http://clyde.itu.dk
A project funded by the Danish Council for Independent Research

Host-SSD Co-Design:

Some Lessons Learnt and Perspectives

Philippe Bonnet – phbo@itu.dk
IT University of Copenhagen

Joint work with Luc Bouganim (INRIA), Niv Dayan (ITU), Matias Bjørling (ITU), Jesper Madsen (ITU)

In Collaboration with Jens Axboe (Facebook), David Nellans (Nvidia), Zvonimir Bandic (HGST), Qingbo Wang (HGST), Aviad Zuck (Tel Aviv Univ.),
Michael Wei, Steve Swanson (UCSD), Dennis Shasha (NYU), Björn Thòr Jònsson (RU)


LAB4

DATABASE TUNING

Principles, Experiments, and Troubleshooting Techniques

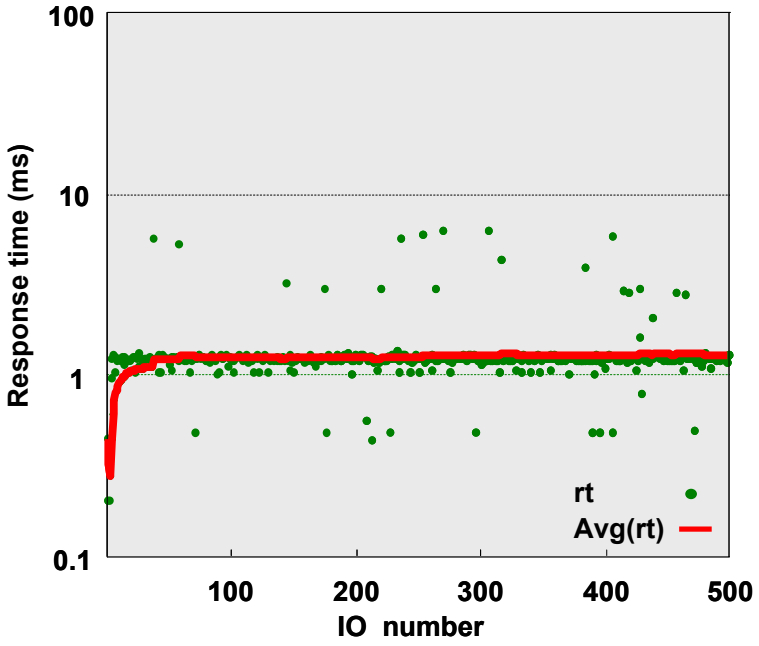
DENNIS SHASHA & PHILIPPE BONNET

FOREWORD BY JIM GRAY (MICROSOFT)

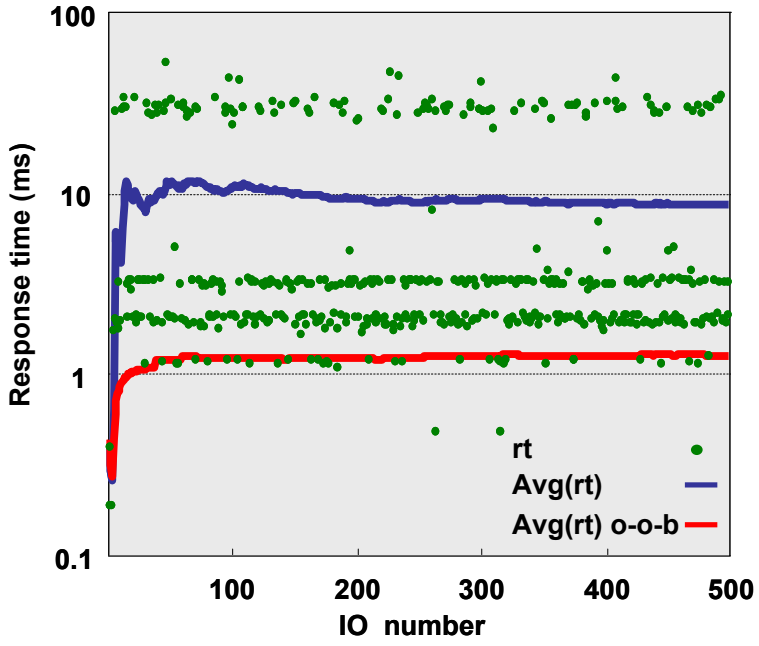
CIDR 2009

Methodology (1): Device state

- Measuring Samsung SSD RW performance
  - Out-of-the-box … and after filling the device!!! (similar behavior on Intel SSD)



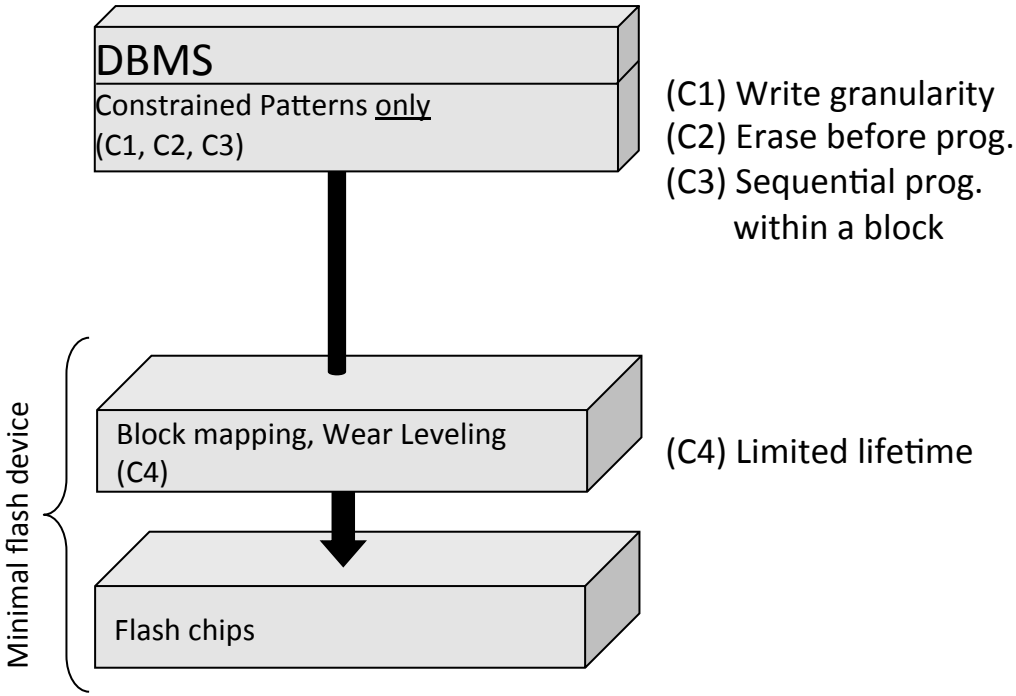
Random Writes – Samsung SSD
Out of the box



Random Writes – Samsung SSD
After filling the device

Minimal FTL: Take the FTL out of the equation!

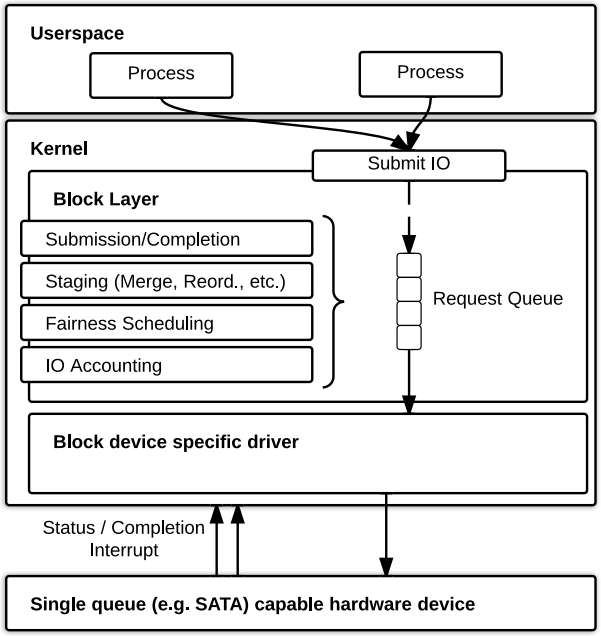
- **Pros**
  - Maximal performance for
    - SR, RR, SW
    - Semi-Random Writes
  - Maximal control for the DBMS
- **Cons**
  - All complexity is handled by the DBMS
  - All IOs must follow C1-C3
    - The whole DBMS must be rewritten
    - The flash device is dedicated

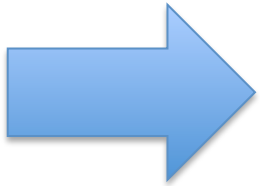


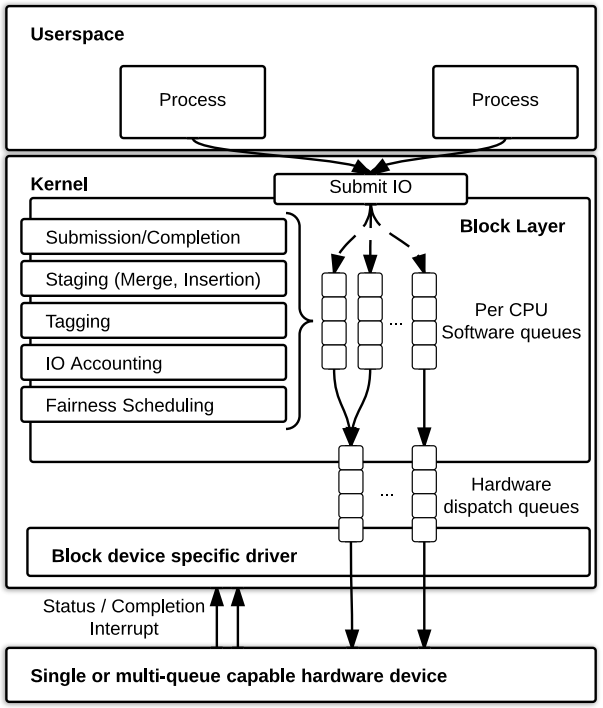
Luc Bouganim, Philippe Bonnet

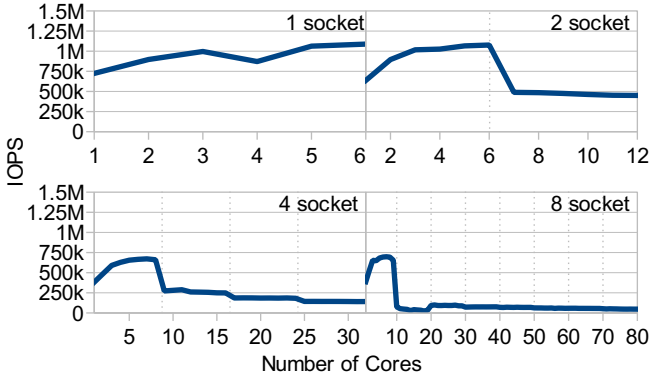
Systor 2013

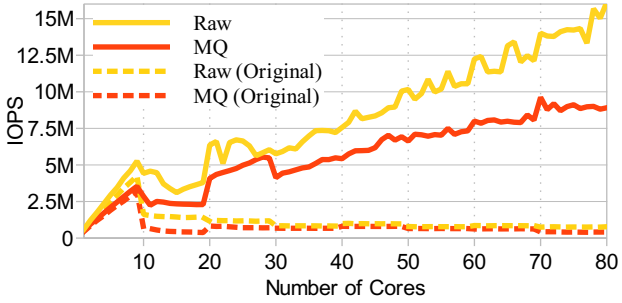
Linux Block Layer: Introducing Multiqueue SSD Access on Multi-core Systems











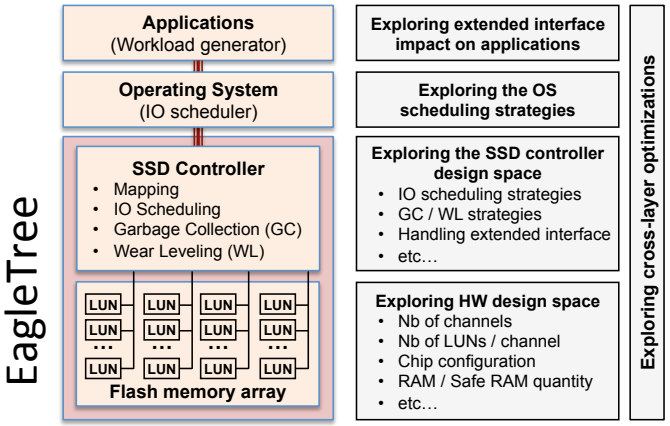
Matias Bjørling, Jens Axboe, David Nellans, Philippe Bonnet

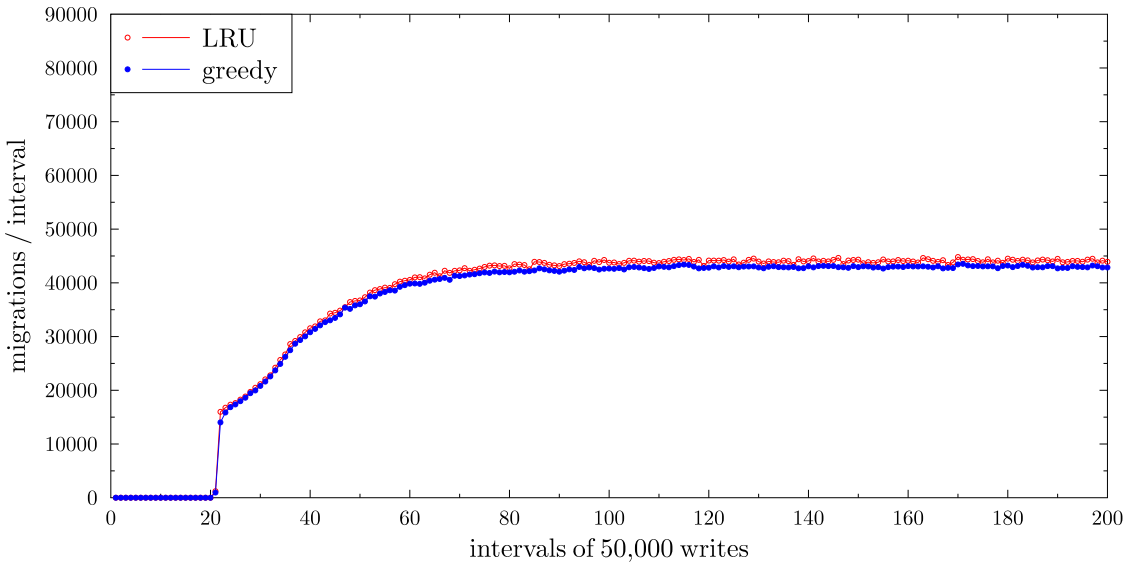
CLyDE Project

- Exploring the design space for Host-SSD co-design:
  - EagleTree [Niv Dayan]
    - Simulated SSD/OS/apps for broad exploration of design space (discrete event simulation)
    - Insights about GC, DB indexing
    - http://github.com/ClydeProjects/EagleTree
  - LightNVM [Matias Bjørling]
    - Host-side SSD management to experiment with actual OS/ apps (wall time clock)
    - Linux support for Open-channel SSDs
    - http://github.com/ MatiasBjorling/LightNVM

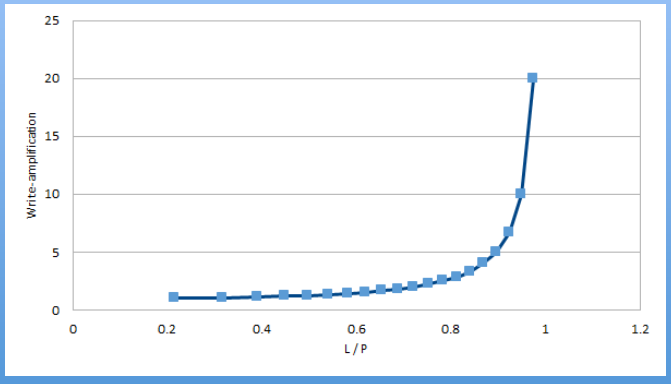
https://github.com/ClydeProjects/EagleTree

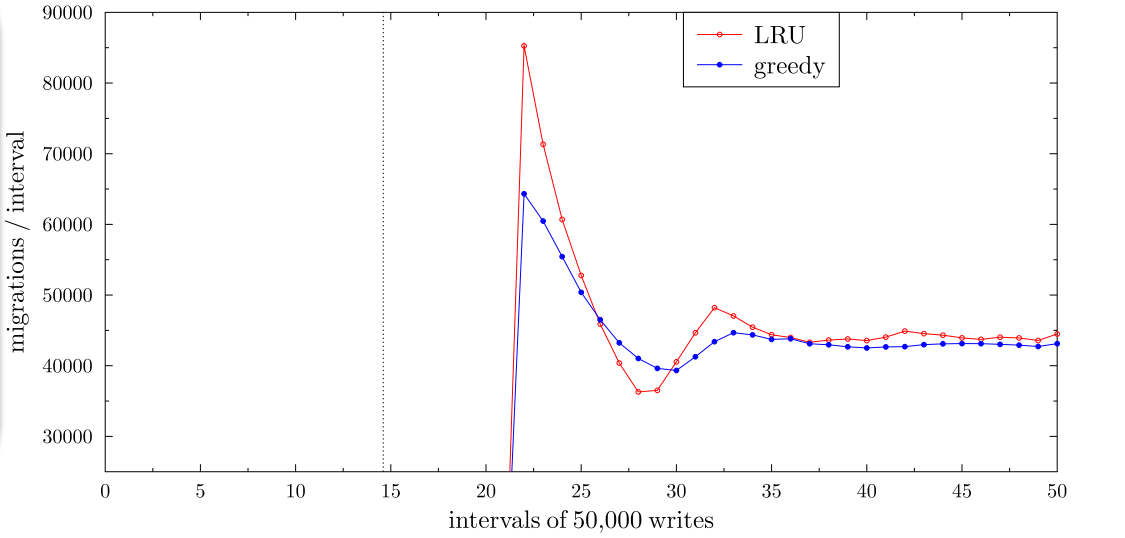
New Insights about GC policies





Closed-form equation for write-amplification

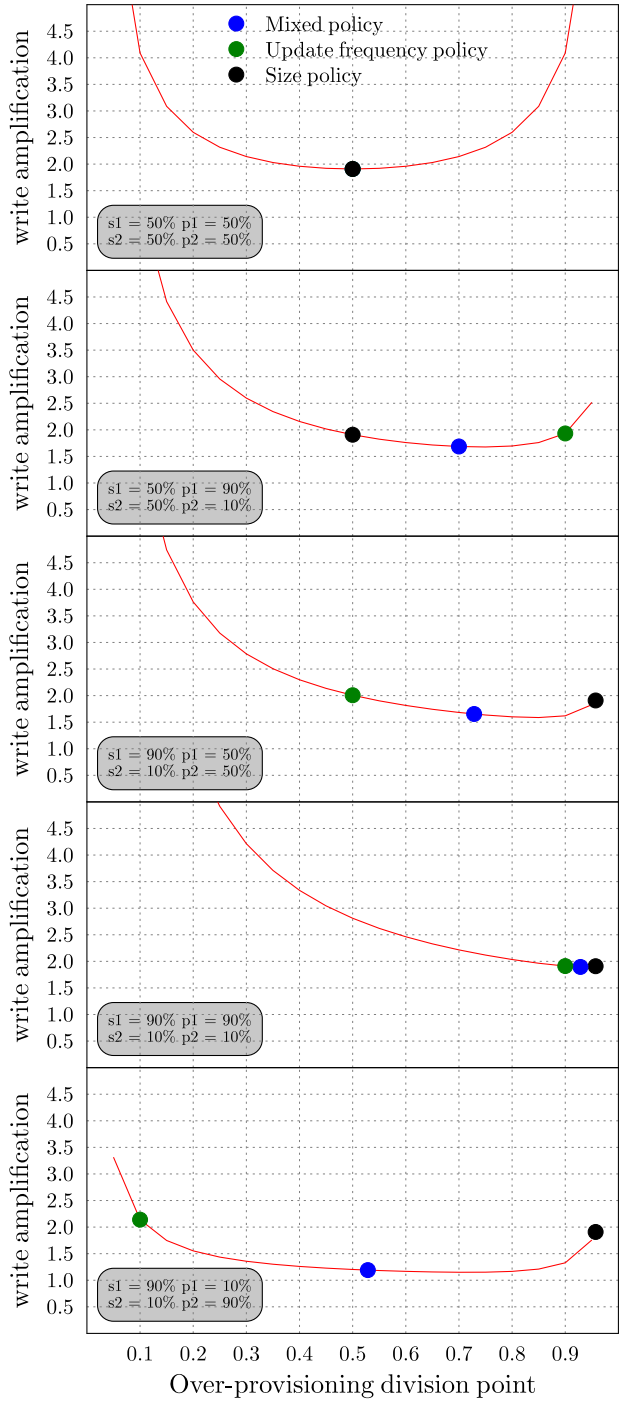


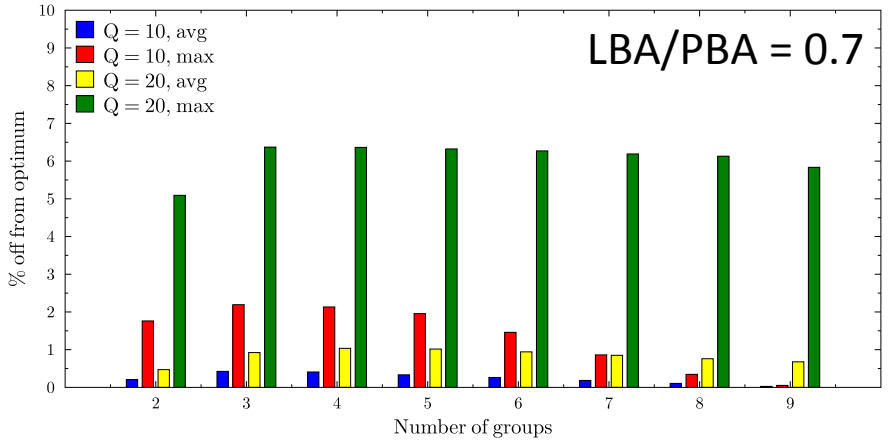


More Insights about GC Policies

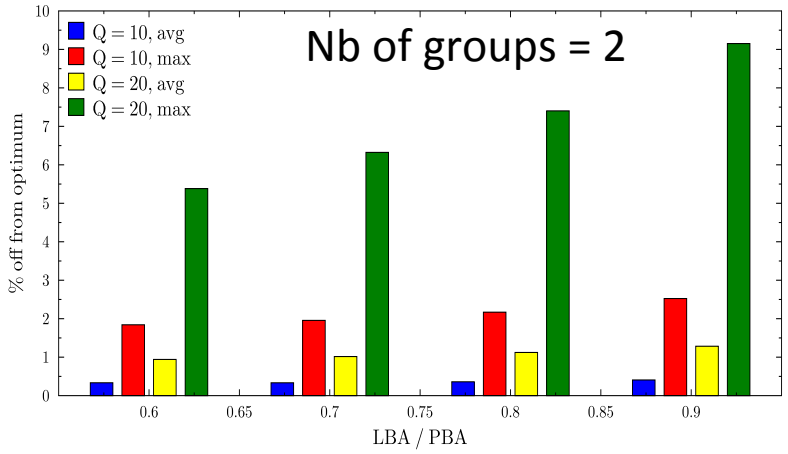
- K-modal workload:
  - Data grouped based on update frequency
  - Previous work:
    - Each flash block is dedicated to a single group
  - Questions:
    1. How to partition over-provisioned blocks across groups?
    2. How to deal with changing update frequencies?

Partitioning Over-provisioned Blocks





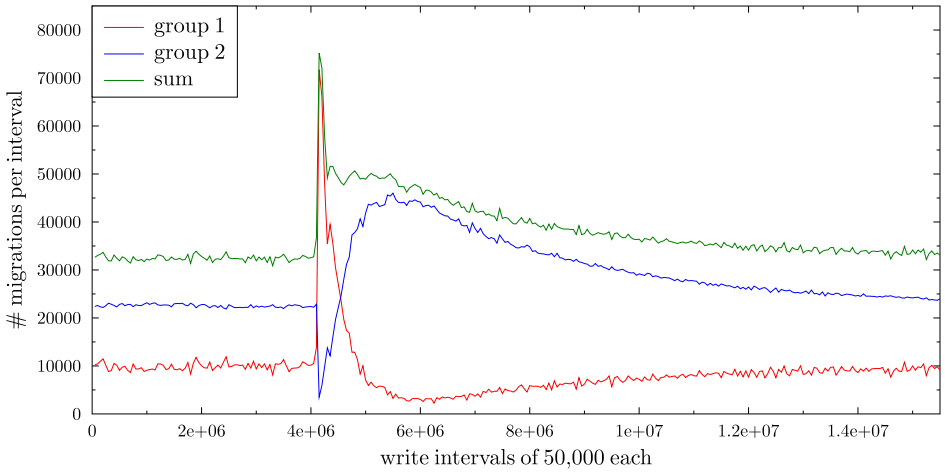
Q: number of distinct update frequencies in the workload

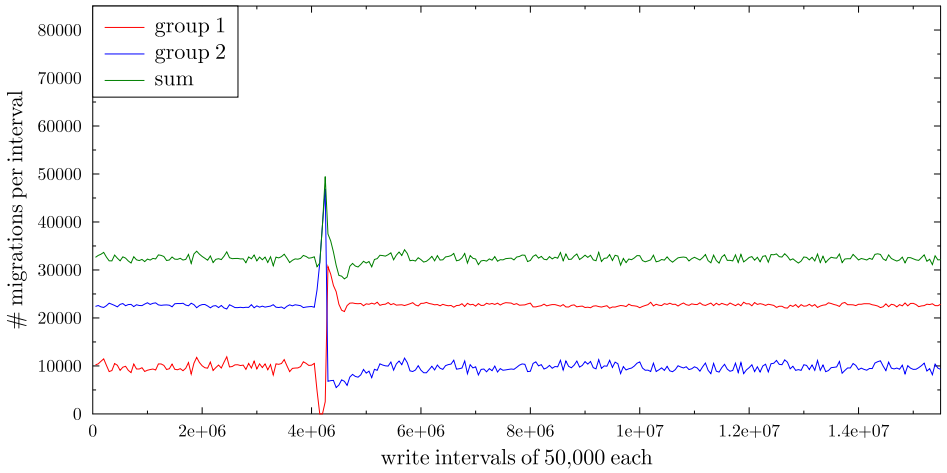


Adapting to Changing Workloads

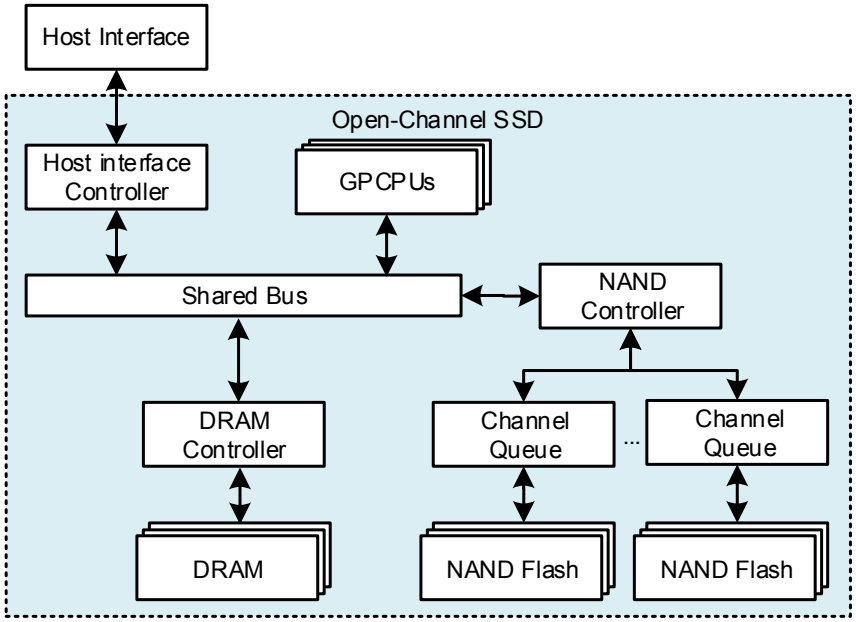
WOLF:

- dynamically measures update frequency
- adapts the number of groups
  - Too few/many groups harm GC efficiency
- triggers garbage-collection aggressively to re-distribute over-provisioned space across groups (from cooling to heating groups)



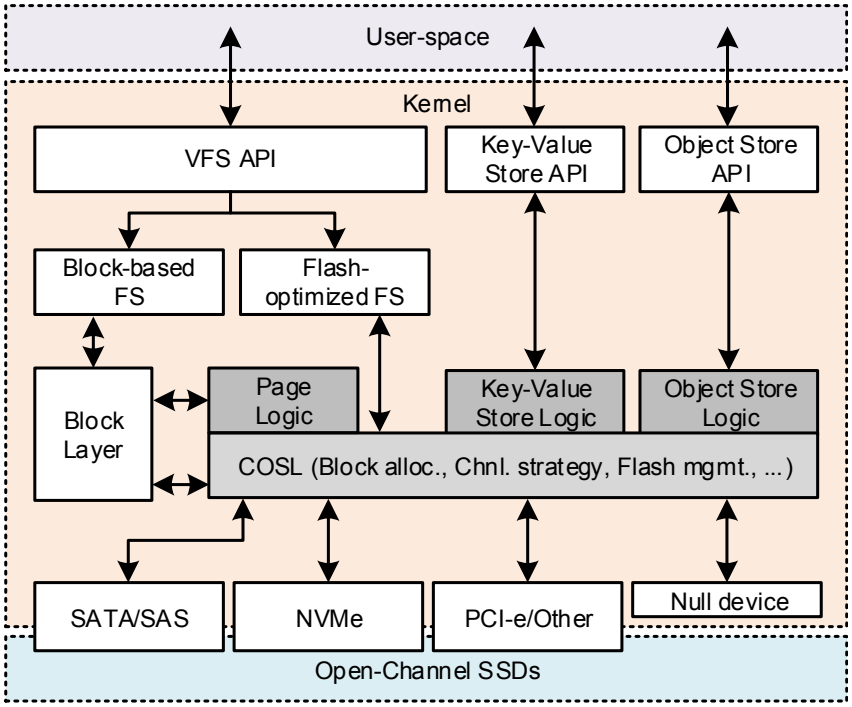


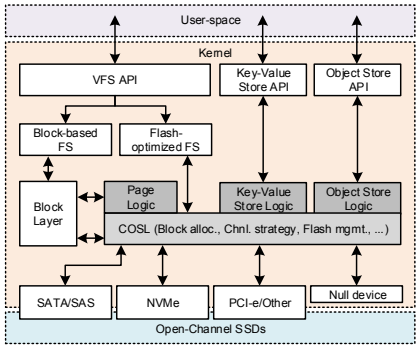
Linux Abstractions for Open-Channel SSDs



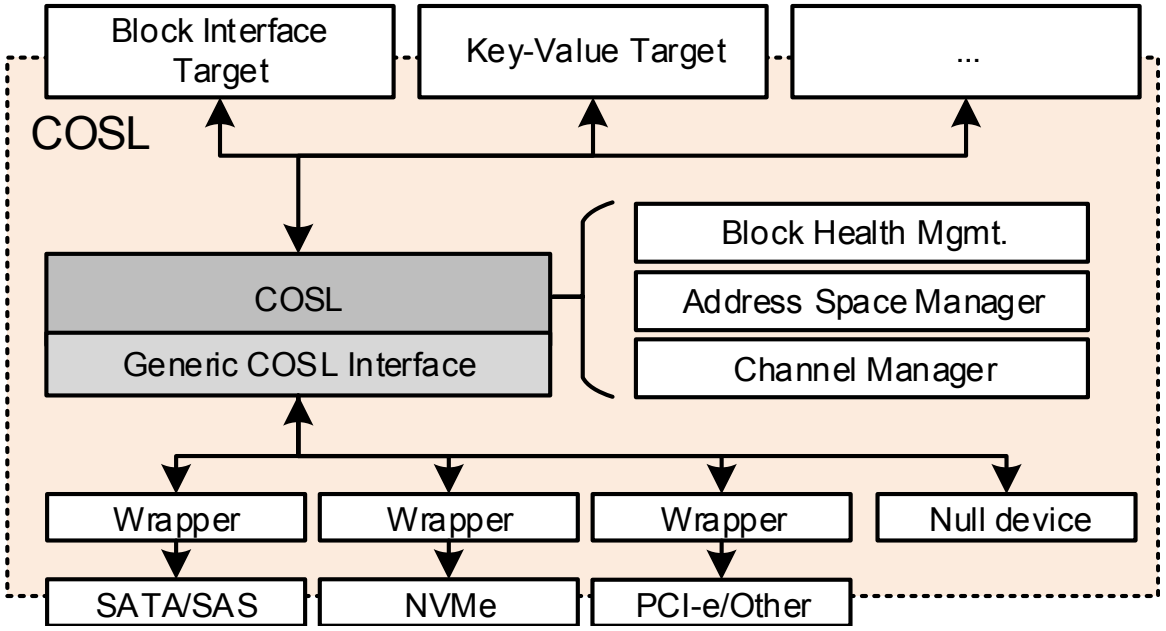
- Extensible
  - Single level of mapping between Applications and physical storage
- Modular
  - SSD Management components should be replaceable
- Low overhead
  - Host-side SSD management should not get in the way of performance, still provide consistency, durability

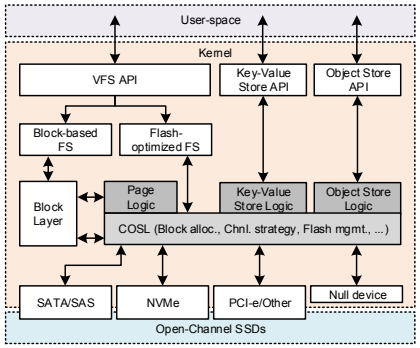
LightNVM Design (1/3)



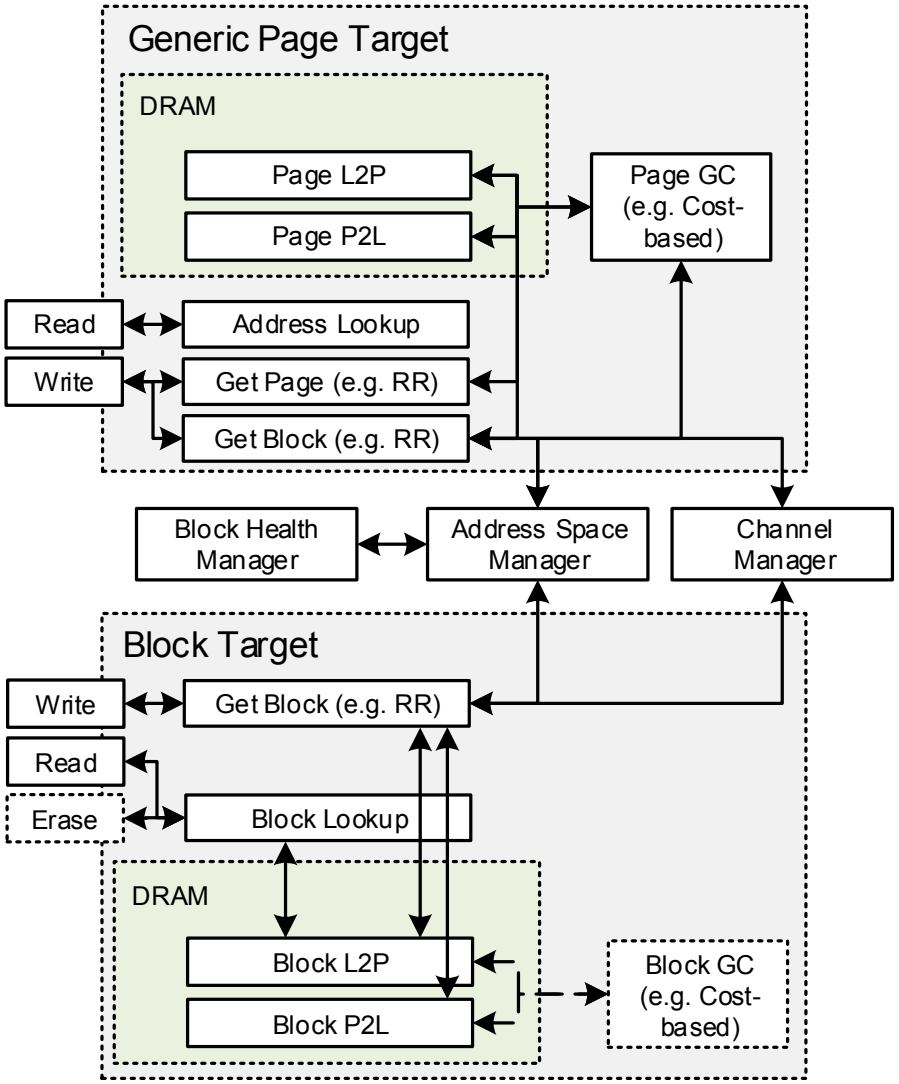


LightNVM Design (2/3)

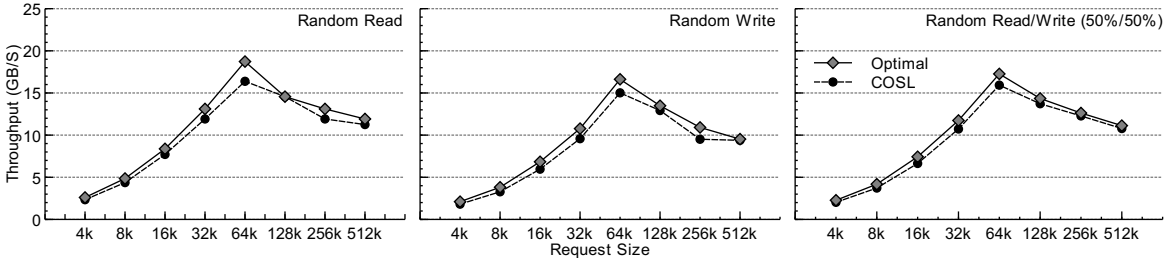


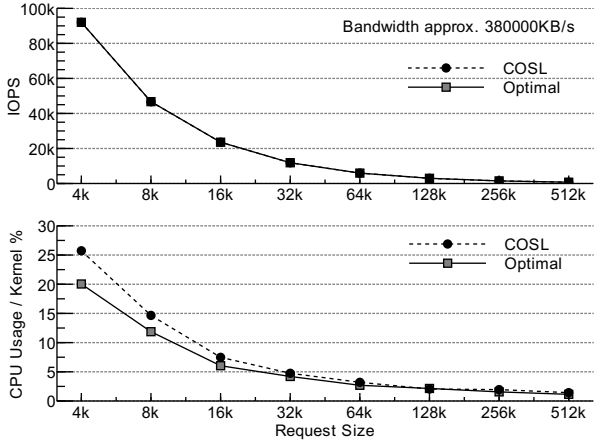


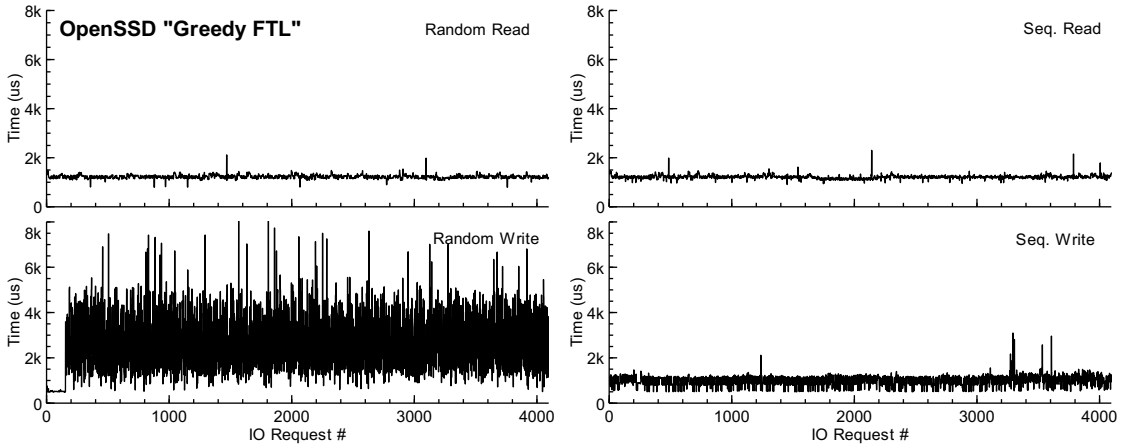
LightNVM Design (3/3)

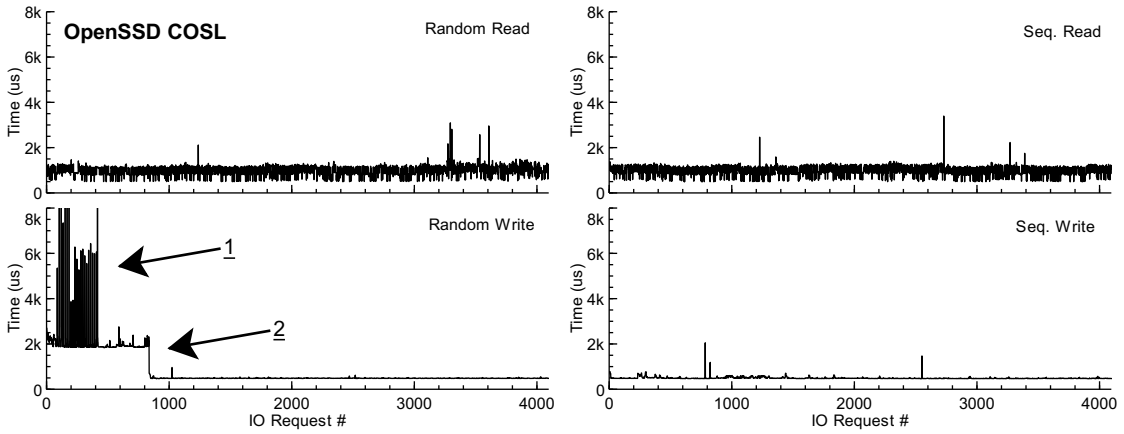


LightNVM Performance









LightNVM: Next Steps

- More research ...
  - New Targets (B Forest, ...)
  - New Open-channel SSDs (Cosmos, ...)
  - SSD Management variants (WOLF)
- Open-source contributions
  - Linux
    - Linux Plumber Workshop on Open-channel SSDs
    - Lower RAM occupation, better performance
  - Lightstor foundation
    - Joint work with GS Madhususan at IIT Chennai
    - Contribution to RapidIO
- http://github.com/ MatiasBjorling/LightNVM

B-Forest

- Tree Logarithmic method for database indexing
  - Writes grouped in chunks, updated at the same time
  - Multiway merges to leverage SSD parallelism
  - Bloom filters to speed up lookups

A variant of the block target in LightNVM

WOLF

- Garbage Collection
  - Flash blocks partitionned in groups based on update frequency
  - Fixed overprovisioning per group (and local GC per group) dominates global over-provisioning (and global GC)
  - WOLF block manager adapts to changing update frequencies
    - Aggressive GC for cooling groups to re-distribute over-provisioning across groups.

A variant of the LightNVM Adress Space Manager and page-based GC

Energy-Proportional Transaction Management

- SSDs are entering the microsecond era.
- SSDs and persistent memories require a profound redesign of system software.

- With Persistent Memories, persistence is no longer tied to secondary storage.
  - How can the OS handle Persistent Memories:

    (1) By extending virtual memory

    (2) By providing a block device interface for such memories

    (3) By designing new file systems tailored to their characteristics

Energy-Proportional Transaction Management

- Back to the good-old single-level store:
  - durability – how/when is data manipulated in the virtual address space made durable?
  - concurrency – how to design data structures that can perform efficiently when accessed concurrently
  - security – how to enforce access control but also integrity for a given portion of memory or storage?

- LightNVM common services extended to Persistent Memories
- Transactional VMM as LightNVM target

- Now with energy proportionality as a design goal.

Energy-Proportional Transaction Management

- We conjecture that durability can be achieved efficiently with **steal/force**
  - making redo log obsolete and ensuring that the synchronous random writes required by a force policy are competitive with the synchronous sequential writes followed by asynchronous random writes that characterize legacy write ahead logging protocols.
  - Virtualized storage that will be integrated in the single-level store is naturally organized as a journal, thus providing the capabilities of an undo log.