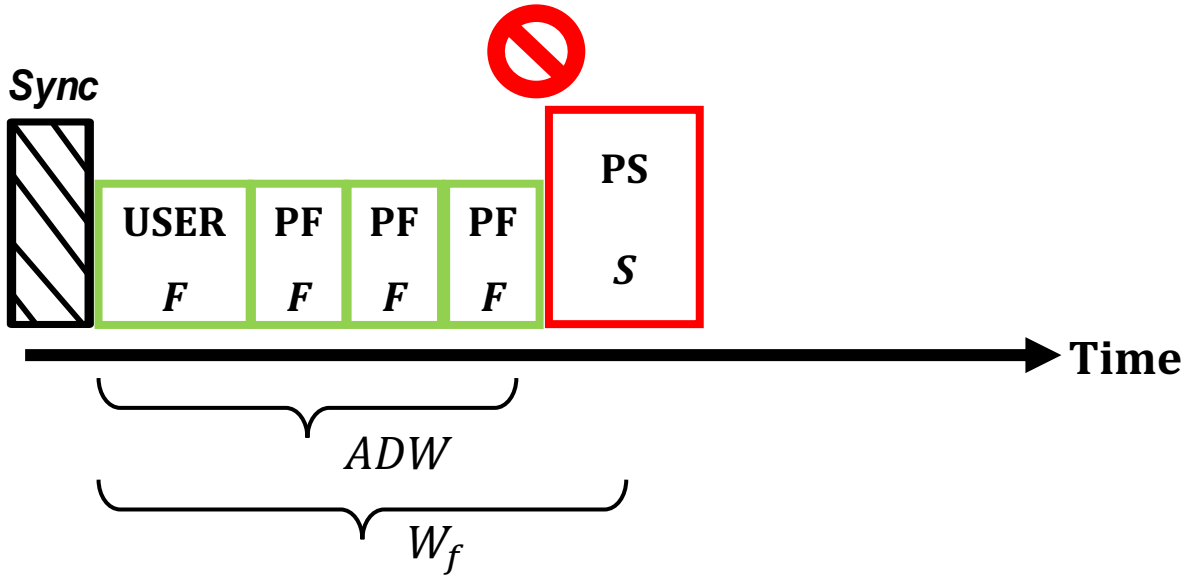
# Buffer-Controlled Writes (BCW)

- When current data written ($ADW$) in the $F$ and $M$ states are close to the $Wf$ and $Wm$ values
  - ① The next HDD write state will be predicted as $S$
  - ② **Stop** receiving user requests



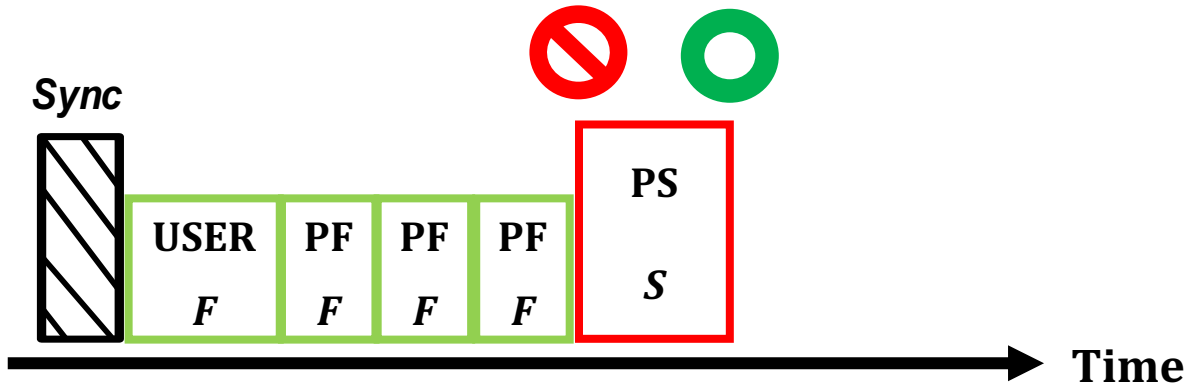
# Buffer-Controlled Writes (BCW)

- When current data written ($ADW$) in the $F$ and $M$ states are close to the $Wf$ and $Wm$ values
  - ① The next HDD write state will be predicted as $S$
  - ② Stop receiving user requests
  - ③ Continuously pads **PS**, until a ***S*** write state is detected



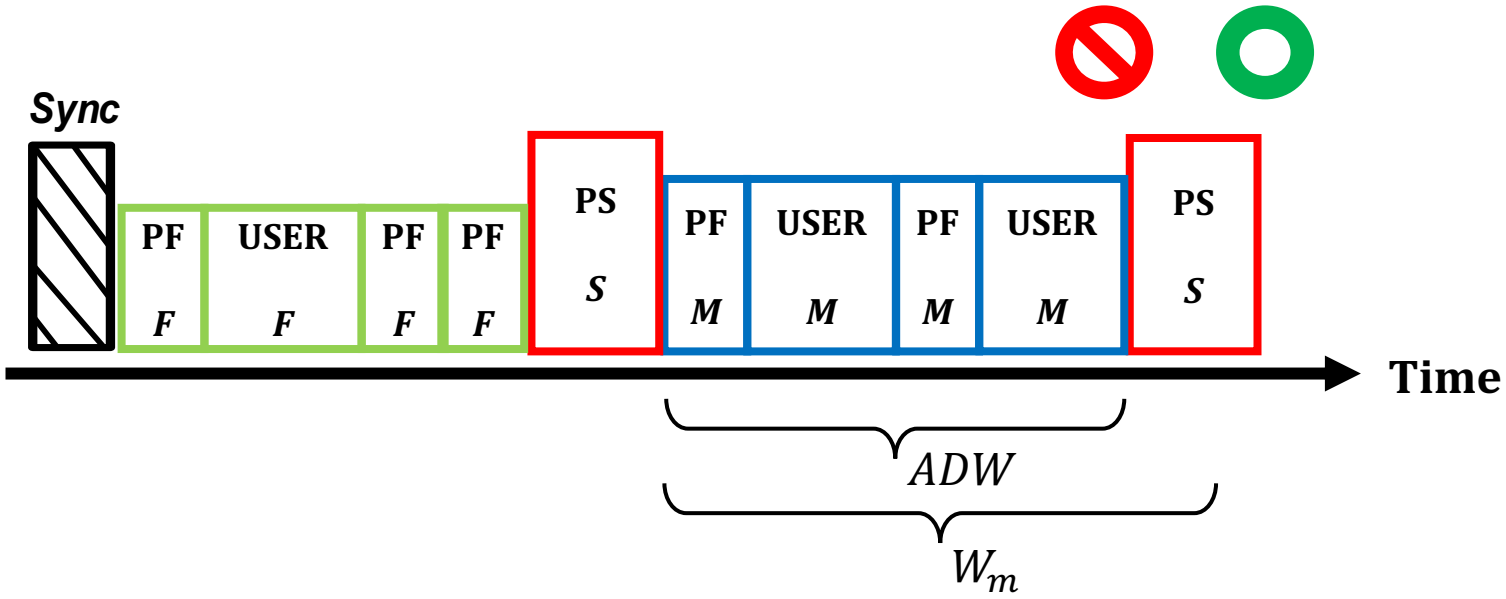
# Buffer-Controlled Writes (BCW)

- When current data written ($ADW$) in the $F$ and $M$ states are close to the $Wf$ and $Wm$ values
  - ① The next HDD write state will be predicted as $S$
  - ② Stop receiving user requests
  - ③ Continuously pads **PS**, until a $S$ write state is detected
  - ④ **Start** receiving user requests



# Buffer-Controlled Writes (BCW)

- Same steps of padding **PF** and **PS** in $M$ stages



# Challenges



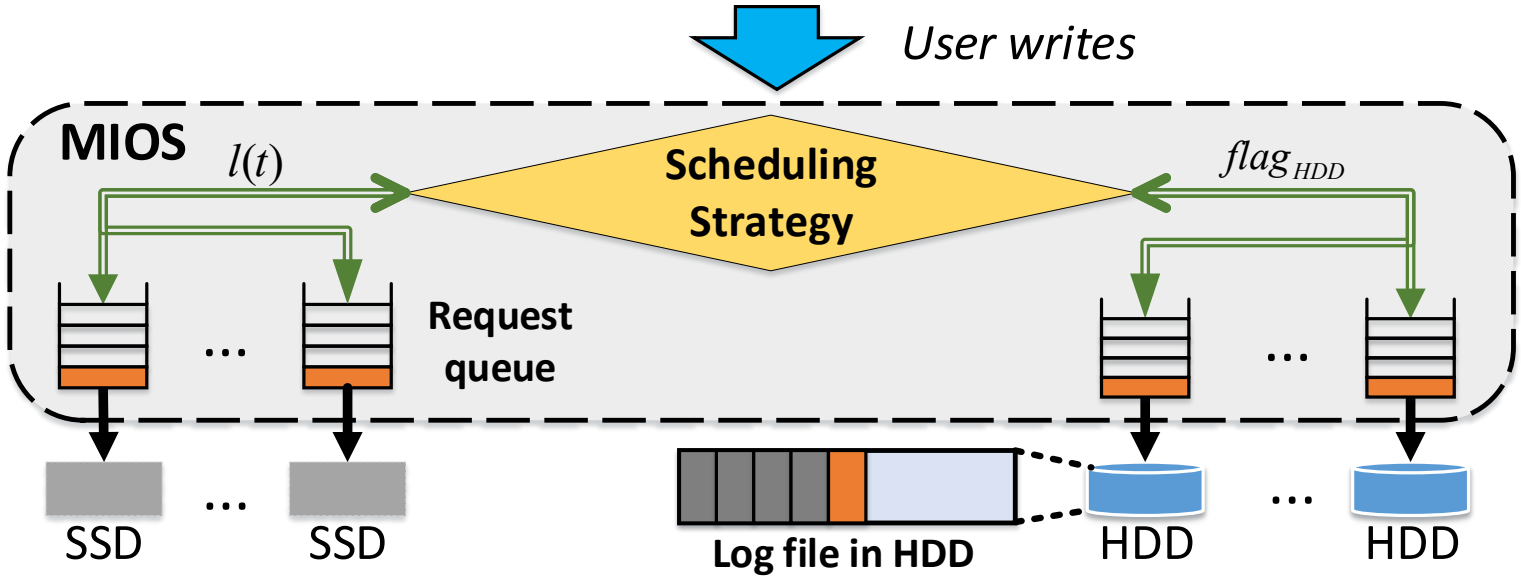

✓ **HDDs can reach $\mu s$-level write latency**

✓ **BCW provides a proactive and controllable buffer writing approach**

? **How to leverage BCW effectively in the hybrid storage nodes?**

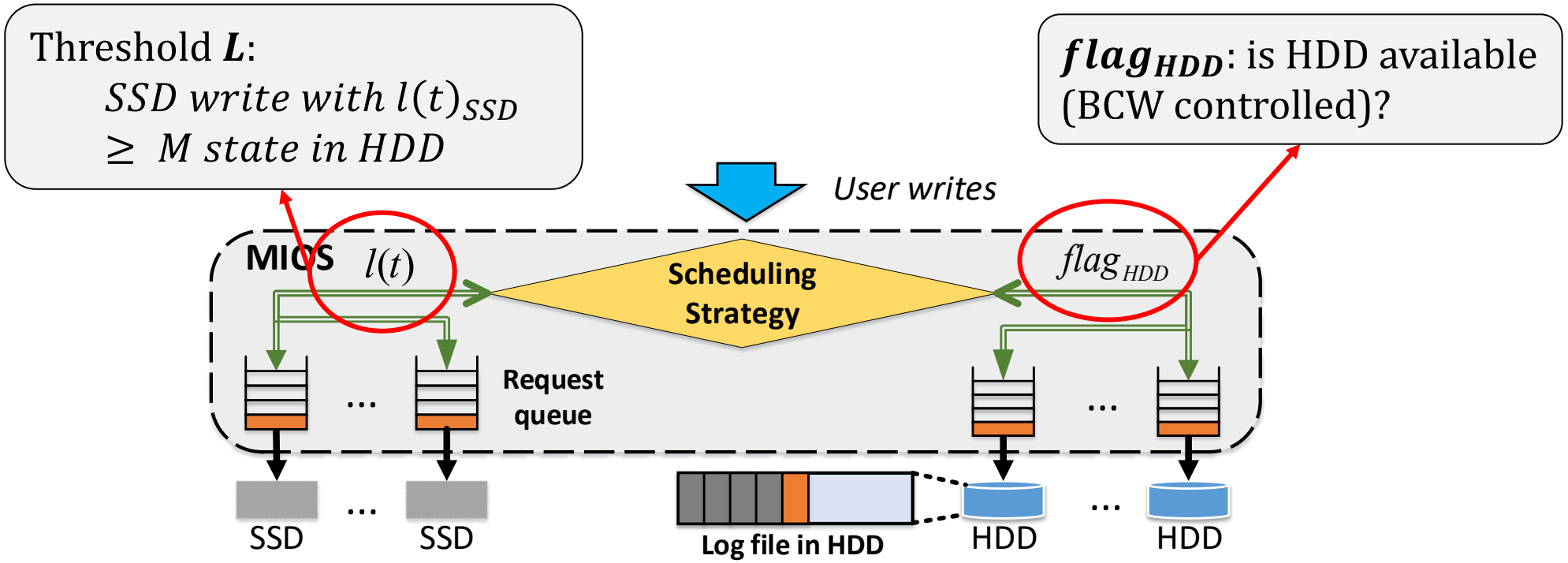
# Mixed IO scheduler (MIOS)

- A scheduler that schedules mixed IOs at runtime
- Architecture
  - A request queue for each SSD and HDD, and monitors their length $\boldsymbol{l(t)_{SSD}}$ *and* $\boldsymbol{l(t)_{HDD}}$
  - Log file in each HDD, a device file storing BCW writes in an append-only manner.



# Mixed IO scheduler (MIOS)

- Key parameters:

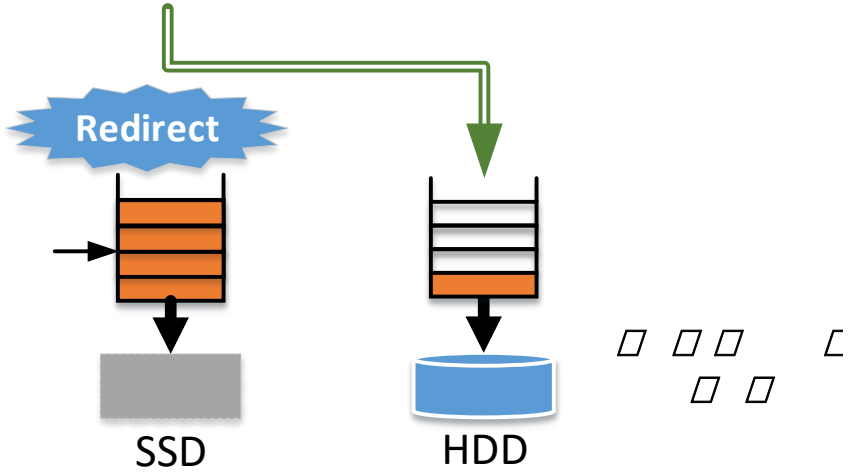


# Scheduling strategies in MIOS

**MIOS_D:**

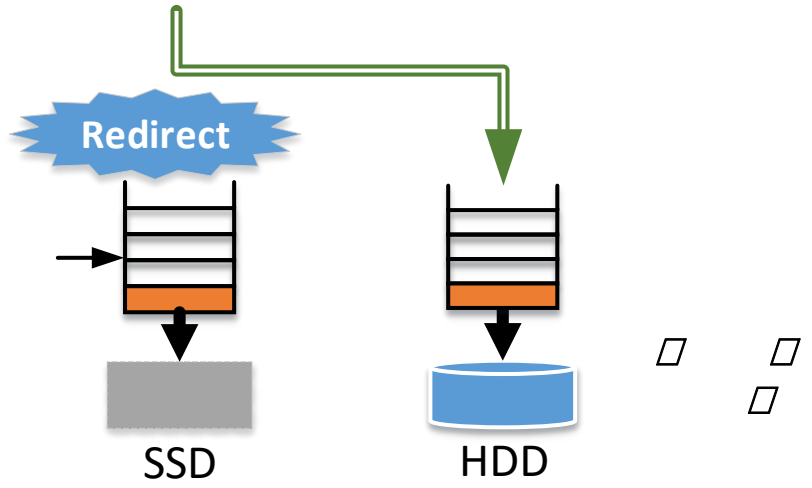
redirect SSD writes to HDDs when:

- $l_{SSD}$ is **higher** than the threshold $L$ **AND**
- HDD is the $F$ or $M$ write state with BCW




**MIOS_E:**

- **Same** as MIOS_D when $l_{SSD} > L$
- Further perform redirection with just $F$ write state when $l_{SSD} < L$



# Evaluation Setup

- Comparisons
  - **Baseline**: Pangu workload replay (writing all data into SSDs)
  - **MIOS_D**
  - **MIOS_E**
- Evaluation environment

| System | Linux version 4.15.0-52-generic |
|---|---|
| CPU | Intel Xeon E5-2696 v4 (2.20 GHz, 22 CPUs) |
| Memory | 128 GB |
| HDDs | West Digital 10TB (default) |
| | West Digital 4TB |
| | Seagate 4TB |
| SSDs | Samsung 960EVO 256GB（NVMe, 2000MB/s） |

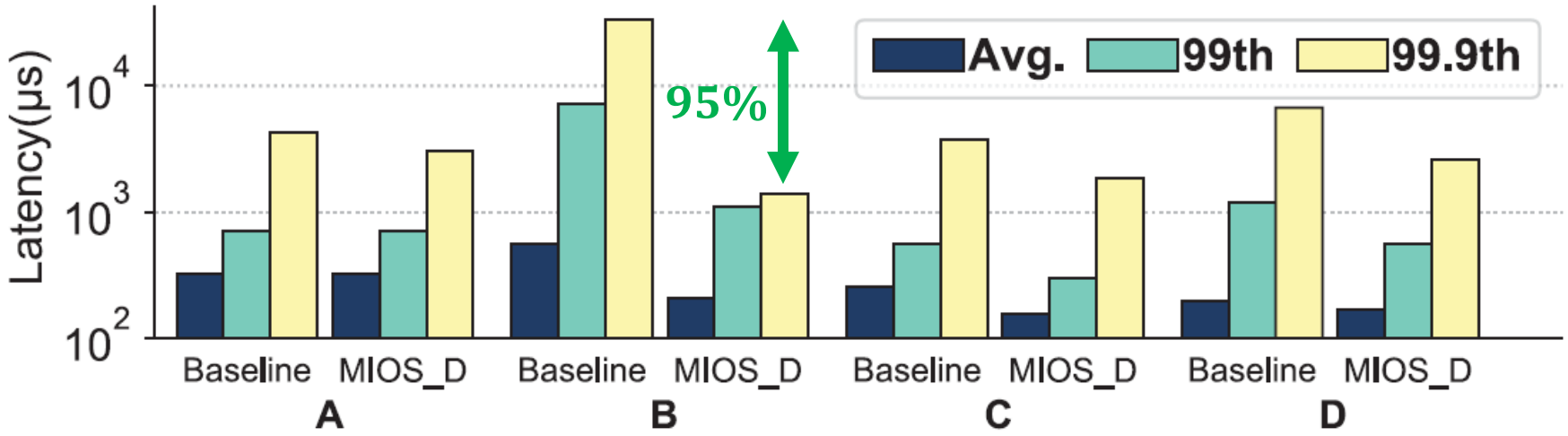
# Evaluation Setup

- Workload characteristics

| Workload Types | A | B | C | D |
|---|---|---|---|---|
| Business | Cloud Computing | Cloud Storage | Structured Storage | Structured Storage |
| SSD Writes (GB) | 14.7 | **61.2** | 7.2 | 7.5 |
| SSD Write Requests (millions) | **0.43** | 4.4 | 4.8 | 4.7 |
| Note | **Lowest IO intensity** | **Most written data** | | |

# Write Performance

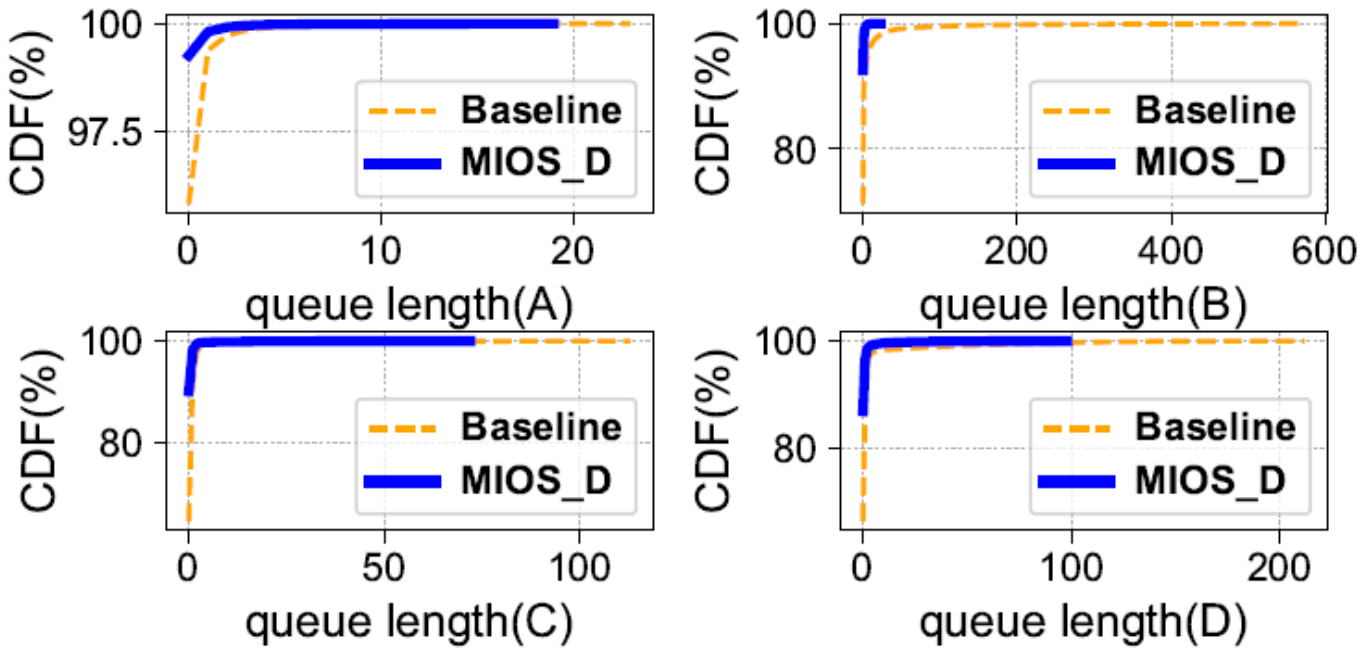
- Average, $99^{th}$ and $99.9^{th}$-percentile tail latency
  - ➢ Reduced by 65%, 85%, and 95% respectively in workload B
  - ➢ Reduced by 2%, 3.5% and 30% respectively in workload A



# Queue Length Reduction

**Workload A**

Minimum (15%) reduction in queue lengths

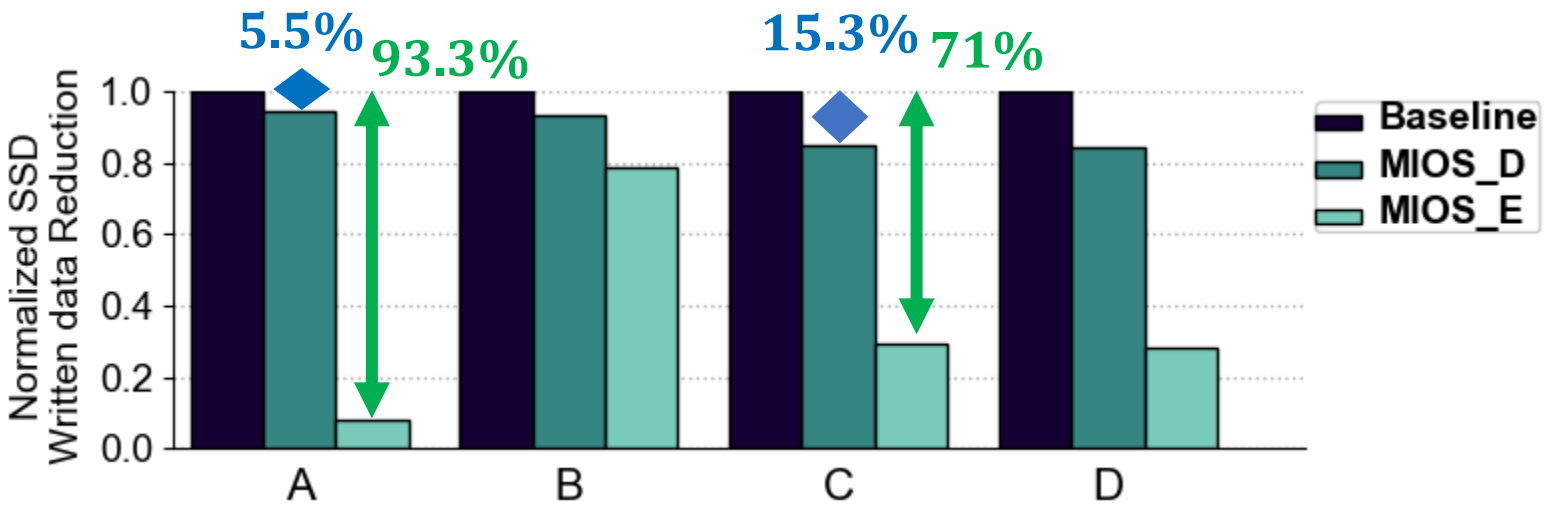



**Workload B**

Maximum (95%) reduction in queue lengths

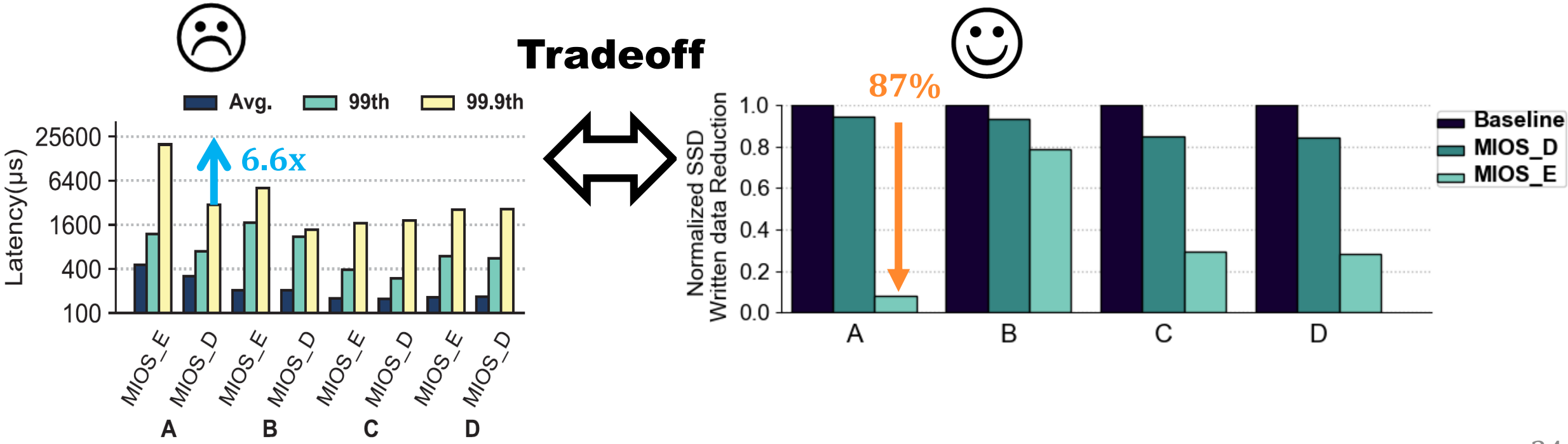
# SSD Written Data Reduction

- **MIOS_D**:
  - Reduced **5.5%** compared with *Baseline* in workload A
  - Reduced **15.3%** and **16%** in workload C and D
- **MIOS_E**:
  - Reduced **93.3%** compared with *Baseline* in workload A
  - **71%** and **72%** less than *Baseline* in workload C and D



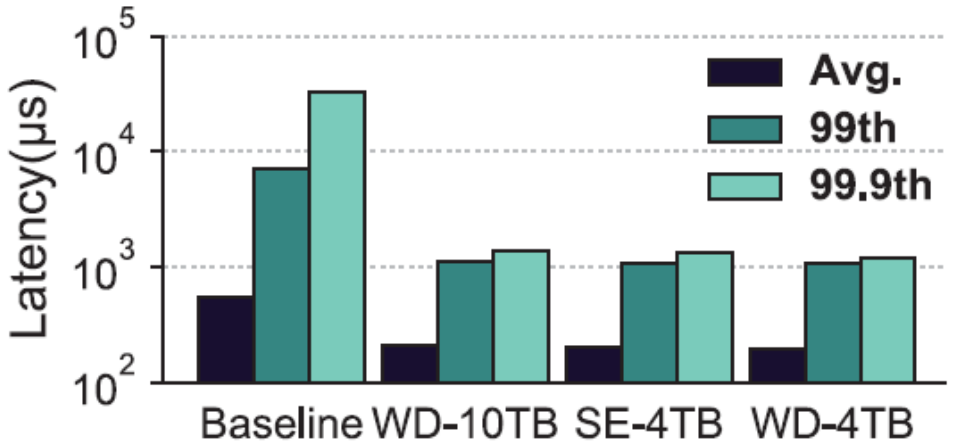
# Write Performance: MIOS_D vs MIOS_E

- MIOS_E leads to **worse** latency performance
  - MIOS_E leads to **1.4x** higher average latency, **1.7x** higher $99^{th}$-percentile latency, **6.6x** higher $99.9^{th}$-percentile latency than MIOS_D in workload A



# Experiment with other HDDs

- Different types of HDDs do not have a significant impact
  - The maximum difference of average and tail latency is less than **3%**
  - Amount of data written to and number of requests processed in SSD with different HDDs is less than **5%**




| | Baseline | WD 10TB | WD 4TB | SE 4TB |
|---|---|---|---|---|
| SSD written data (GB) | **61.2** | **4.1** | **4.2** | **4.4** |
| SSD write requests (thousands) | 4453 | 720 | 724 | 769 |

# *Thanks! Q&A*

***Email:*** ***wsczq@hust.edu.cn***