



## CS561 Spring 2026 - Research Project

**Title:** *Analyzing GC and GC-free approaches in ZNS SSDs*

**Background:** ZNS exposes storage as a collection of fixed-size zones that have sequential write requirement. Due to hardware resource constraints the number of open zones is limited (between 8-32) [2,3]. Once the limit is reached, the host application can run the **zone FINISH** command to lock the zone for the writes and indicate to the controller to release the write resources for that zone. From the application perspective the write pointer simply moves to the end of the zone [1]. However, on the controller side the leftover empty blocks in the zone are filled with dummy data [1]. This process leads to device-side write amplification (DLWA) and interference with the host I/O. After the **zone FINISH** command – and once the data in the zone are not needed anymore – the **zone RESET** command reclaims the zone to use it for new writes.

Applications like RocksDB with ZenFS filesystem backend support both approaches where the host can adjust the finish threshold, for example by setting the finish threshold low would allow the applications to run the **zone FINISH** command early, and allocate a new zone if the lifetime of the data does not match the currently open zones [2]. With this approach the application could run with zero host-side garbage collection. However, this comes with the cost of DLWA. The other approach is to delay the **zone FINISH** command and mix data with the currently open zone even if the lifetime does not match perfectly. This would lead to space-amplification as valid and invalid data could be mixed in the zone, but then the host can trigger garbage collection (GC) to reclaim the space. However, this process could interfere with the throughput.

**Objective:** The objective of the project is to study the tradeoffs between host-triggered garbage collection and zero host-side GC by adjusting the zone finish threshold considering the implication on throughput, space amplification and DLWA.

### Steps:

- Understand the architecture of the ZNS SSD and configuration options of ConfZNS++ emulator [1]
- Evaluate the performance of RocksDB with ZenFS filesystem backend by adjusting the finish threshold and enabling the host-triggered garbage collection
- Try different configurations of the ZNS device such as small-zoned SSD vs large-zone SSD

**Responsible mentor:** *Teona Bagashvili*

### References

- [1] Doekemeijer, Krijn, et al. "Exploring I/O Management Performance in ZNS with ConfZNS++." *Proceedings of the 17th ACM International Systems and Storage Conference*. 2024.
- [2] Bjørling, Matias, et al. "ZNS: Avoiding the block interface tax for flash-based SSDs." *2021 USENIX annual technical conference (USENIX ATC 21)*. 2021.



- [3] Song, Inho, et al. "Confzns: A novel emulator for exploring design space of zns ssds." *Proceedings of the 16th ACM International Conference on Systems and Storage*. 2023.