Virtual Guard: A Track-Based Translation Layer for Shingled Disks

MANSOUR SHAFAEI & PETER DESNOYERS

NORTHEASTERN UNIVERSITY

Outline

- Introduction
- Previous work
- Virtual Guard
- Evaluation
- Conclusion

Shingled Disks

- Partially overlapping tracks for more capacity

- Random writes may corrupt data on the next track in shingling direction
  - Need a translation layer to map data to a location with no/invalid data ahead

Shingling Translation Layers (STLs)

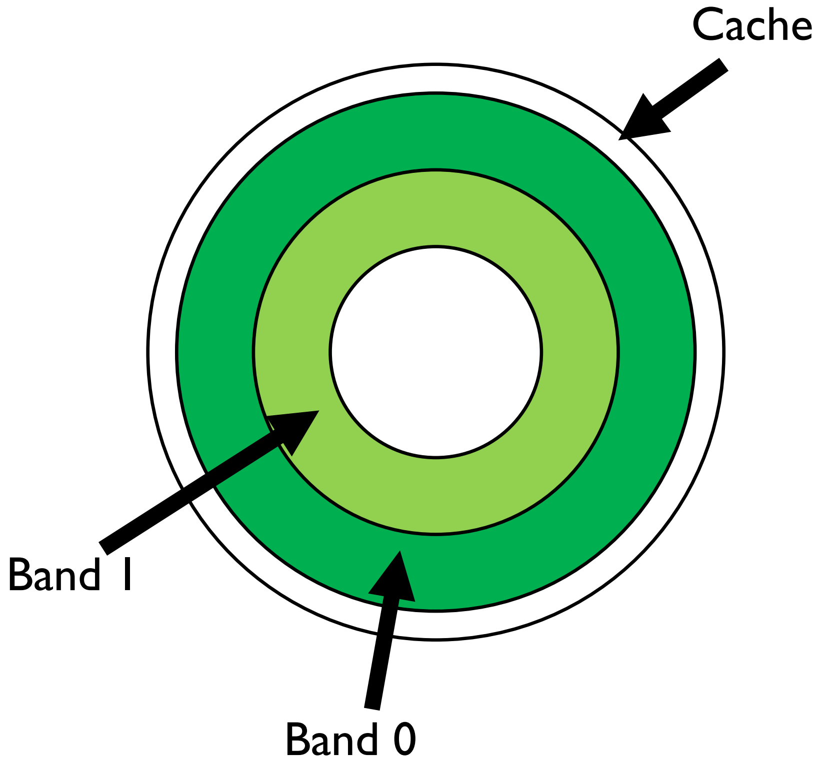
- Mapping type
  - Static
    - Band + persistent cache (Read-Modify-Write)
  - Dynamic
- Mapping granularity
  - LBA based → DM-SMR [FAST'15]
  - Track based → SMaRT [He & Du]
- Mapping location
  - Host
  - Device → plug compatibility
  - Host+Device

STL Issues

1. Large mapping tables
   - Requiring multiple rotations to persist
2. High cleaning latencies
   - Sensitive to utilization
3. Not handling track size differences
   - OD to ID, adaptive formatting or slip sparing

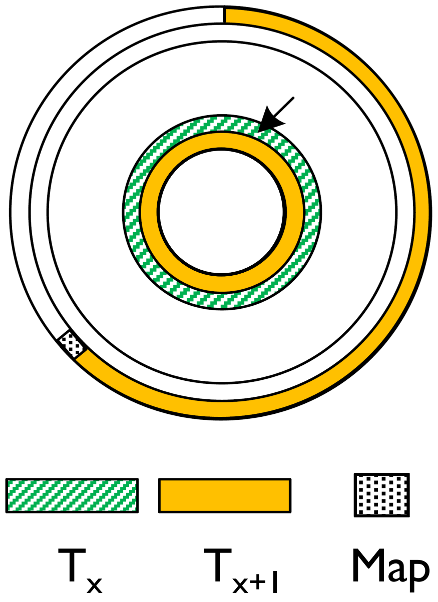
"Traditional" STLs (DM-SMR)

- Space divided into multiple bands
- A persistent cache
- Updates go to cache
- Cache cleaning with one band at-a-time



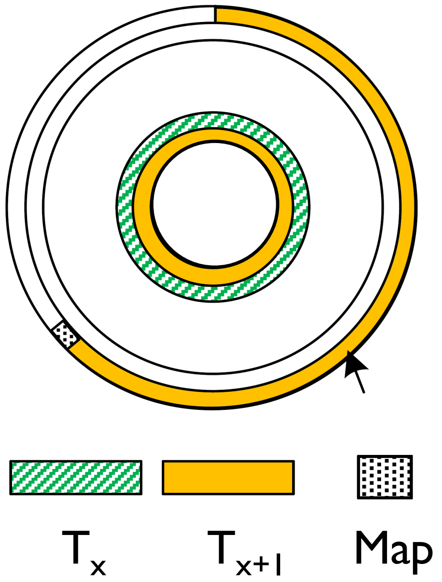
Virtual Guard

- A track-based shingling translation layer
- Static mapping with a cache at outer diameter
- Caches the next track in shingling direction before any updates
- Writes in-place



Virtual Guard (Cont.)

- Treating cached tracks the same
  - Relocating the next track to the WF and then write in-place
- On-demand FIFO based 2-band cleaning
- Extremely small map size (<30K for a 5TB drive)
  - Per track info for tracks in cache
  - Persistent cache at outer diameter (Biggest tracks)
  - Piggyback the map info on track that was copied



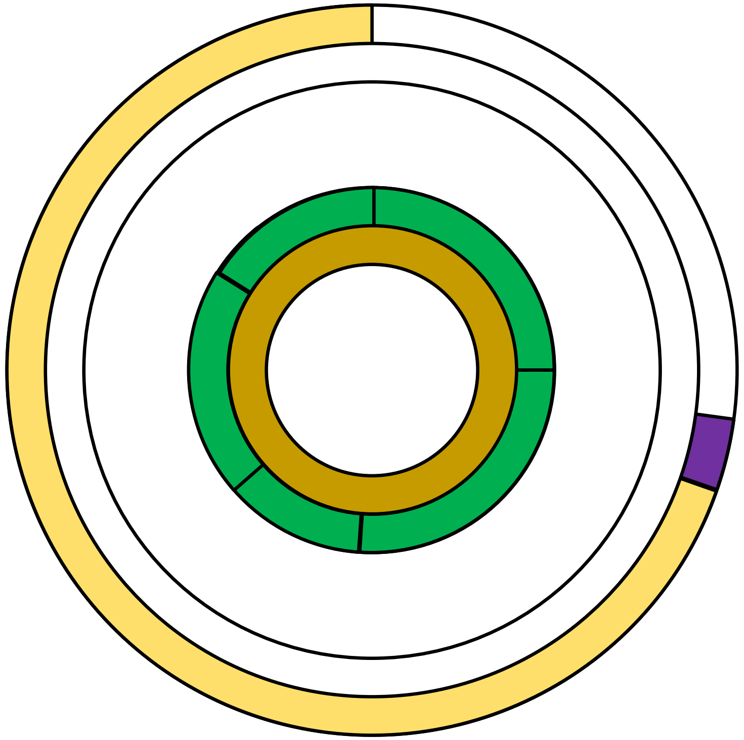
Virtual Guard (Cont.)

- Less number of cleanings due to track level write locality
  - Cache usage not a function of number of writes any more

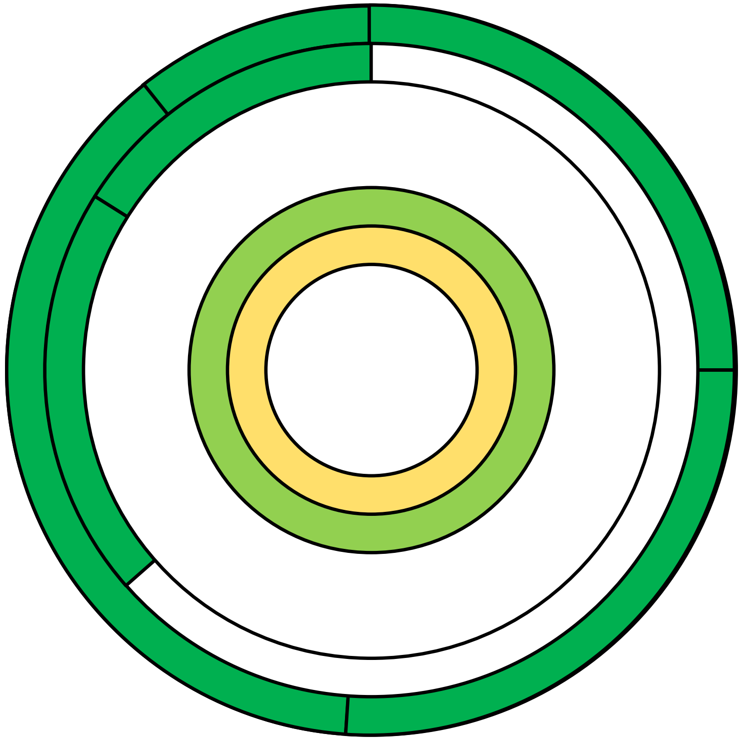
- Low cleaning overheads
  - Reading tracks as oppose to scattered updates

Demo

Virtual Guard



Traditional STLs

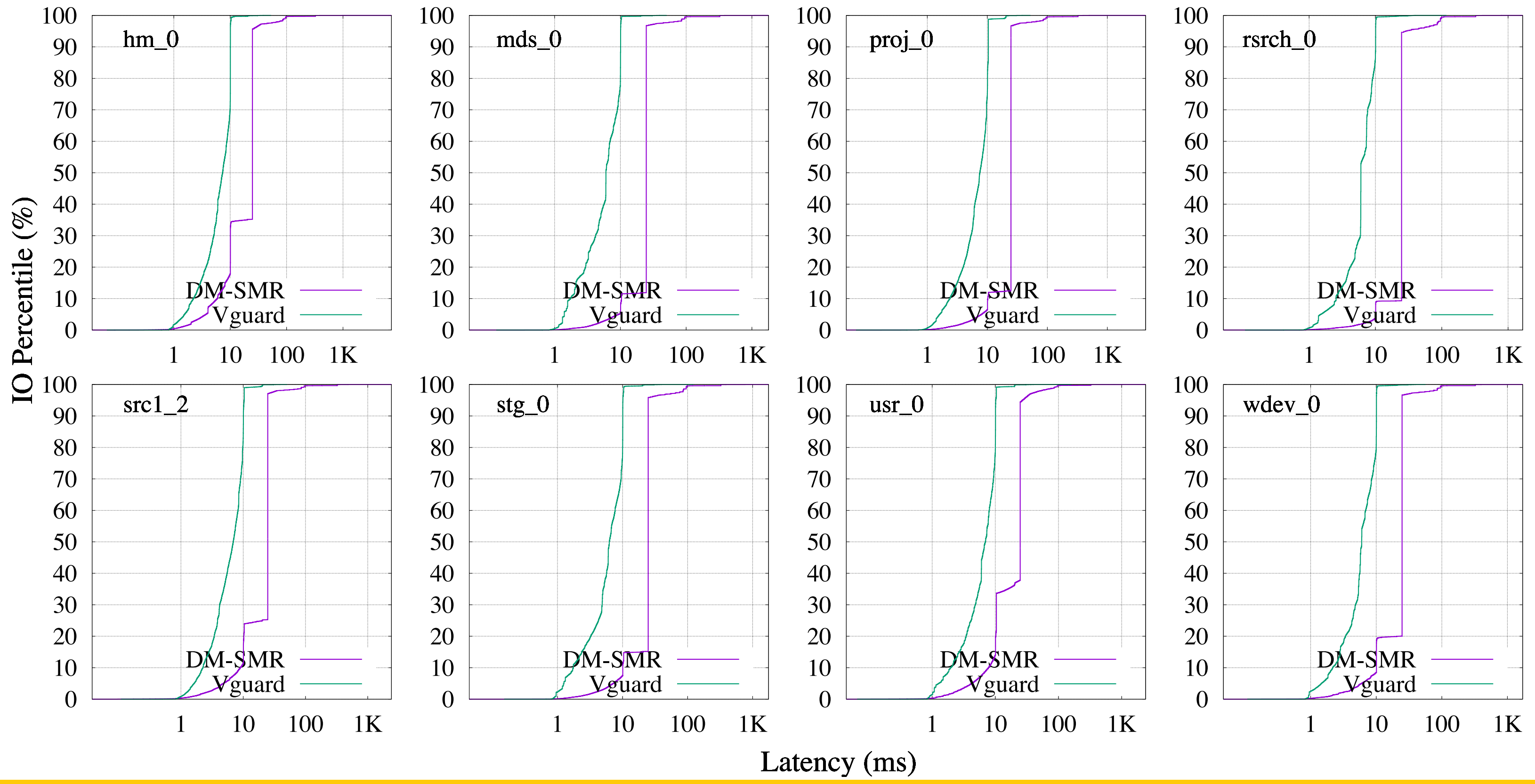


Evaluation

- Implemented in an accurate SMR simulator
- Compared to DM-SMR with identical characteristics
  - Form factor: 3.5”
  - Size: 5TB
  - RPM: 5980
  - Track size: 1.8-0.9 MB
  - Mapping type: static
  - Band size: 20 tracks
  - Cache size: ~24GB
  - Cache location: outer diameter
  - Map size: ~30K vs ~1.3MB
  - Cleaning thresholds: 9194 vs. 22986
- Traces → MSR Cambridge, CloudPhysics and random writes

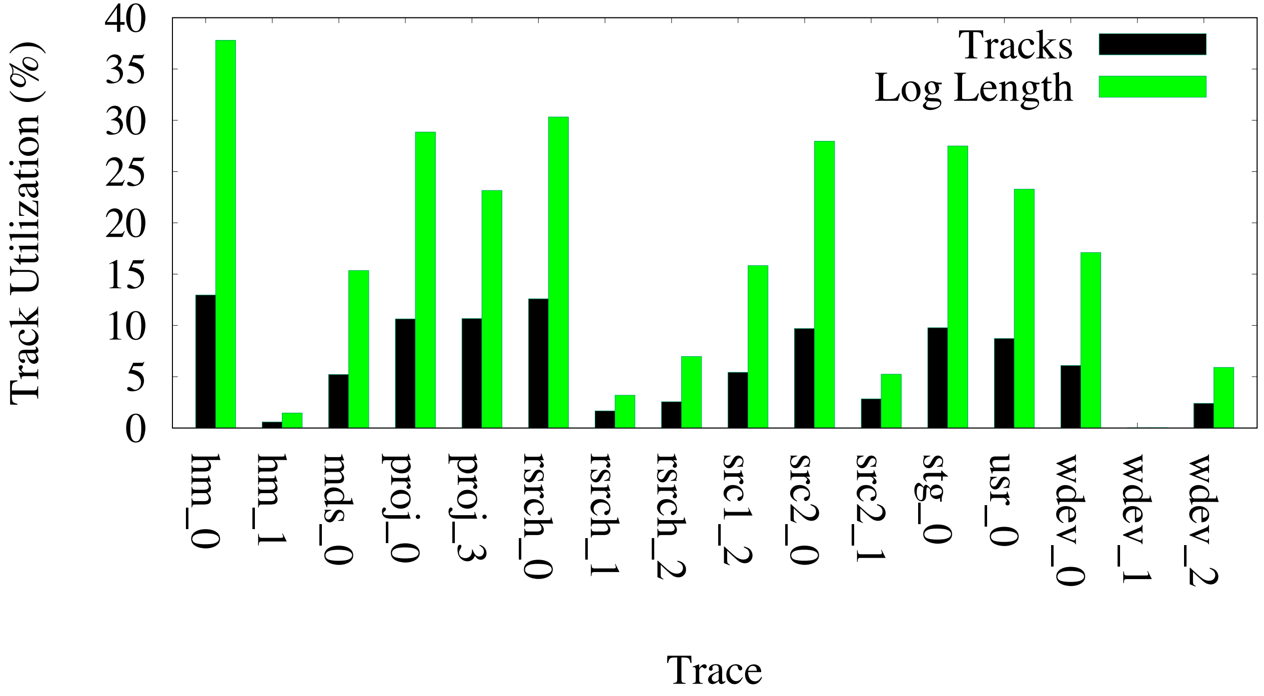
MSR Block Traces

- Up to 15X reduction in 99.9th percentile latency



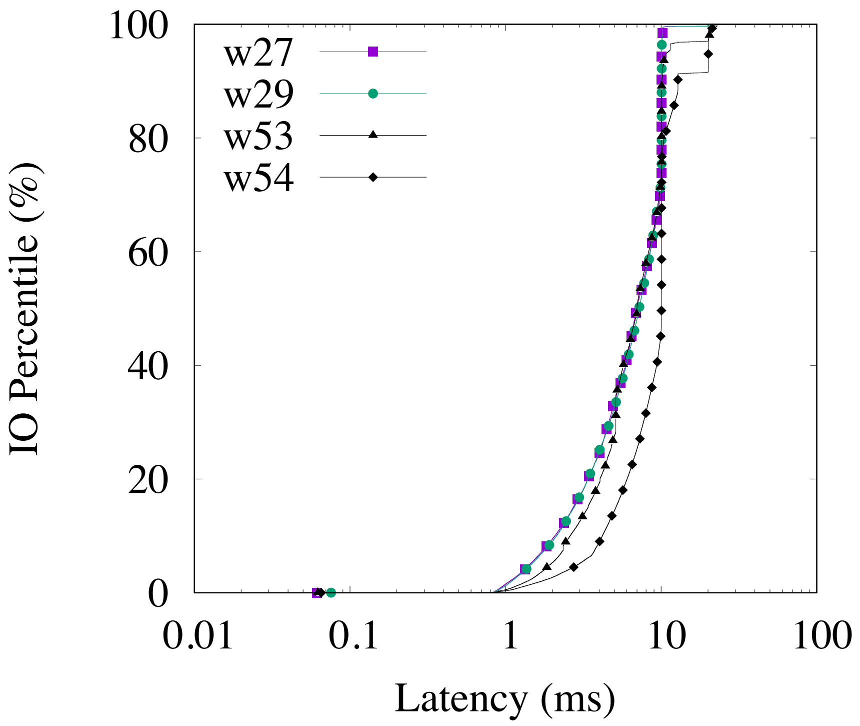
MSR Traces -- Cache Utilization

- Less than 40% in terms of log length

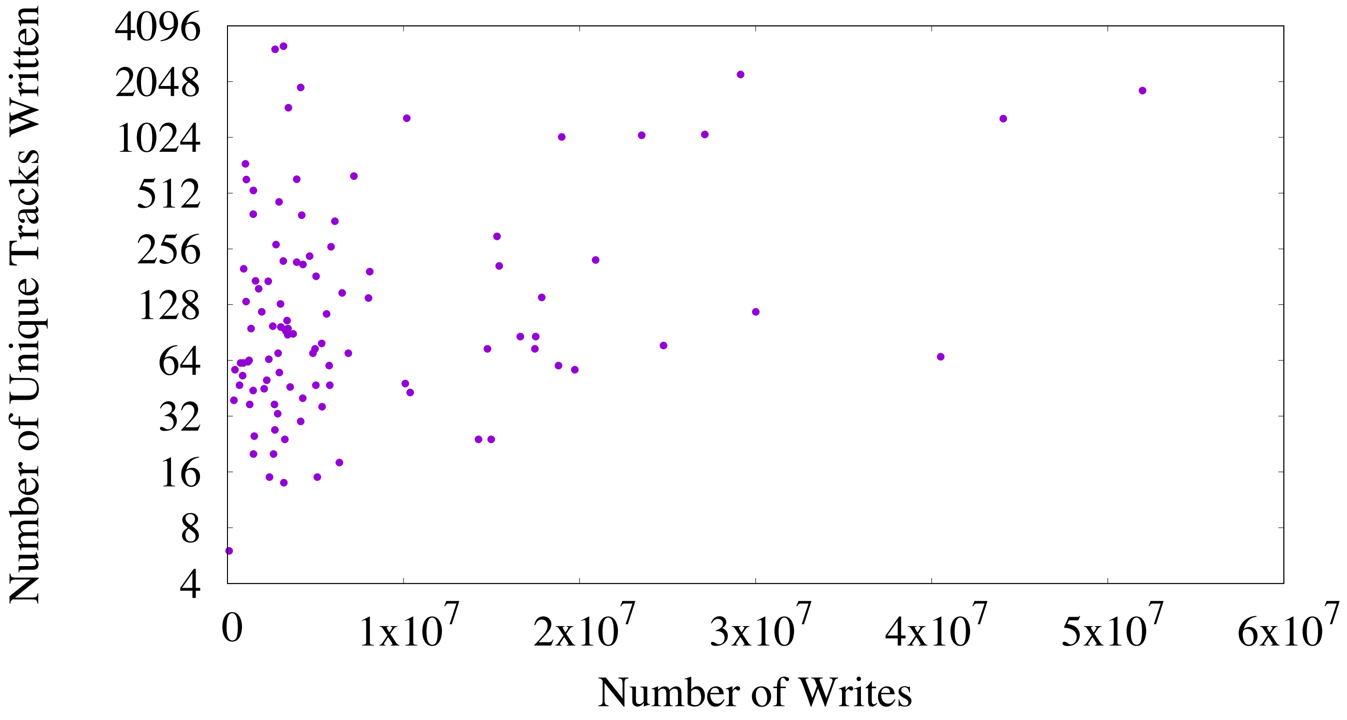


CloudPhysics Traces

- Traces on multi TB drives

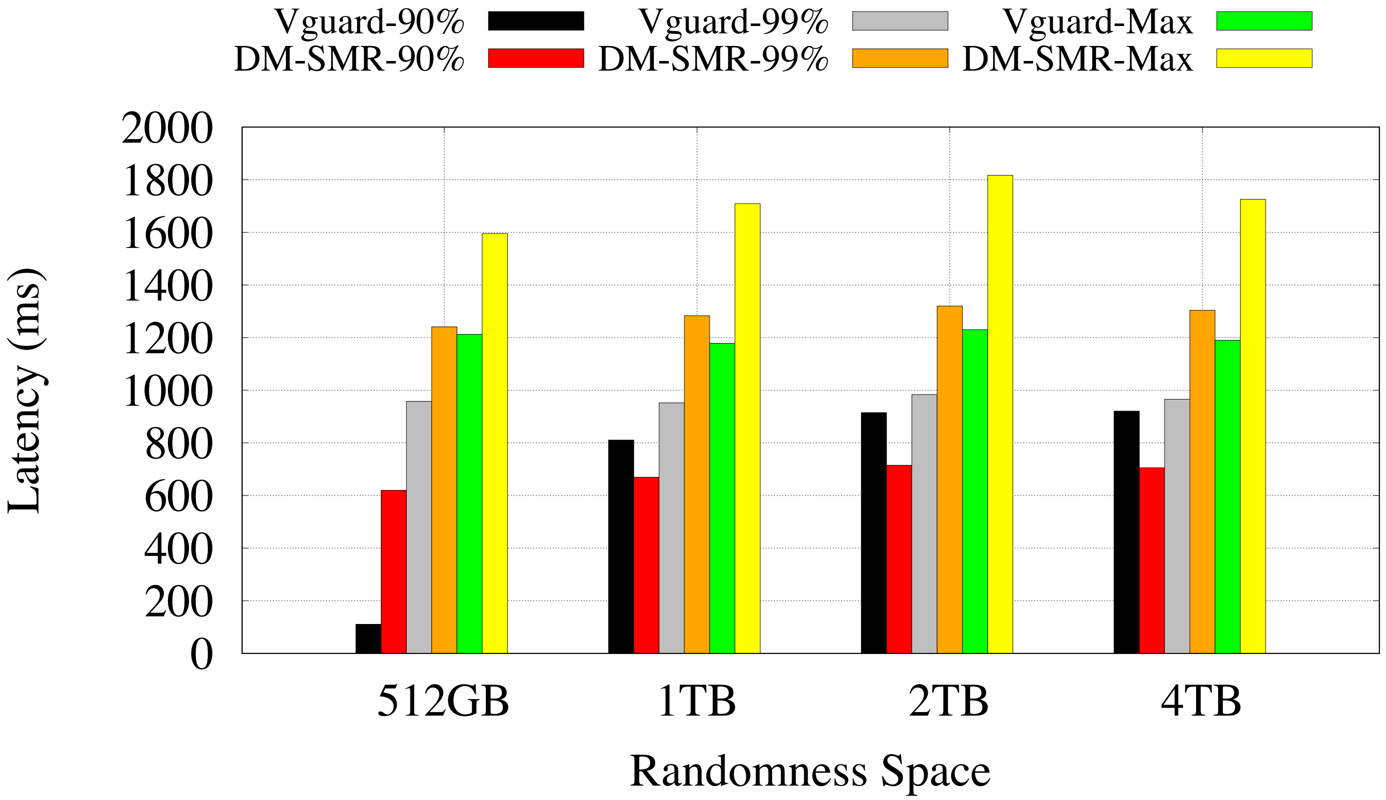


- Track utilization across all traces



Random Writes

- ~30% reduction in 99th percentile and max latency



Conclusions

- Vguard represents a novel approach to STLs, using persistent cache space for non-written tracks while performing writes in-place.
  - Cache consumption not a function of the volume of data written, but rather of the pattern of written LBAs regardless of the number of times they are written

- In many real-world cases the guard track set is seen to fit comfortably within a rather small persistent cache
  - Offering near-conventional-drive levels of performance.