Design Tradeoffs for SSD Performance

Nitin Agrawal
University of Wisconsin-Madison

Vijayan Prabhakaran,
Ted Wobber, John D. Davis,
Mark Manasse, Rina Panigrahy
Microsoft Research, Silicon Valley

Introduction

- SSDs have HUGE potential to revolutionize
  - Semi conductor memory (no moving parts)
  - High: performance, power savings, reliability

- Challenges in realizing the potential
  I. Non-overwrite
  II. Limited serial bandwidth
  III. Limited lifetime (erasures)

Our Work

- Goals
  - Address the challenges by exploring various designs
  - Understand the impact on performance/lifetime
- Built an SSD simulator
  - Extended from CMU's DiskSim
- Trace-based evaluation
  - Traces: Synthetic, FS & DB benchmarks, Exchange

Results

- Key takeaway
  - Numerous tradeoffs
  - Performance/lifetime: arch, algorithm, workload

- Specific takeaways
  I. Non-overwrite: allocation & layout
  II. Limited serial bandwidth: interleaving
  III. Limited erase cycles: wear-leveling

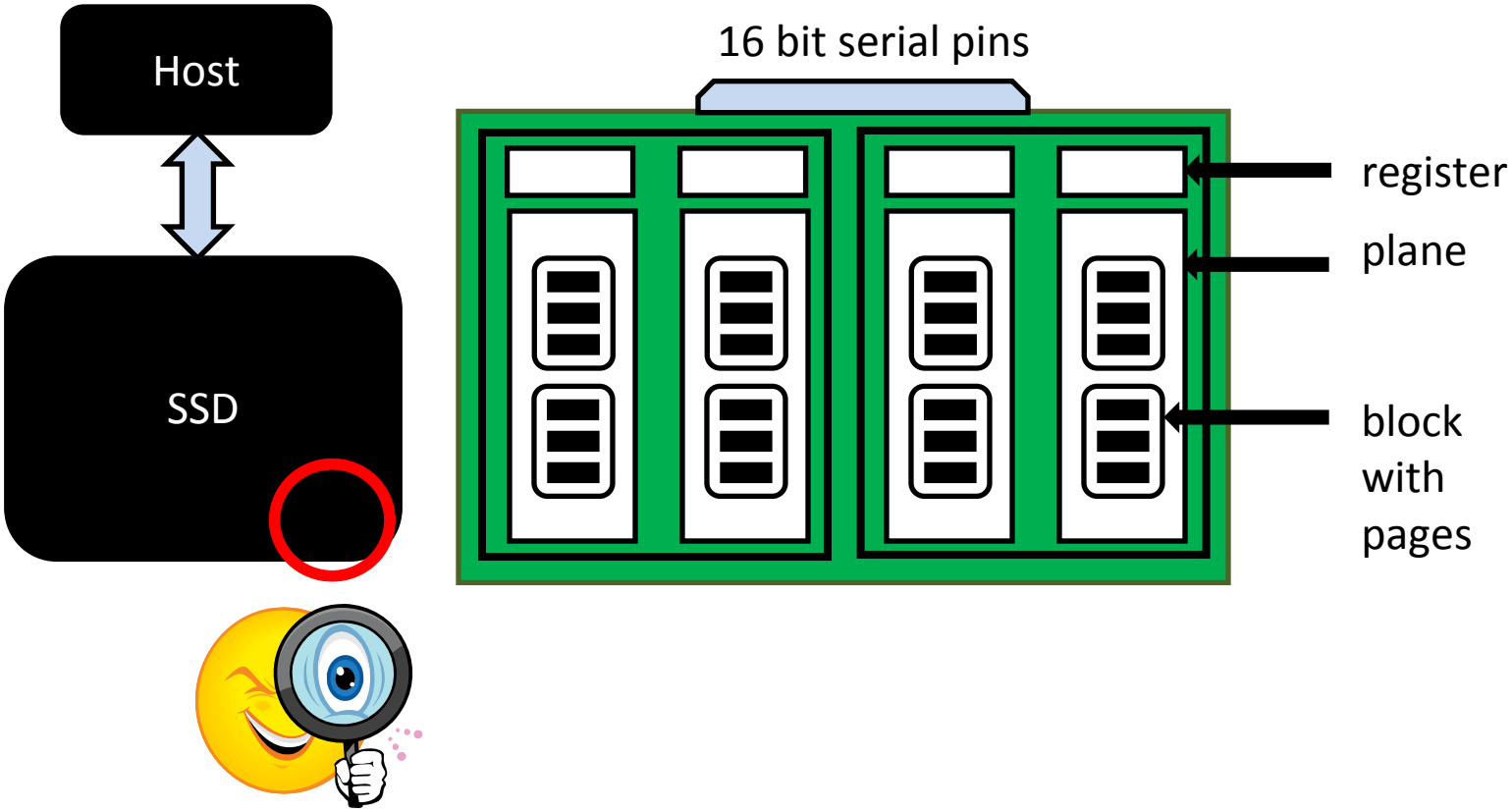
~~Introduction~~

Architecture

Challenges

Conclusion

SSD Architecture



~~Introduction~~

~~Architecture~~

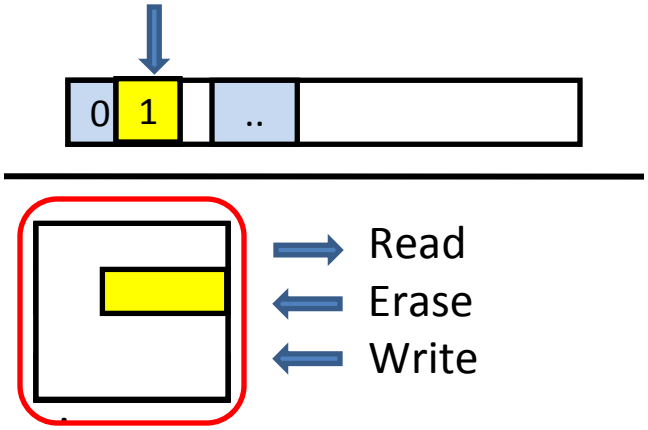
Challenges

Conclusion

I. Non-overwrite

Allocation: Direct Mapping

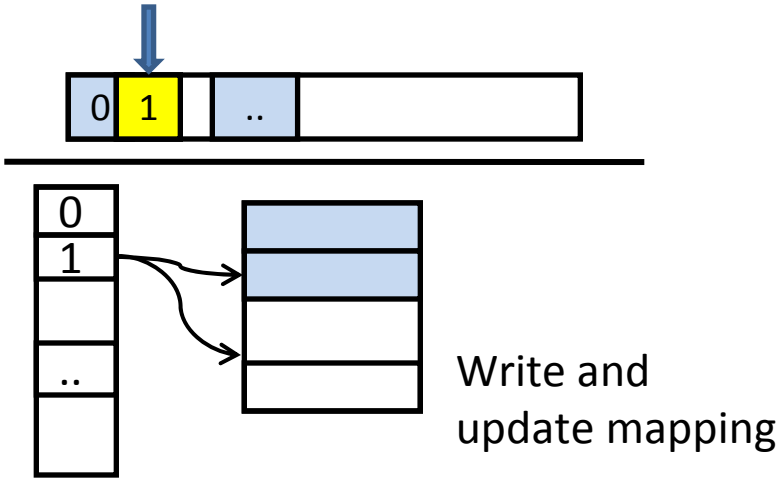
- Direct mapping
  - Write: read-modify-erase-write



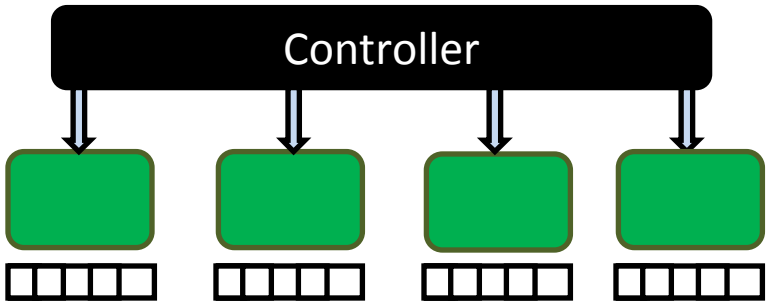
- Random writes: high overhead
  - ~ 270 random IOPS on a 32 GB

Allocation: Log Structured

- Log structured design
  - High mapping cost
  - Cleaning in background



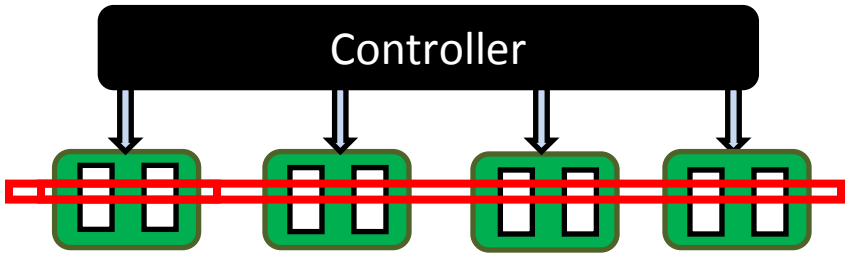
- Multiple Logs
  - One per package
  - Cleaning in parallel with foreground I/O
  - 18.5 K IOPS



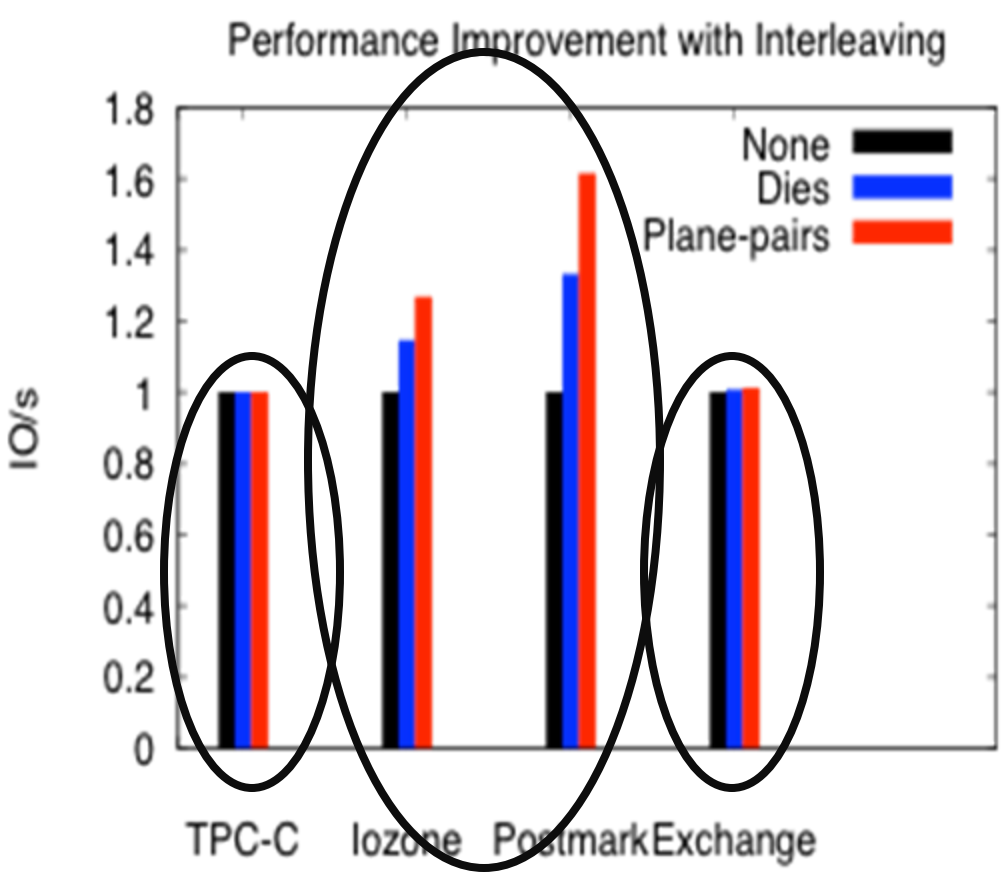
II. Limited Serial Bandwidth

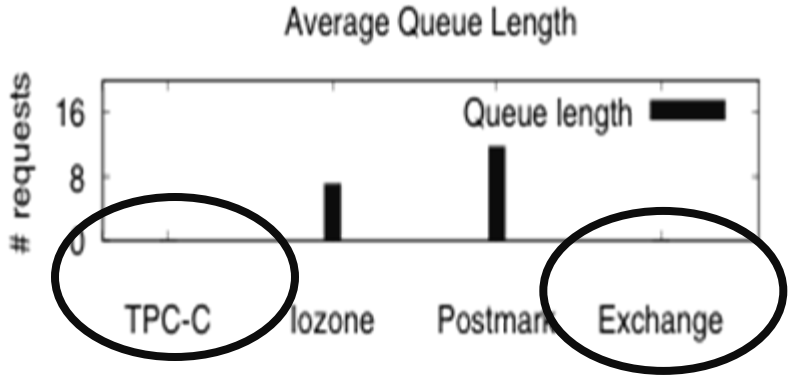
Exploiting parallelism: Interleaving

- Inherent parallelism : multiple packages, dies, planes
  - Stripping across and within packages



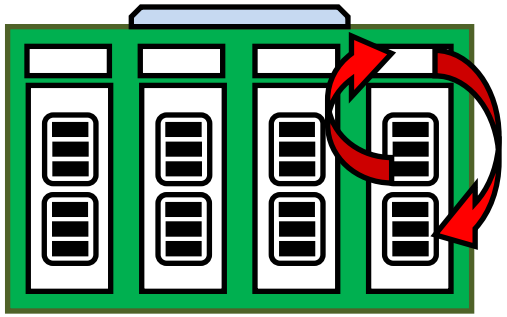
Interleaving Within Package





Copy-back

- Copy-back : copy pages within a flash package
  - Cleaning and wear-leveling



Workload	Improvement with Copy-back	Efficiency
TPC-C	40%	70%
Iozone	0%	100%
Postmark	0%	100%

III. Limited Erase Cycles

Block Usage

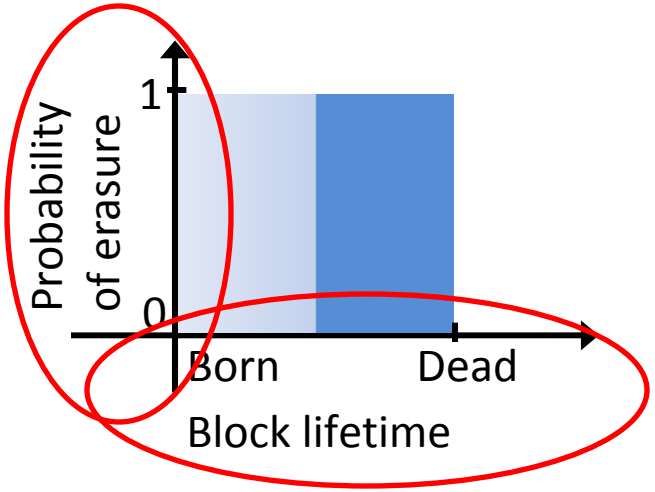
- Flash blocks have limited lifetime
  - Fixed number of erasures

- Greedy cleaning
  - Choose blocks with best cleaning efficiency

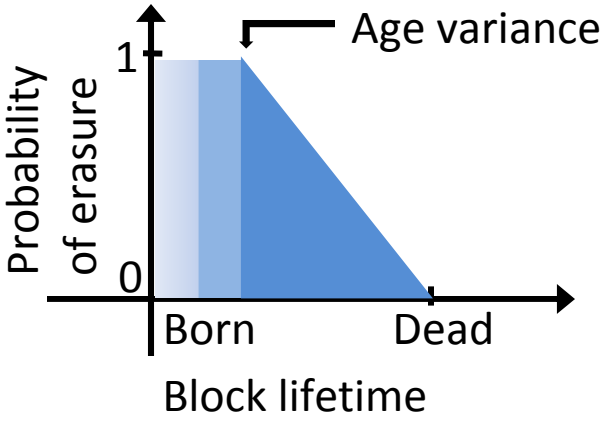
- Goal: use all blocks uniformly

Wear-leveling

- Greedy cleaning
  - Irrespective of remaining lifetime



- Wear-leveling
  - Rate-limiting factor
  - Probabilistically reduce erasures
  - Cold data migration



~~Introduction~~

~~Architecture~~

~~Challenges~~

Conclusion

Conclusion

- Understand design tradeoffs for SSD performance
  - Trace-based simulator
- Tradeoff across all dimensions
  - Architecture, Algorithms, Policies, Workload
- WALL-I
  - Wear-leveling, ALlocation and Layout, Interleaving

Thanks!