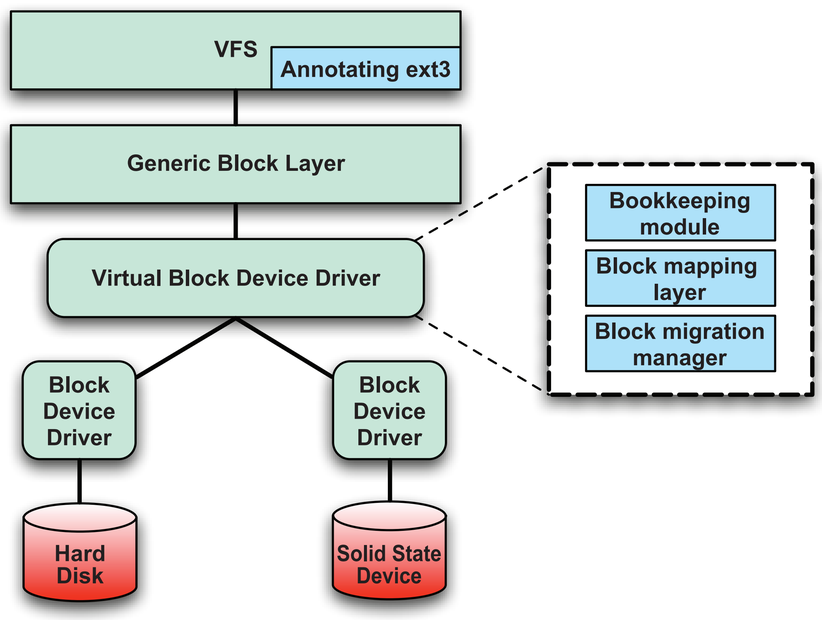
Efficient Data Placement in a Hybrid Storage Architecture

Jiri Simsa, Milo Polte, Garth Gibson

Overview

- Diminishing, but still high, cost of Solid State Devices motivates a hybrid cost-effective magnetic disk and high-performance SSD
- Goal is to optimize the average service time for throughput and latency for entire filesystem
- Disks and Solid State are based on technologies with different performance behaviors
  - Sequential I/O request performance is comparable
  - Random I/O request performance can vary
- To better understand SSD performance we examine the underlying technology both theoretically and practically
- At the same time we implement infrastructure for
  - data partitionning - based on block types
  - caching - based on recency and request size

Architecture



Status and Plans

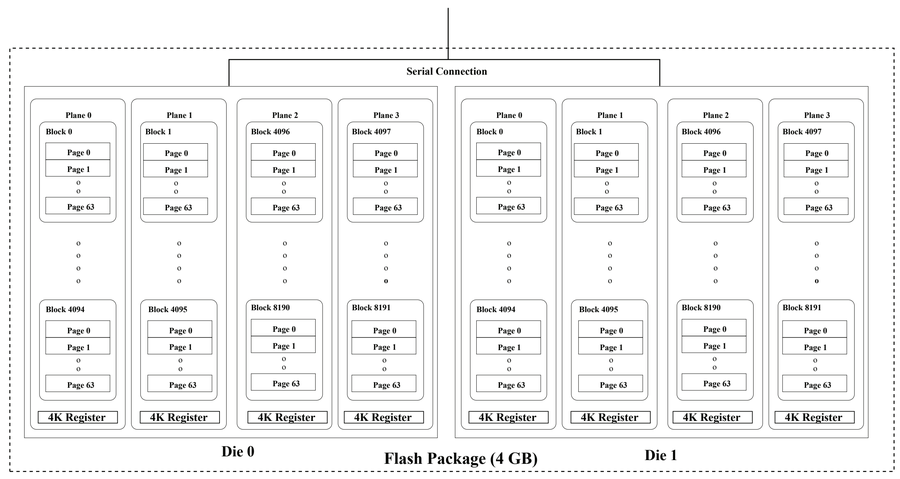
Status:

- Testing performance of various commercial SSDs
  - Some SSDs do random writes as poorly as disk
  - Where random writes are faster requires use of a volatile write buffer
- Coding framework for experimenting with assignment/caching policies in SSD

Plans:

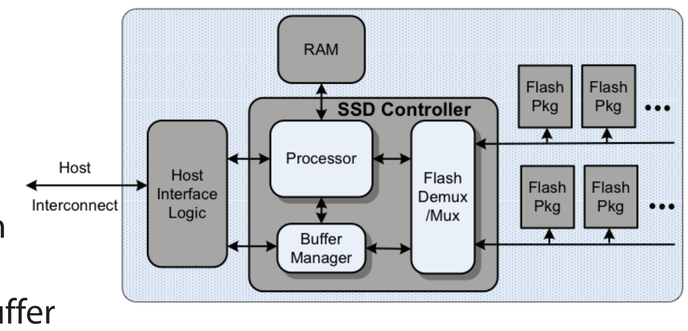
- Continued testing of newer SSDs (X25-E, FusionIO)
- Caching policy experiments with faster SSDs
  - Policies: metadata in SSD, all small random I/O in SSD
  - Apply file system type information if useful
    - Annotated ext2/ext3 so far

Insides of SSD



In order to write to a page, its block has to be erased.

Key attributes:
Write amplification
I/O parallelism
Volatility of SSD buffer

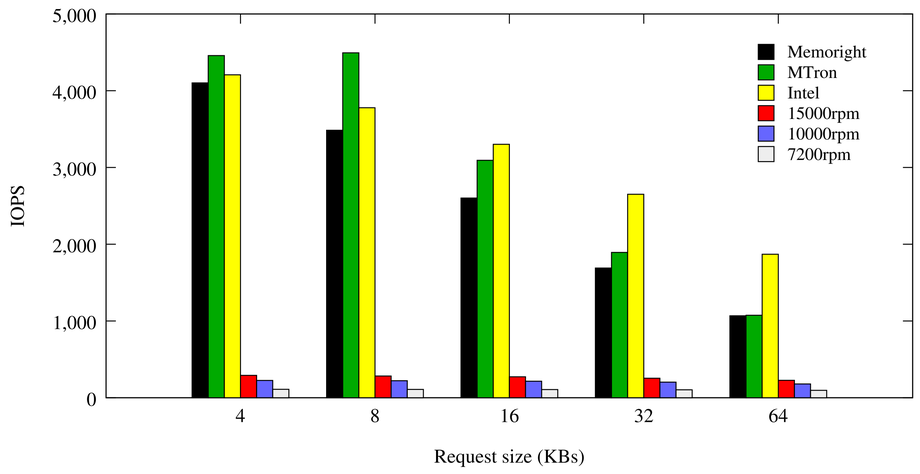


Performance Measurements

IOZone accessing 4GB file through ext3, with buffer cache enabled
Mtron 16GB @ $370, Memoright 16GB @ $510, Intel X25-M 80GB @ $730
Sequential speeds range from 80MB/s to 240MB/s

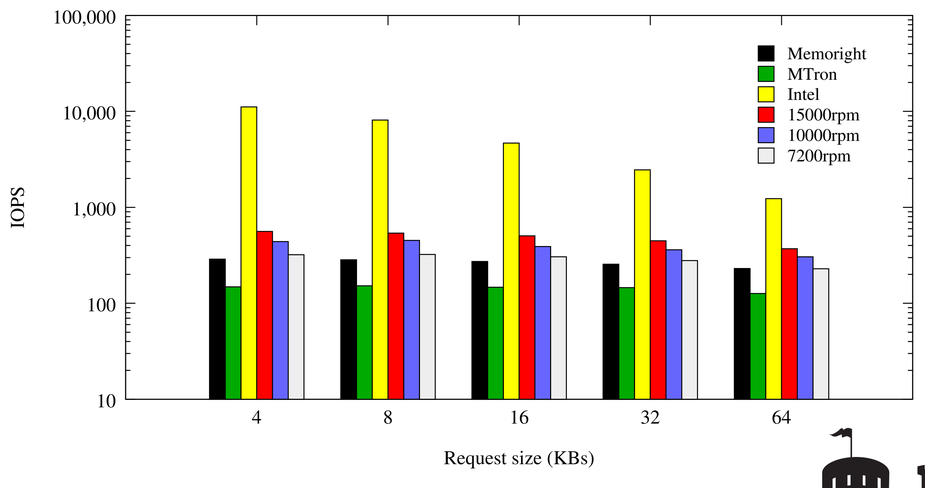
Random reads performance

All SSDs improve IOPS per $ by 5 - 10x versus disk



Random write performance

Intel X25-M SSD improve IOPS per $ by 5x versus disk
For this improvement, use of volatile buffer is essential



Carnegie Mellon

Parallel Data Laboratory