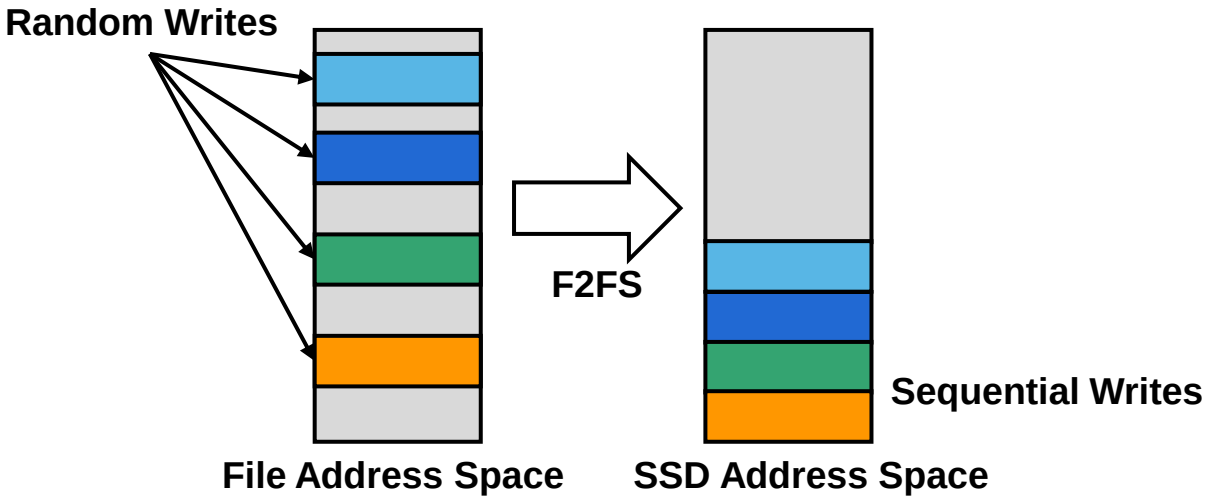
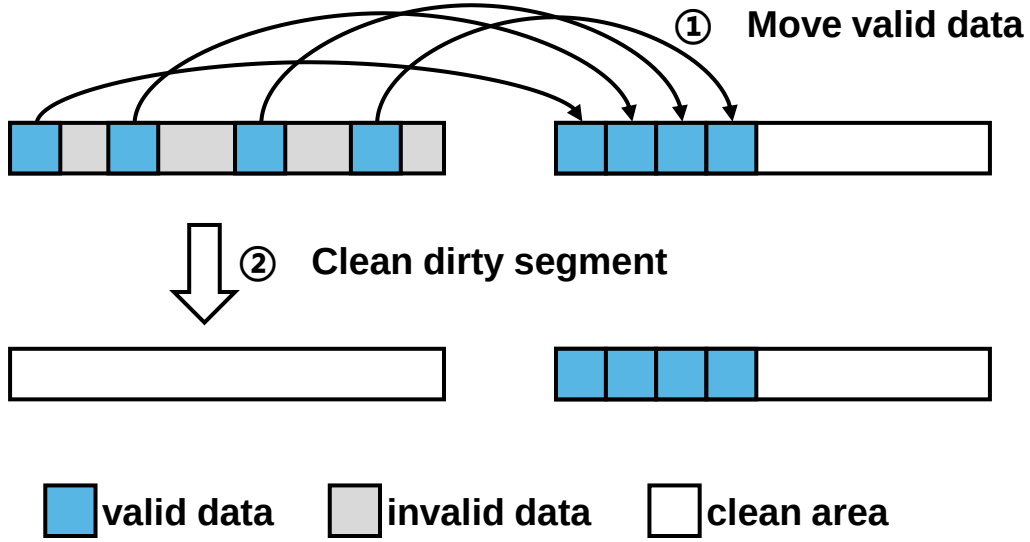
When F2FS Meets Compression-Based SSD!

Yunpeng Song, Yiyang Huang, Yina Lv, Yi Zhang, Liang Shi
East China Normal University

F2FS and Compression

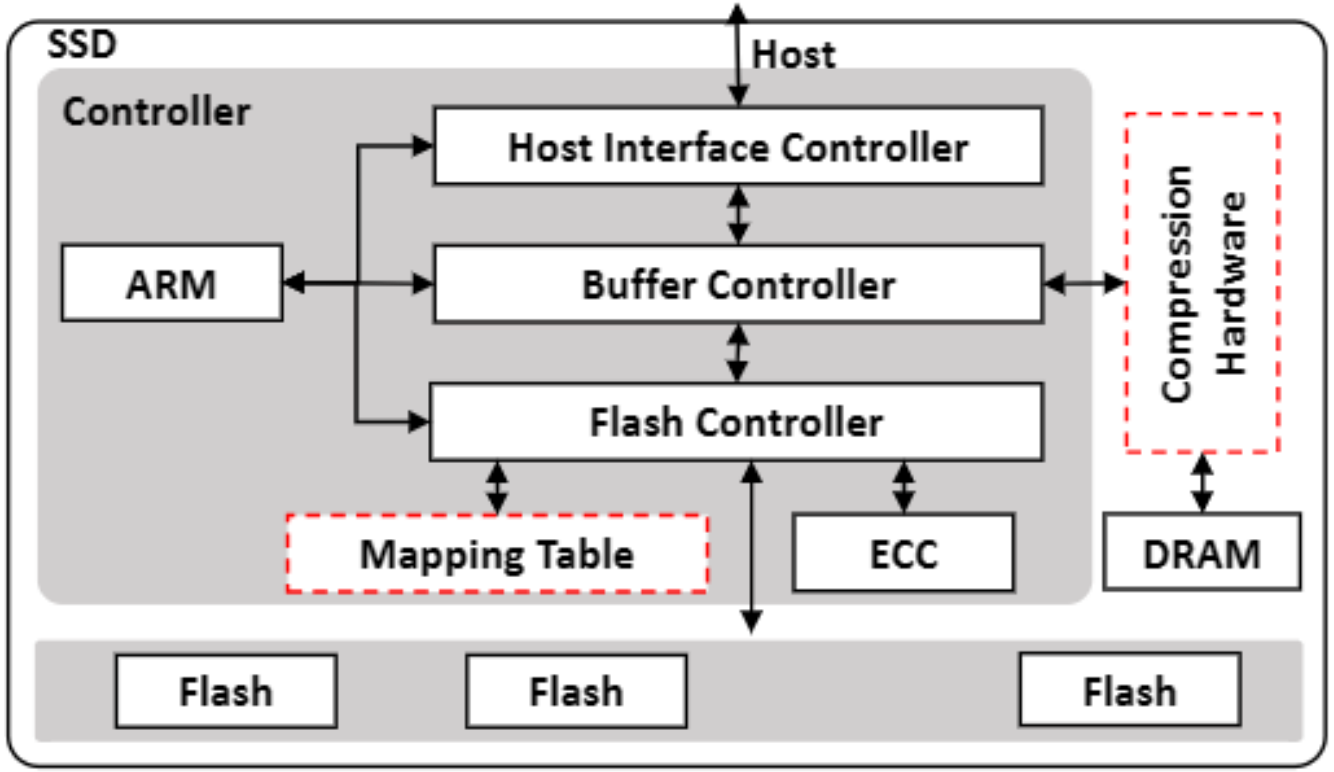
- F2FS converts **random** writes to **sequential** writes
- F2FS needs **over-provision space** (**OPS**) to perform **Garbage Collection** (**GC**)
- F2FS uses **compression** to improve the **lifetime** of flash memory
- The space saved by F2FS compression is reserved for **OPS** thus reduces GC cost





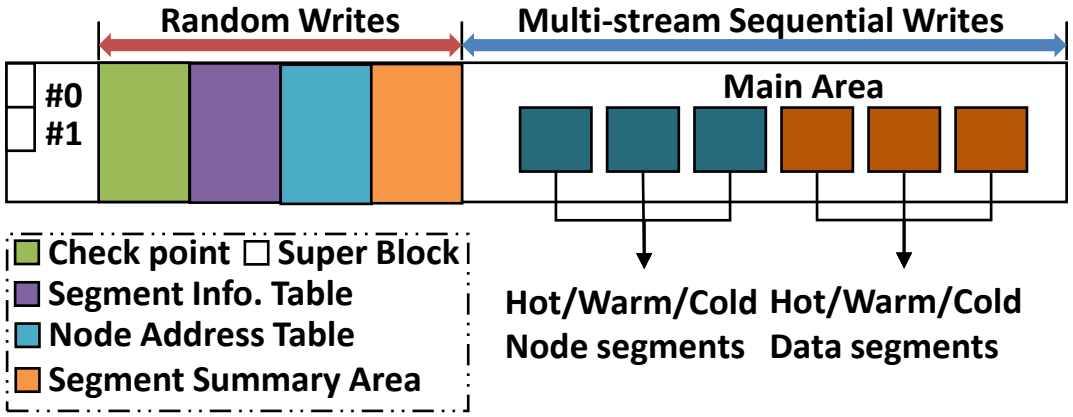
Intra-Device Compression

- **Hardware-implemented** compression module on the **read/write path**
- Uses extended **mapping table** to produce variable-length compressed data
- The **logical** and **physical** space of the CSD is **decoupled**

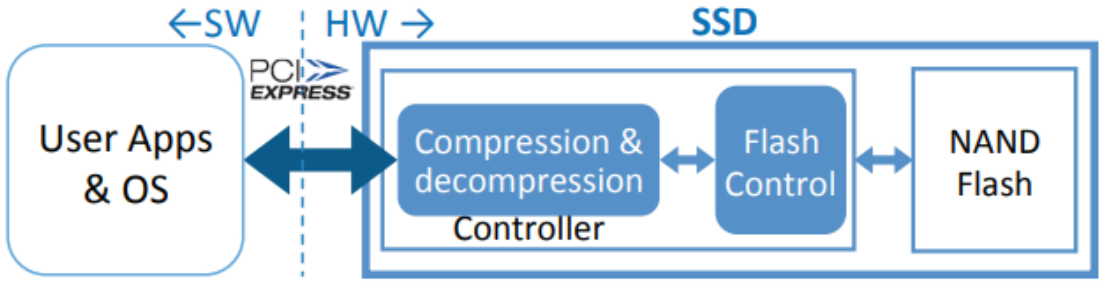


State-of-the-art works

- **Mitigate GC affects**
  - improving GC efficiency
  - optimise GC timing



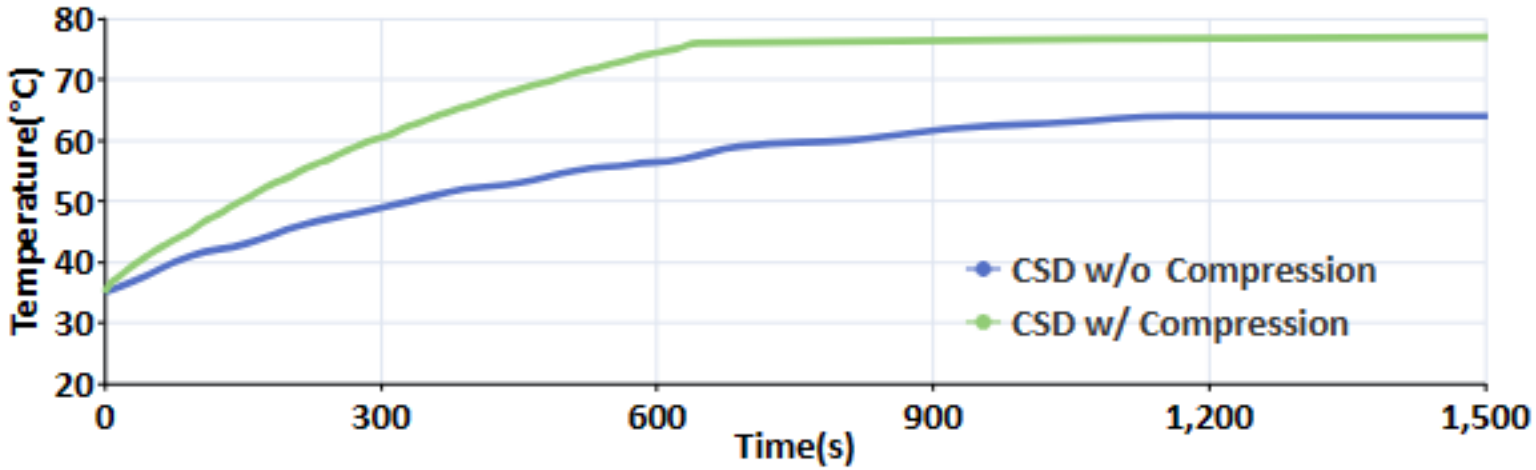
- **Intra-device compression**
  - mitigate host cost
  - optimize compression throughput



None of them focus on F2FS and CSD co-design for a win-win solution.

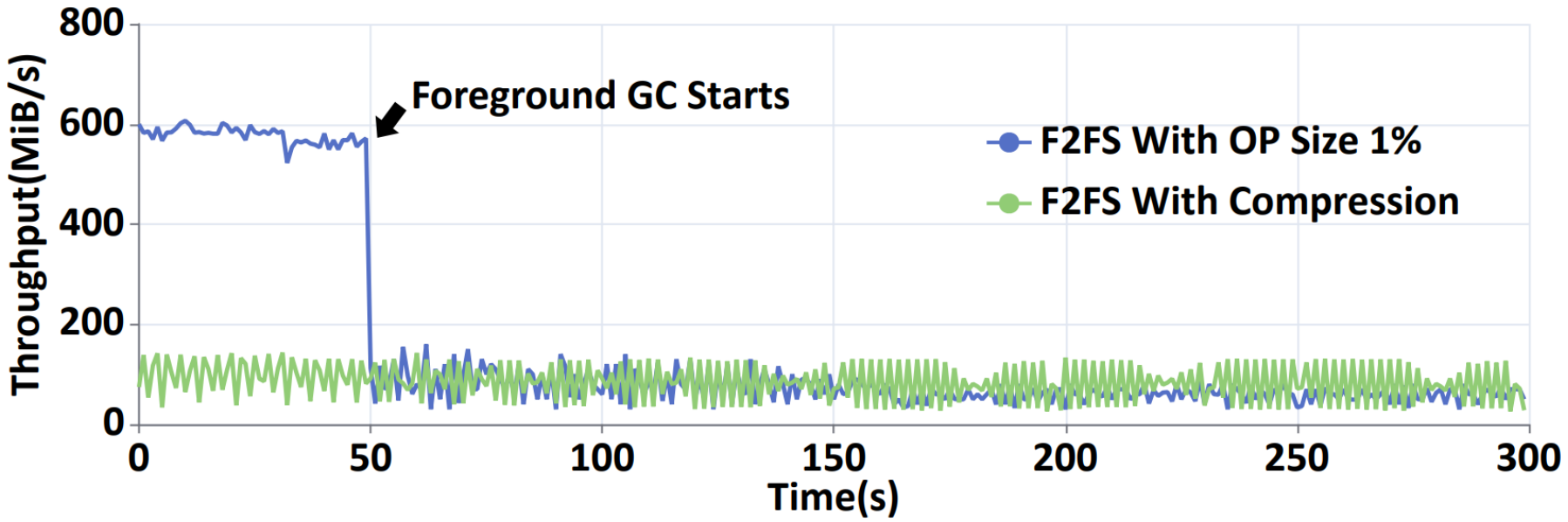
Power Consumption for CSD

- 128KB sequential write workload
- Temperature of CSD increases with time so does power consumption
- Temperature of CSD with compression is higher than that of CSD without compression
- Temperature of CSD with compression stables because of heat protection



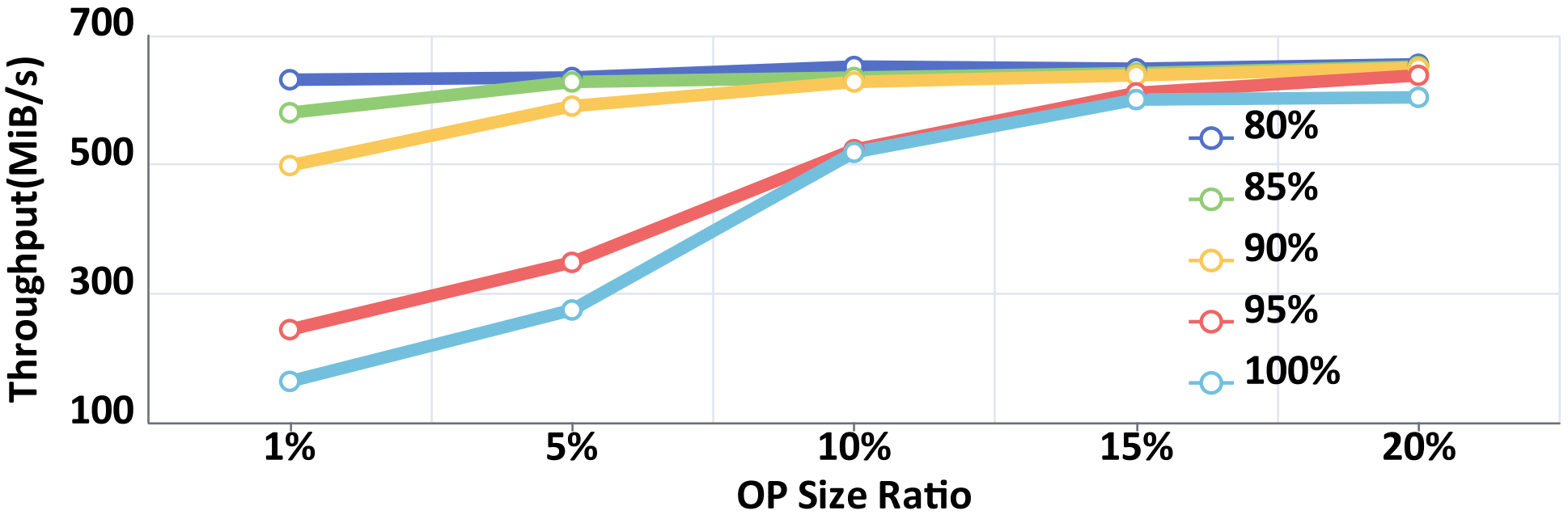
GC for F2FS

- GC causes severe **performance degradation**
- performance of F2FS with compression is **similar** to the performance with GC



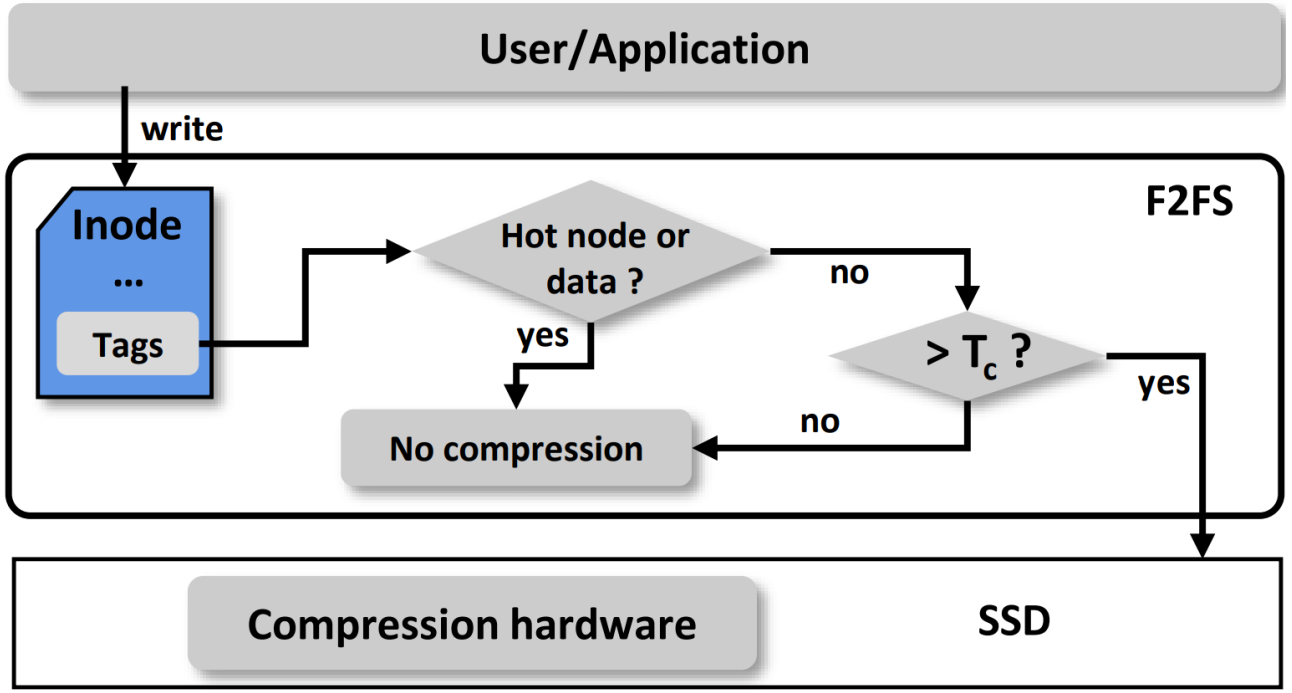
Performance with OP size ratio

- The performance **peaked** when the OP size is 15%
- **OP Space Ratio > 15% brings negligible benefit**



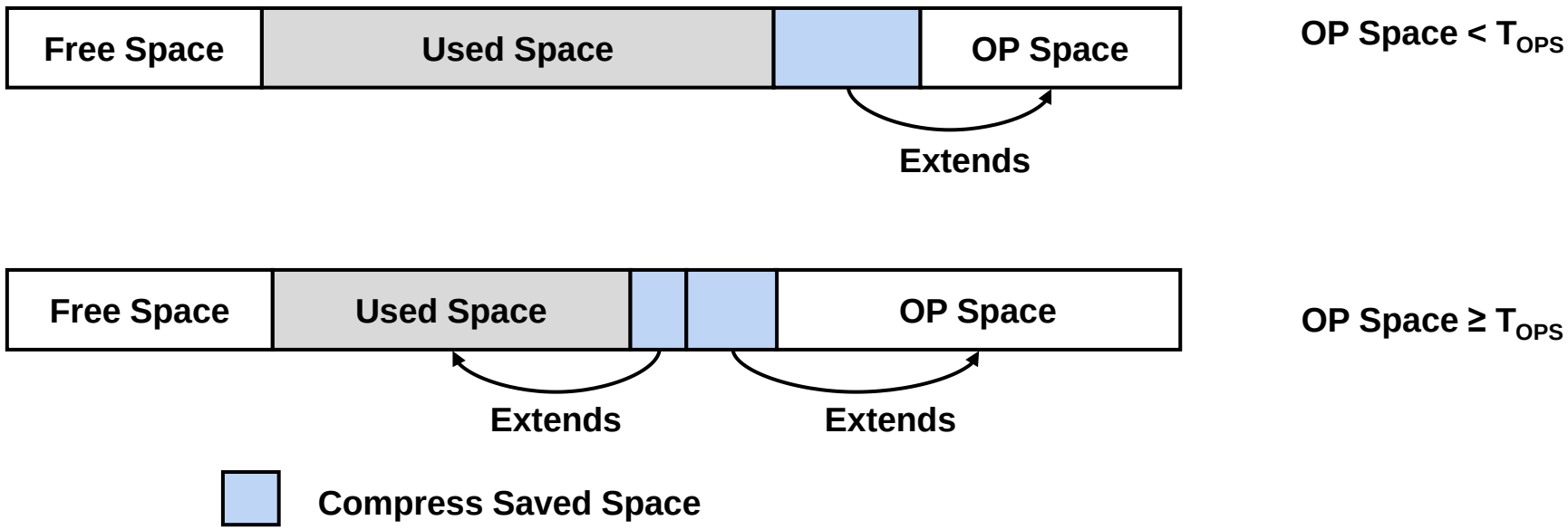
COCO: Feature-Aware Compression Scheduling

- **Tag** is determined based on the hot and cold tags of F2FS
- **compression ratio(CR)** is determined by the first three data blocks of the file
- when CR > Tc and data is cold, the compression tasks are scheduled to the CSD



COCO: On-Demand Space Allocation

- Increasing OPS **does not** improve performance when it reaches a specific value
- Expose part of the space saved by compression to **user**
- **Space saved by compression** is obtained the actual usage of CSD
- **OP size growth threshold** is set through offline analysis



Implementation and Overhead Analysis

- Use the **top 1 bit of the block's logical address** as the **status bit**
- only **writes** will change the status bit
- Record some thresholds as global variables
- **The metadata overhead can be negligible**
- No modifications to CSD hardware

Experimental Setup

- Experiment platform
  - R7-5950x@5.9GHz with 6 cores; 32GB RAM
  - 3TB CSD (**hardware zlib compression**)
    - 200GB F2FS partition
  - Linux kernel 5.15.0

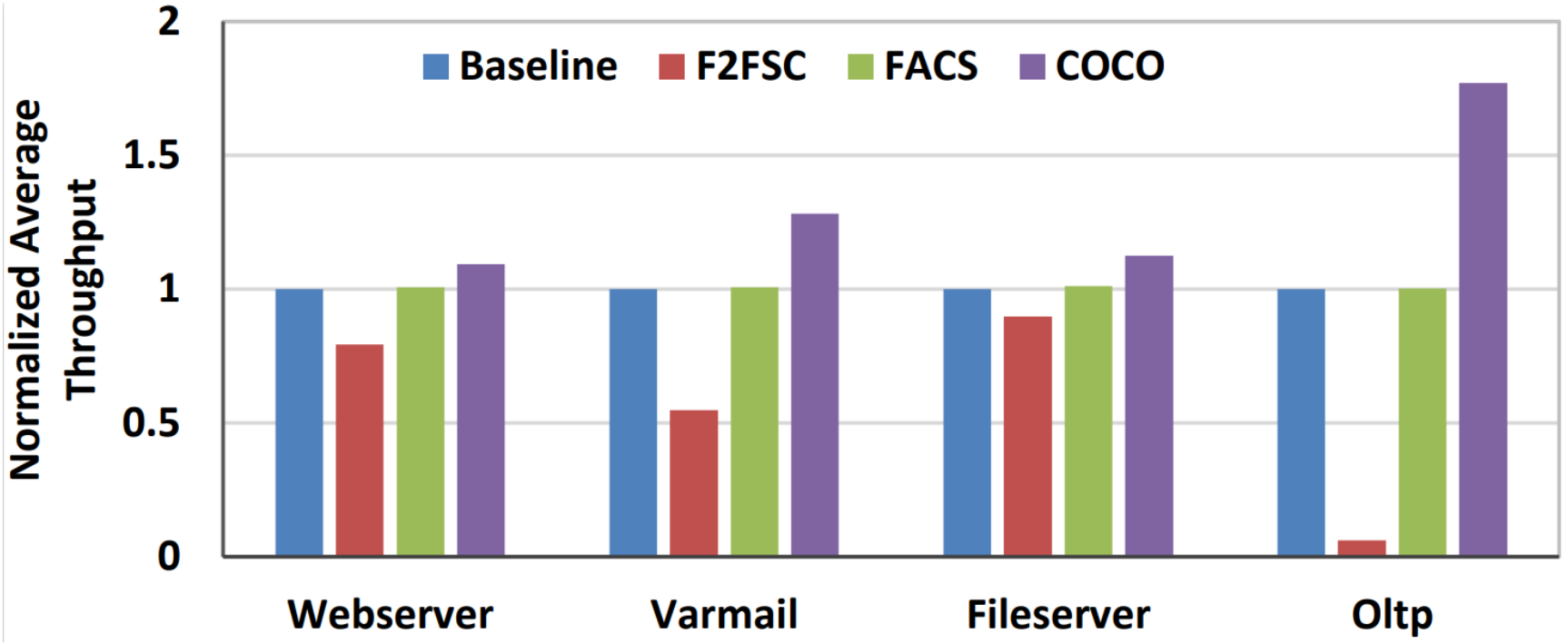
Workloads

- Final performance is the average of the result of 3 times test
- The compression ratio is determined by the workload characteristics
- Assume the **compression ratio distribution** of the real workloads confirms the **Zipfian distribution**

Workloads	Pattern	I/O Size	R/W Ratio	Compression Ratio
Webserver	Sequential	16KB-1MB	9 : 1	1 - 2
Varmail	Sequential	1MB	1 : 1	2 - 3
Fileserver	Random	16KB	1 : 2	2 - 3
Oltp	Random	2KB-256KB	1 : 9	3 - 4

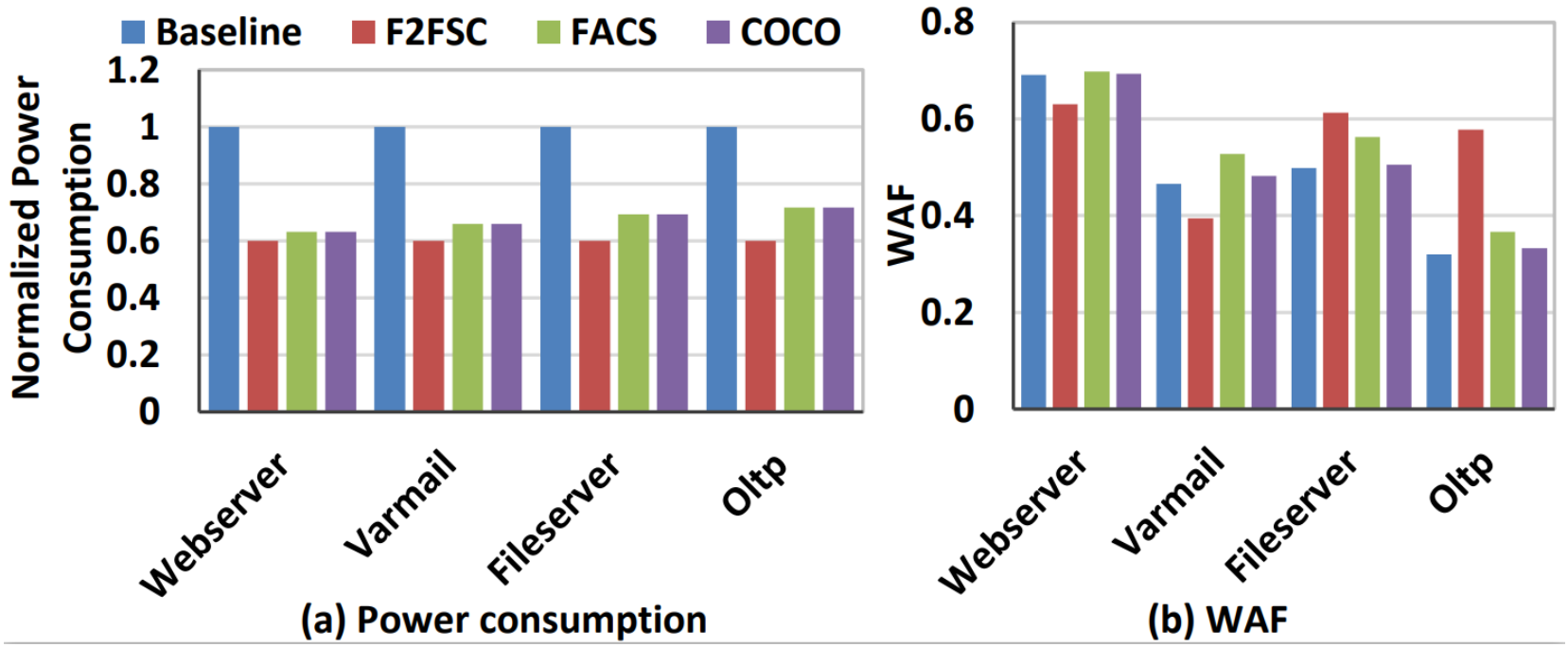
Experimental Results

- Throughput of **F2FSC** was reduced by an average of **42.6%**
- **FACS** slightly improves the performance
- **COCO** can improve the average performance by **31.7%**



Experimental Results

- **F2FSC** has the lowest power consumption
- **FACS** can reduce power consumption by an average of **32.5%**
- WAF of **F2FSC** is improved by **12%** on average
- WAF of **FACS** is improved by **4.5%** on average



Conclusion

- Evaluates the power consumption and performance of CSD
- COCO is proposed to optimize the power consumption of CSD and improve the performance of F2FS
  - Feature-Aware Compression Scheduling
  - On-Demand Space Allocation
- Experimental results show that COCO can improve the performance while reduce the power consumption of CSD

When F2FS Meets Compression-Based SSD!

Thank you!

Questions?

If any questions, please contact us!

Liang Shi shi.liang.hk@gmail.com
Yunpeng Song yunpengsonga@gmail.com