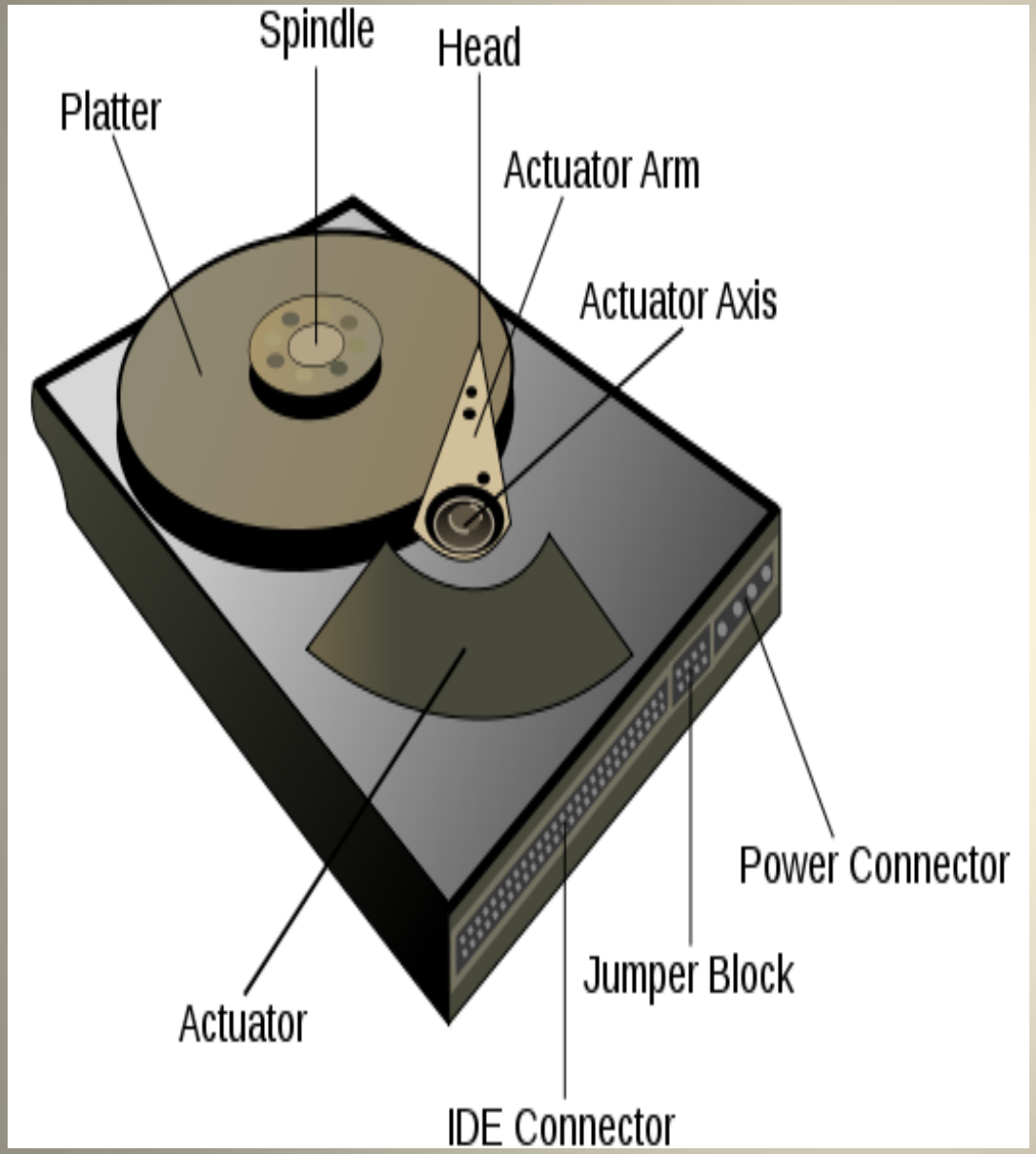
Design Tradeoffs for SSD Performance

Preview

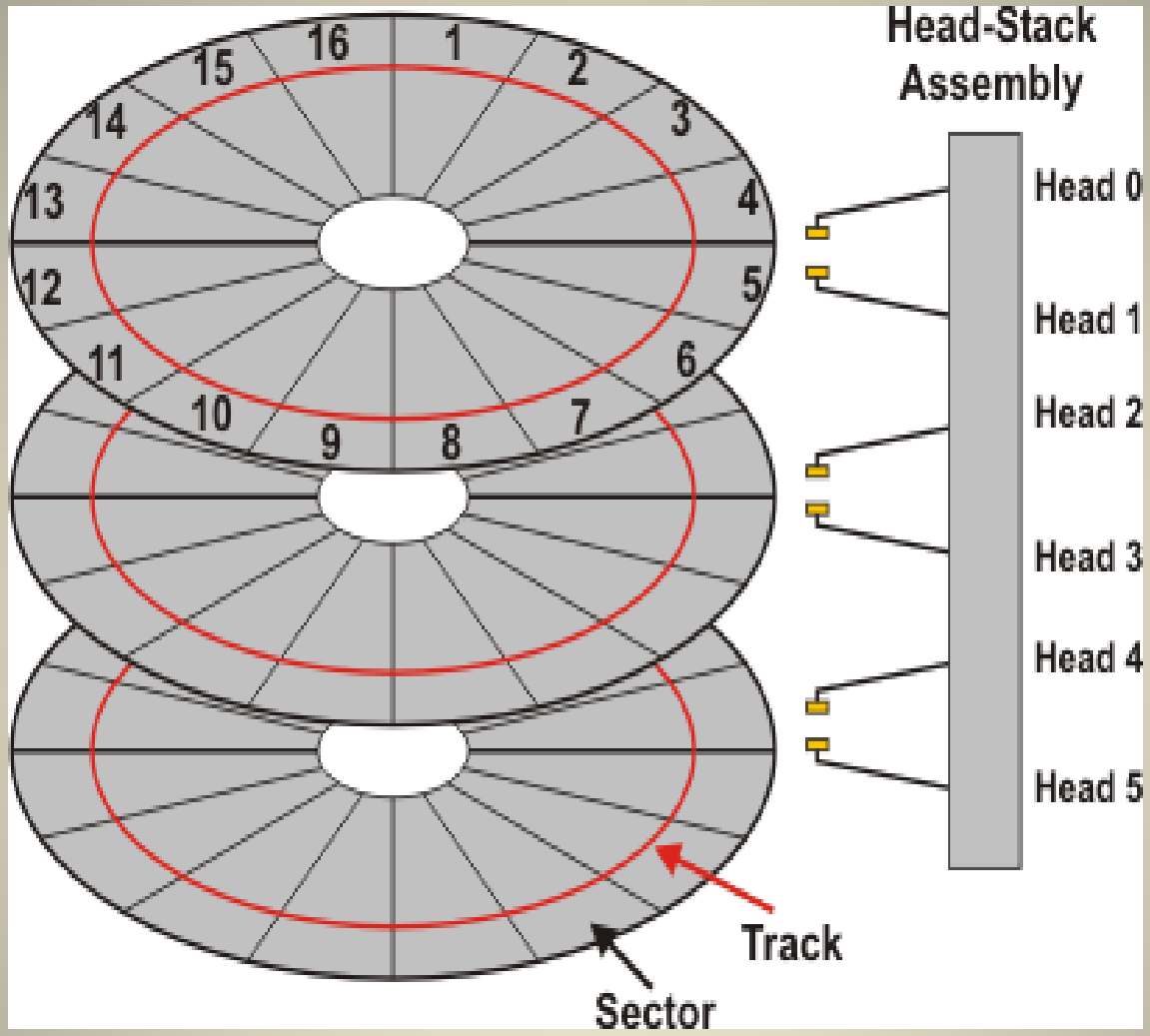
1. Creating an environment.
2. Background.
3. Basic functionalities and major challenges in implementation.
4. Simulation environment.
5. SSD wear leveling.
6. Related work.
7. Concludes.

Preview

1. Creating an environment.
2. Background.
3. Basic functionalities and major challenges in implementation.
4. Simulation environment.
5. SSD wear leveling.
6. Related work.
7. Concludes.



- Spindle holds platter
- Platters are made of non-magnetic material
- Covered with thin, shallow layer of magnetic material
- Conceptually divided into magnetic domains
- Read-write head magnetizes the region.


Head-Stack
Assembly
Head 0
Head 1
Head 2
Head 3
Head 4
Head 5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Track
Sector

Drawbacks of the Hard Drives

• Sign up time : Several seconds and not instantaneous.

•No random access.

•Mechanical reliability.

•Not shock resistive.

•Lower write speed than read.

•Power consumption for high performance HDDs require 12-18 watts.

•High capacity, hence high latency.

Solid State Drive(SSD)

- Why? :- Drawbacks of HDD
- Very high Bandwidth.
- Random I/O.
- Significant savings in power budget.
- Absence of moving parts improves system reliability.
- Very portable, Shock resistive, small in size.
- And many more.

Why still not in market?

- Cost/unit capacity is significantly high.

  e.g. Samsung 128 GB for $300.
- Intellectual property.
- Very little literature is available.

Our discussion

- SSDs available in market are NAND flash based.
- Where NAND flash based memories are used?

  camera, USB drives, IPods, etc.

Issues of SSD performance

- Data placement: Careful placement of data for load balancing and to effect wear leveling.
- Parallelism: Memory components must coordinate to operate in parallel.
- Write ordering: An important drawback of NAND flash.
- Workload management: Performance is highly workload dependant.

Preview

1. Creating an environment.
2. Background.
3. Basic functionalities and major challenges in implementation.
4. Simulation environment.
5. SSD wear leveling.
6. Related work.
7. Concludes.

Our Main Focus

- 4 GB Samsung's K9XXG08UXM series NAND flash part.
- Other Vendors.
- Specifications of single level cell flash (SLC).

Why?? 1.One bit per cell.

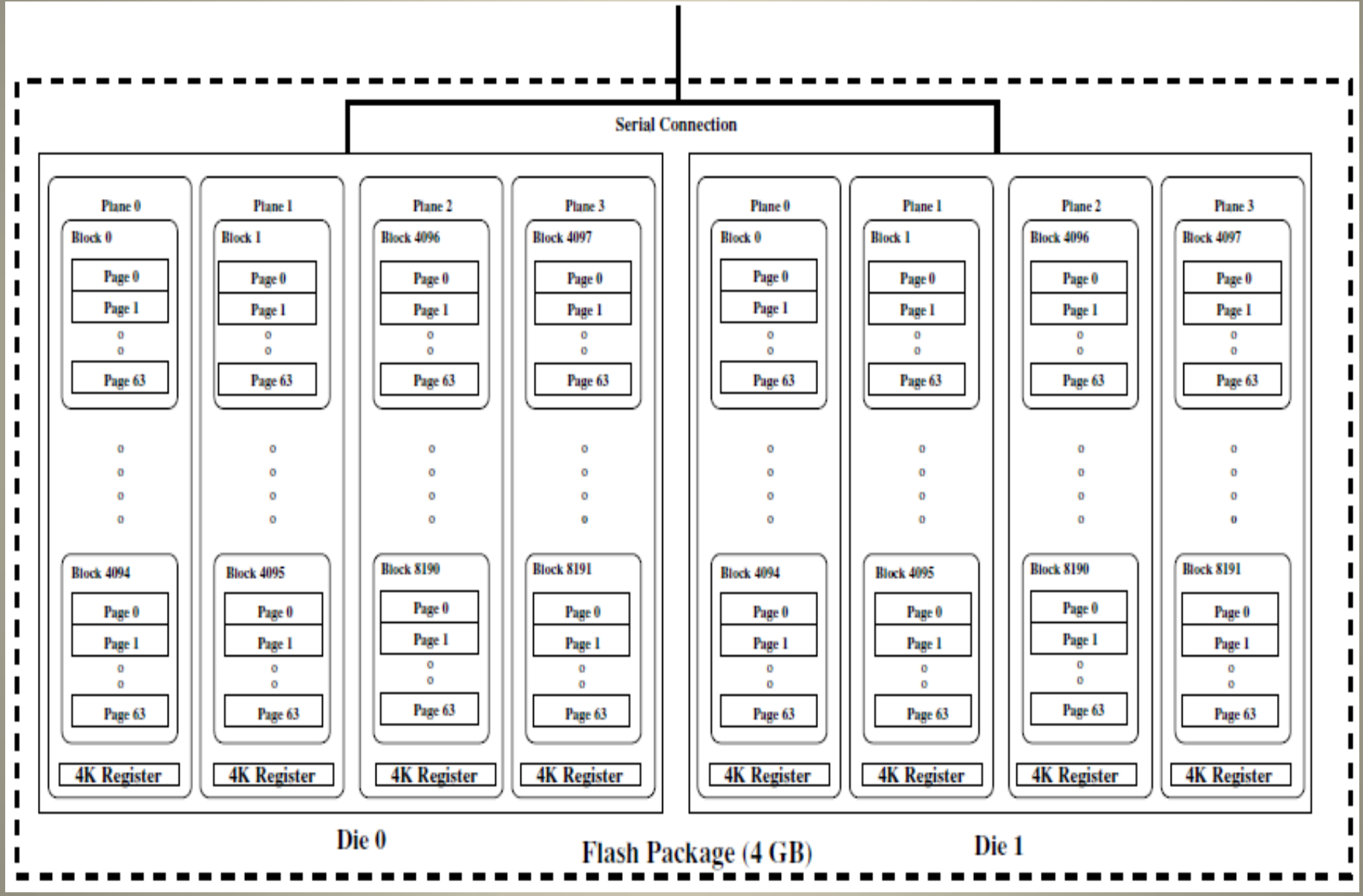
2.Costlier as compared to MLC.

3.Faster write speeds

4.Lower power consumption

5.Higher cell endurance

Samsung 4GB Flash Internals



Specifications

- Composed of one or more dies.(chips).
- Two 2GB dies.
- Sharing 8-bit serial I/O bus and common control signals.
- Separate chip enable and ready/busy signals.

  Hence, one can accept data and other can perform operations.
- Supports interleaved operations. (contin)

Specifications

- Each die contains 8192 blocks, organized among 4 planes.
- Each plane contains 2048 blocks.
- Each block contains 64 pages each of size 4KB.
- Each page has data and 128 byte region for metadata.
- Operation is possible among adjacent planes only. E.g. 0 & 1 and 2 & 3.

Properties of Flash Memory

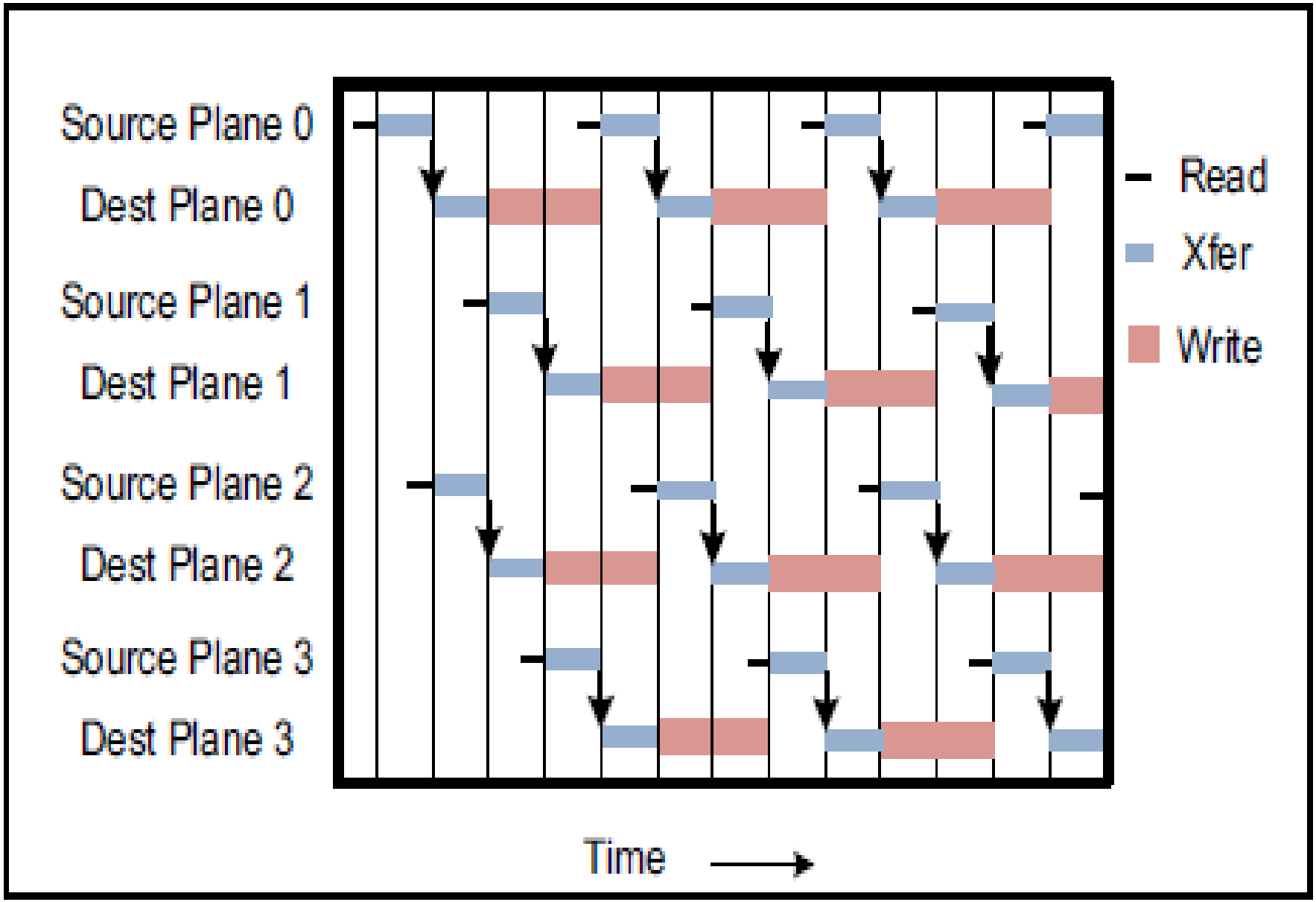
Page read to register **Serial access to register** **Write from register** **Block erase**	**25 µs** **100 µs** **200 µs** **1.5 ms**
Die size Block size Page size Data register Planes per die Dies per package Program/erase cycles	2 GB 256 KB 4KB 4KB 4 1, 2 or 4 100 K

Bandwidth and Interleaving

- Serial interface is a primary bottleneck for SSD performance.
- 25µs to move data into the register from NAND cell and 100µs to transfer 4KB page from on-chip register to off chip register.
- It makes bandwidth of 32MB/sec (8000 page reads/second). If interleaving is provided within the die then 40MB/sec (10000 page).
- For write, without interleaving 13MB/sec (3330 pages/ sec).

Constraints on Interleaving

- Operations on the same flash plane can’t be interleaved. E.g. copy-back operations. Data can be copied within the same flash plane without interleaving but two such copies can be interleaved among themselves.
- Same package interleaving is best employed for a choreographed set of related operations. E.g. multipage read or write.

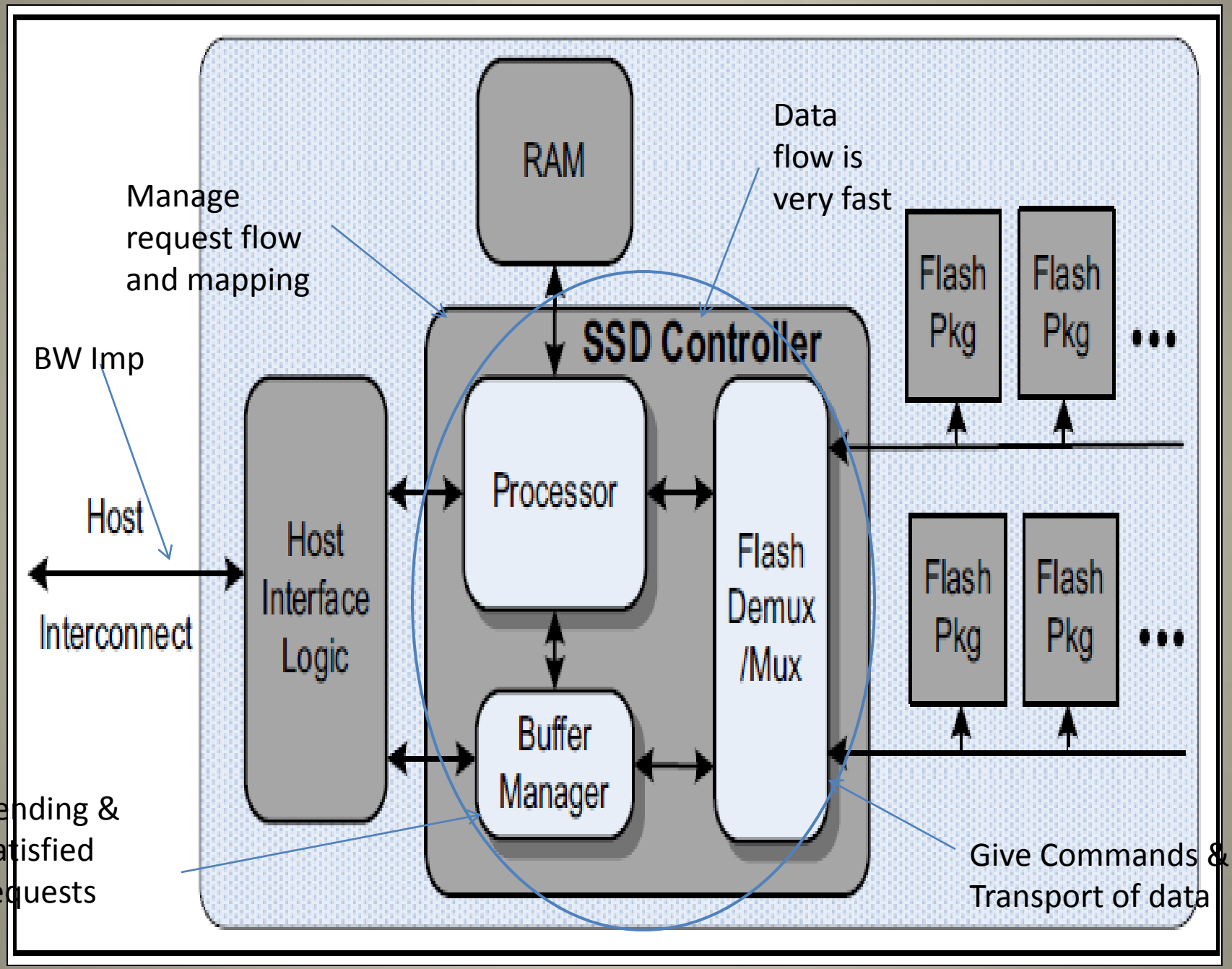

Source Plane 0
Dest Plane 0
Source Plane 1
Dest Plane 1
Source Plane 2
Dest Plane 2
Source Plane 3
Dest Plane 3
Read
Xfer
Write
Time

Preview

1. Creating an environment.
2. Background.
3. Basic functionalities and major challenges in implementation.
4. Simulation environment.
5. SSD wear leveling.
6. Related work.
7. Concludes.

SSD Basics

- Our focus: Organization of flash array and the algorithms needed to mapping between logical disk and physical flash address.


RAM
Data flow is very fast
Manage request flow and mapping
SSD Controller
BW Imp
Host
Interconnect
Host Interface Logic
Processor
Buffer Manager
Flash Demux /Mux
Flash Pkg
Flash Pkg
Flash Pkg
Flash Pkg
Pending & Satisfied requests
Give Commands & Transport of data

Logical Block Map

- Logical block address is used for specifying location of blocks of data stored on computer storage device.
- SSD must maintain mapping between LBA & physical flash location.
- LBM is held in volatile memory and reconstructed from stable storage at start time.

Concept of allocation pool

- Pre-allocating a number of memory blocks with the same size called the memory pool.
- Handling a write request, each target logical page is allocated from pre-determined pool.
- Scope of allocation pool: small as flash plane or large as multiple flash packages.

Variables and Constraints

Variables:

- Static map: fixed mapping to allocation pool.
- Dynamic map: lookup key.
- Logical page size: very
- Page span: accessing sections in parallel.

Constraints:

- Load balancing:
- Parallel access:
- Block erasure:

Avoid Following

- Large space is statically mapped: no load balancing.
- Many LBAs mapped to same physical die: no sequential access.
- Small logical page size: affects erasure operation.

Cleaning

New page write → previously mapped page location is superseded as data becomes out of date → the pages which are not superseded in candidate block must be written elsewhere.

Cleaning efficiency : $\frac{\text{Superseded pages}}{\text{Total pages in block cleaning}}$

Many algorithms. Optimize cleaning efficiency.

- NAND flash has limited number of erasures/block. Hence blocks need to be chosen properly (evenly growth).
- Hence for safer side, SSDs are over provisioned with spare blocks to reduce the demand for cleaning blocks in foreground.
- If active block and cleaning state/plane is maintained, then cleaning operation can be arranged with high probability.

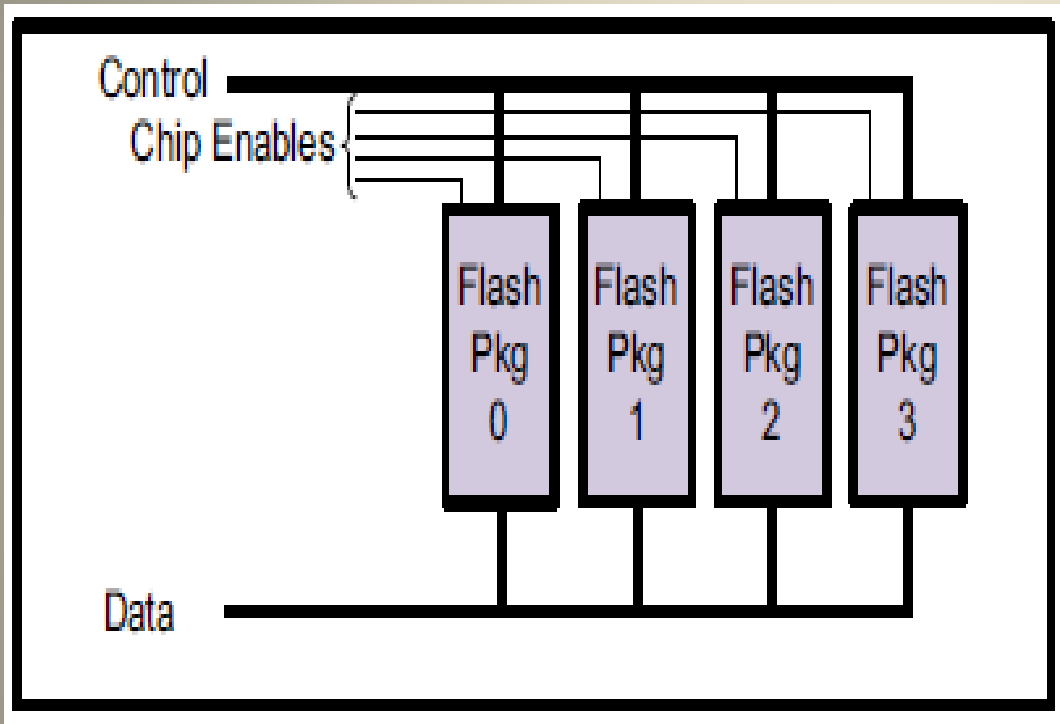
Parallelism and Interconnect Density

- For high BW, handle I/O requests on multiple flash packages in parallel.
- Techniques:

  1. Parallel requests: Each entity can accept separate flow of requests. Drawback of maintaining multiple queues.

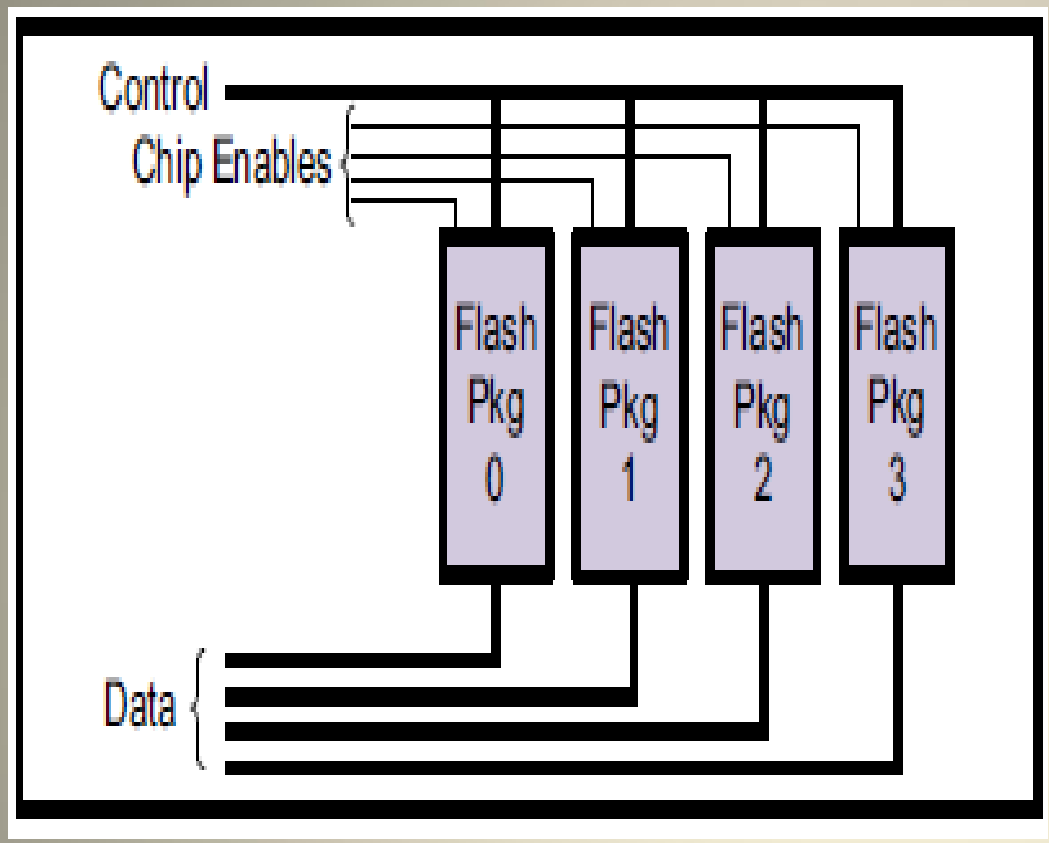
  2. Ganging: Gang of flash packages are synchronized. No multiple queues. But rest of the elements will seat idle.

Two choices:



Shared Bus Gang

- Data and control line are shared.
- Controller selects the target for each command.
- Less pins.
- BW is not very large.



- Each package has separate data path to controller.
- Shared control pins.
- Many number of pins.
- Provides high BW.

Shared Control Gang

3. Interleaving: Increase BW.

4. Background cleaning: Idle components are cleaned in background.

The best choice is will be dictated by workload properties.

Persistence

- For recovery: Building LBM, Data Structures.
- Each flash page contains dedicated area for metadata storage, which stores LBA.
- Logical block maps can be hold in phase change RAM or magnetoresistive RAM. But both are very costly.

Industry Trends

NAND flash are broadly divided into three categories:

1. Consumer Portable Storage: USB flash sticks, camera memories.

   Very poor write.
2. Laptop Disk Replacement:
3. Enterprise/database accelerators: Very strong random read, write and sequential performance.

	Sequential		Random 4K	
	Read	Write	Read	Write
USB	11.7 MB/sec	4.3 MB/sec	150/sec	<20/sec
MTron	100 MB/sec	80 MB/sec	11K/sec	130/sec
Zeus	200 MB/sec	100 MB/sec	52K/sec	11K/sec
FusionIO	700 MB/sec	600 MB/sec	87K/sec	Not avail

1

2

3

3

Preview

1. Creating an environment
2. Background.
3. Basic functionalities and major challenges in implementation.
4. Simulation environment.
5. SSD wear leveling.
6. Related work.
7. Concludes.

Thank You.

Preview

1. Creating an environment.
2. Background.
3. Basic functionalities and major challenges in implementation.
4. Simulation environment.
5. SSD wear leveling.
6. Related work.
7. Concludes.

Simulator

- Modified version of DiskSim simulator from the CMU parallel data lab.
- Reason: infrastructure for processing trace logs and its extensibility made it a good choice for customization.
- Implemented an SSD module derived from the generic rotating disk module.

What things added?

- An auxiliary level of parallel elements, each with a closed queue – for supporting multiple request queue.
- Added logic for serialization.
- Data structures for representing SSD logical block maps, cleaning state, and wear-leveling state.
- Delay introduced as per the table.
- Supports features such as background cleaning, gang-size, gang organization, interleaving.

Workloads

- Trace: It is simply a logging of a set of data regarding the performance of storage device focusing on I/O requests.
- Workload traces: TPC-C, Exchange, IOzone and Postmark.
- First, examined synthetic workload to characterize baseline behavior.

(continue)

- IOzone and Postmark: std. file system benchmark. Can be simulated on a single SSD.
- TPC-C: instance of the well established benchmark. Trace for this is 30 min trace.

  warehouses: 16000, RAID controllers: 14, each supporting 28 high speed 36 GB disks. Workload contains twice as many reads and writes (8 KB).

  Alignment is important.
- Exchange: server running Microsoft Exchange. Workload with 3:2 read- to- write ratio.

Simulation Results

- Baseline configuration: SSD with 32GB of flash(4GB * 8). Allocation pools size = flash package. Logical page size and strip size = 4 KB.
- Cleaning is invoked when less than 5% free blocks remain.
- TPC-C requires 6 while Exchange requires 10 attached SSD to above.

Microbenchmarks

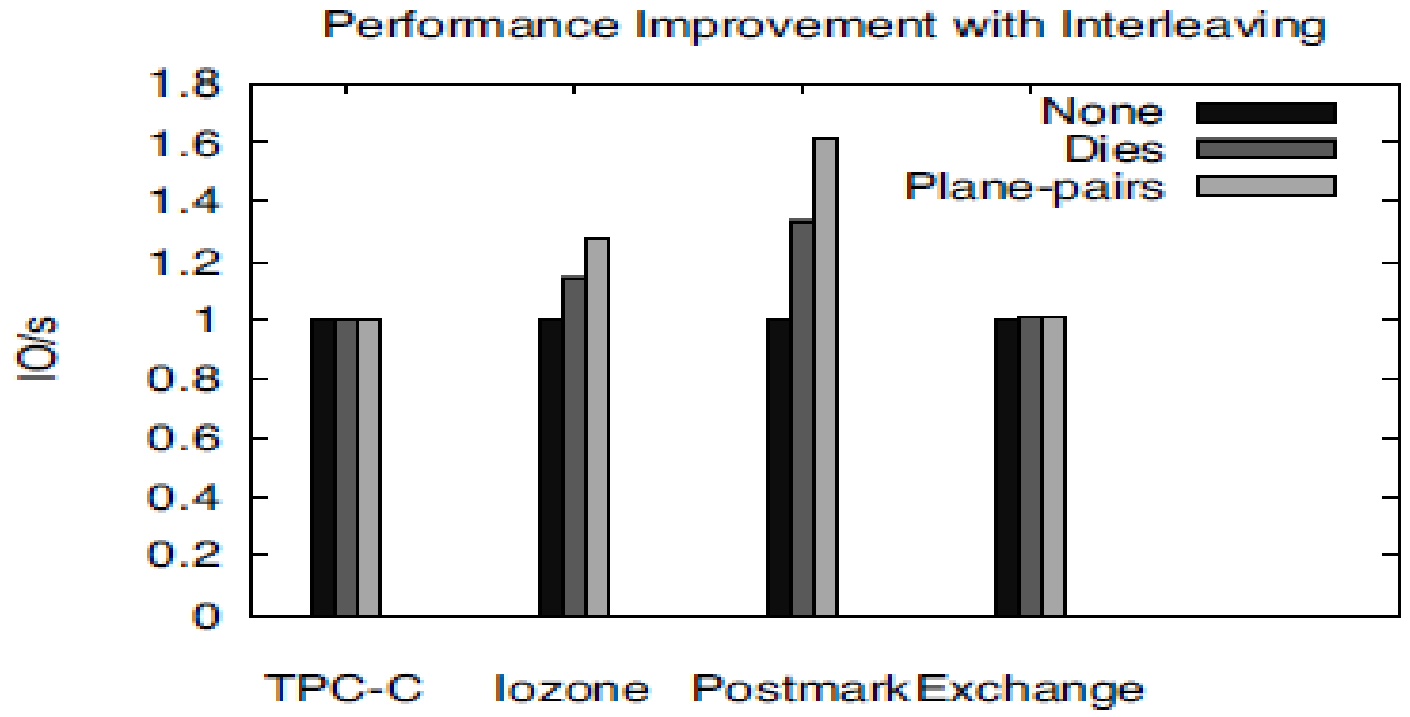
Microbenchmark	Cleaning	Latency (μs)	IO/s
Sequential read	x	130	61,255
Random read	x	130	61,255
Sequential write	x	309	25,898
Random write	x	309	25,898
Sequential write	$\surd$	327	24,457
Random write	$\surd$	433	18,480

Definition: A benchmark designed to measure the performance of a very small and specific piece of code.

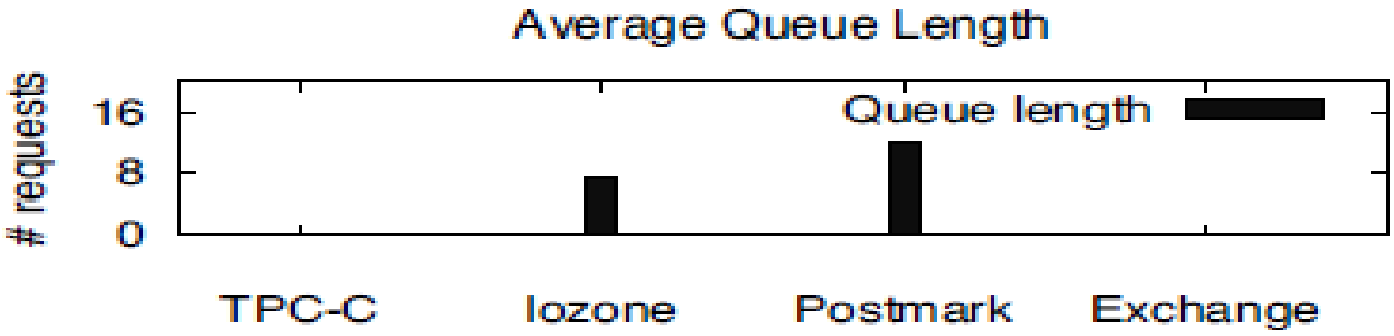
Page Size and Interleaving

- Choice of logical size has substantial effect.
- Example: TPC-C produces an average I/O latency of over 20 ms, when the page size is full block (256KB) and produces an average latency of 200 μs with a page size of 4 KB.
- Different types of interleaving has different performance effects.

(contin)


Performance Improvement with Interleaving
IO/s
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
None
Dies
Plane-pairs
TPC-C
Iozone
Postmark
Exchange

(a)


Average Queue Length
requests
16
8
0
Queue length
TPC-C
Iozone
Postmark
Exchange

(b)

Gang Performance

- Shared Gang Performance:

	No gang	8 - GANG	16 - GANG
HOST IO LATENCY	237 µs	533 µs	746 µs
IOPS PER GANG	4425	1087	1340

- Shared control gang:

It can organized in two ways:

1) Separate allocation and cleaning decisions on each package for opportunistic parallel operation. This is referred as asynchronous shared control ganging.
2) All packages in a gang in synchrony by utilizing logical page depth equal to gang size. E.g.: 8 wide, page size 32 KB.

Synchronous ganging uniformly underperforms when compared to asynchronous ganging due to page size.

Copy back VS Inter Plane Trnasfer

	No of blocks cleaned / flash	Avg. time (ms)	Efficiency
TPC - C (inter plane)	114	9.65	70%
TPC – C (copy back)	108	5.85	70%
IOzone	101170	1.5	100%
Postmark	2693	1.5	100%