Hystor: Making the best use of solid state drives in high performance storage systems

Authors: Feng Chen
David Koufaty
Xiaodong Zhang

Overview

- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.

Overview

- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.

Following are the important issues for fully exploiting the SSD performance:

- Effectively identifying the most performance critical blocks.

  : Performance gains highly dependent on workload access patterns.

  : Hence identifying the blocks which are going to be accessed is essential.

- Efficiently maintaining data access history with low overhead for accurately characterizing access patterns.

(continue)

- Avoiding major kernel changes in existing systems while effectively implementing the hybrid storage management policies.

Overview

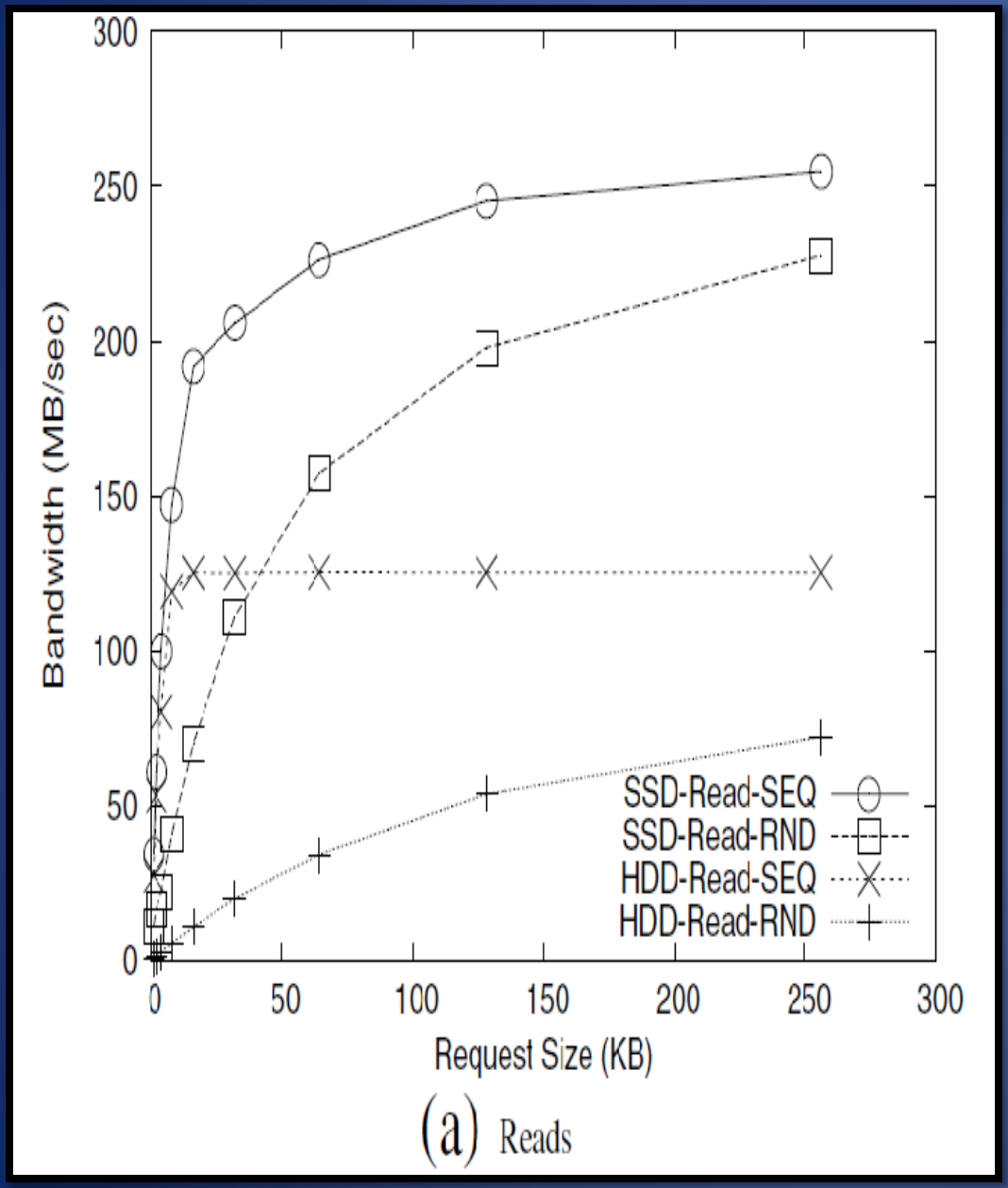
- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.

Performance advantages of SSD

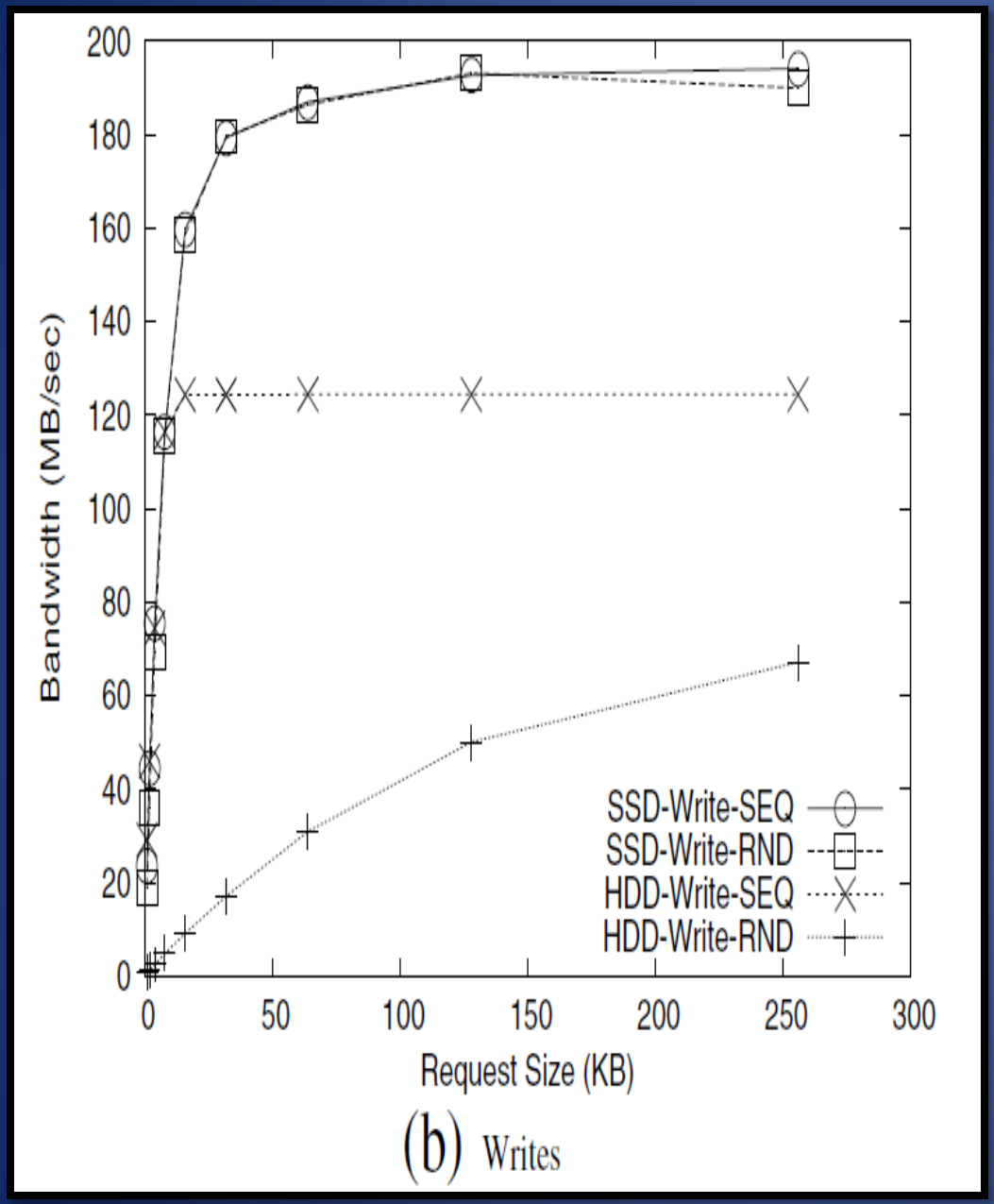
- To understand the relative performance strengths , four typical workloads, namely random read/write, sequential read/write are generated using Intel open storage toolkit.
- Storage devices:

  Intel X25-E 32GB SSD

  15,000 RPM Seagate Cheetah 15.5k SAS HDD



- Most significant performance gain in random read on the SSD.
- For 4KB random reads: 7.7 times more
- For 256KB sequential reads 2 times.



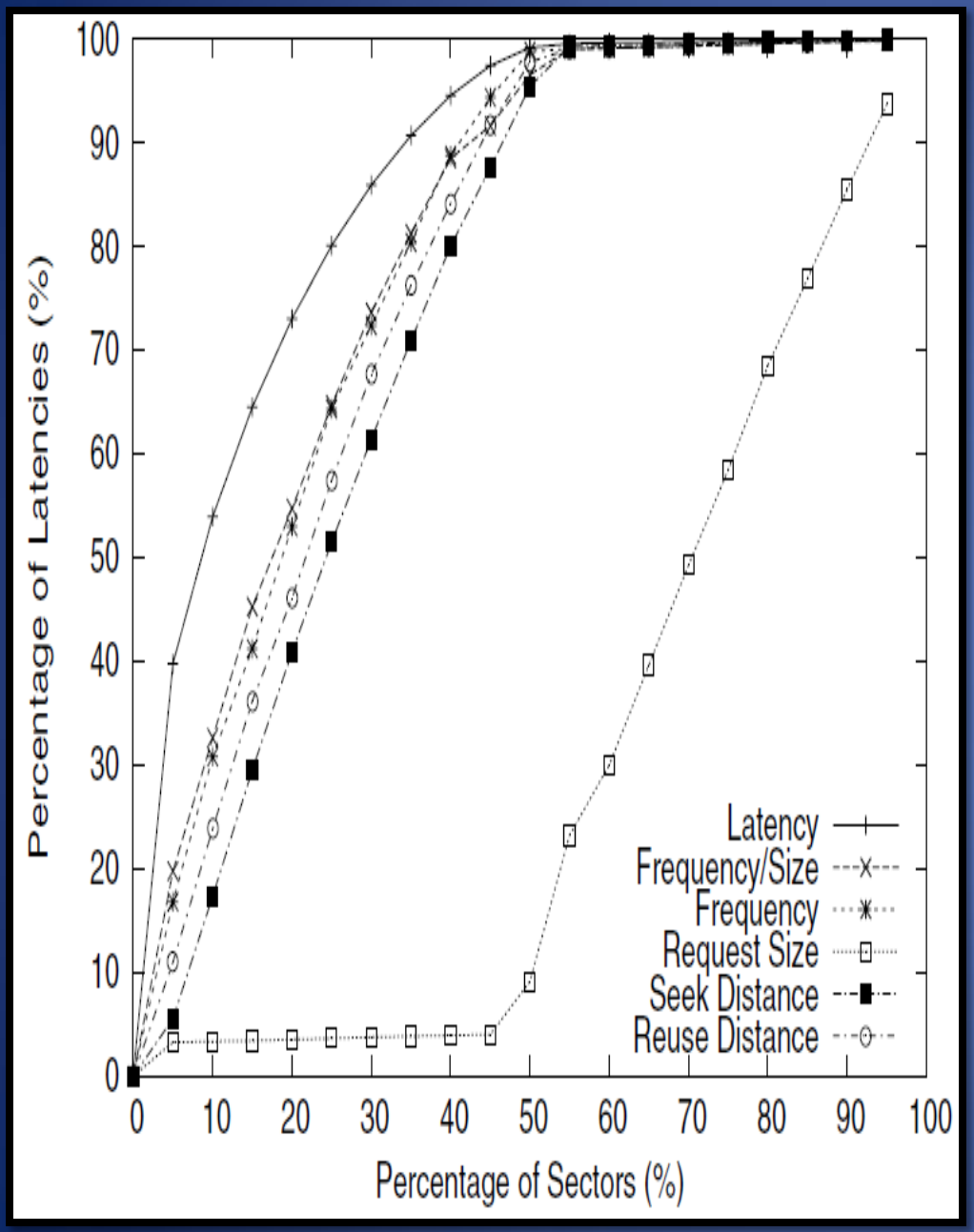
- Most significant performance gain in random writes on the SSD.
- For 4KB random writes: 28.5 times more
- For 256KB sequential writes 1.5 times.

Conclusions:

- Achievable performance benefits are highly dependent on workload access patterns, and we must identify the blocks that can bring the most performance benefits by migrating them into SSDs.
- Random writes can achieve almost identical performance as sequential writes.

Overview

- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.



- Associate each block with a selected metric and update the metric value by observing accesses to the block.
- Frequency/request size graph is the closest to the latency curve and hence will be the most effective one.

- Representing indicator metric:

$$b = 2^{max(0,7-\lfloor log_2 N \rfloor)}$$

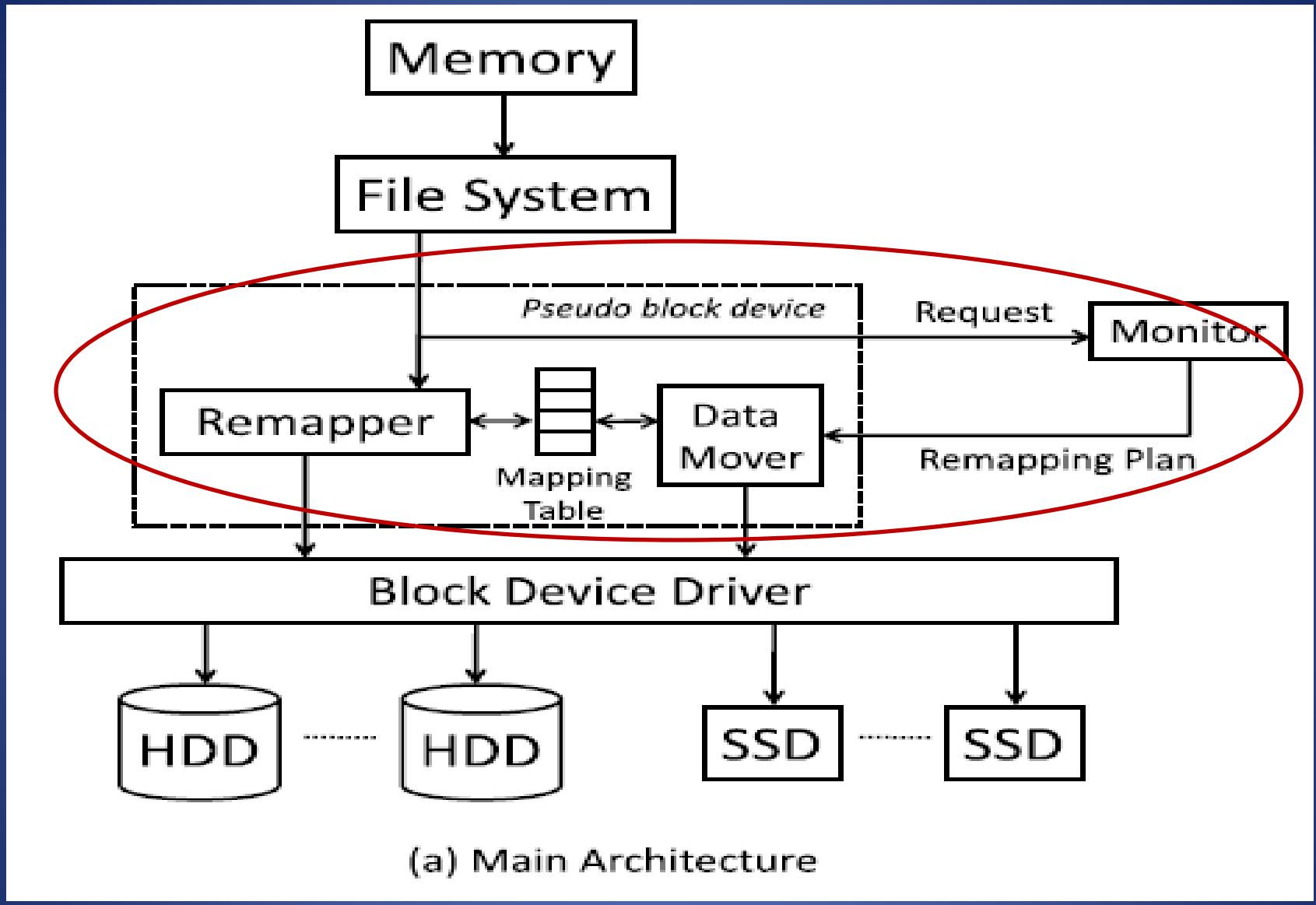
The technique of encoding request size and frequency is called inverse bitmap.

When a block is accessed by a request of N sectors, an inverse bitmap is calculated using above formula.

It encodes request size into a single byte. The smaller the request is, the bigger the inverse bitmap is.

Overview

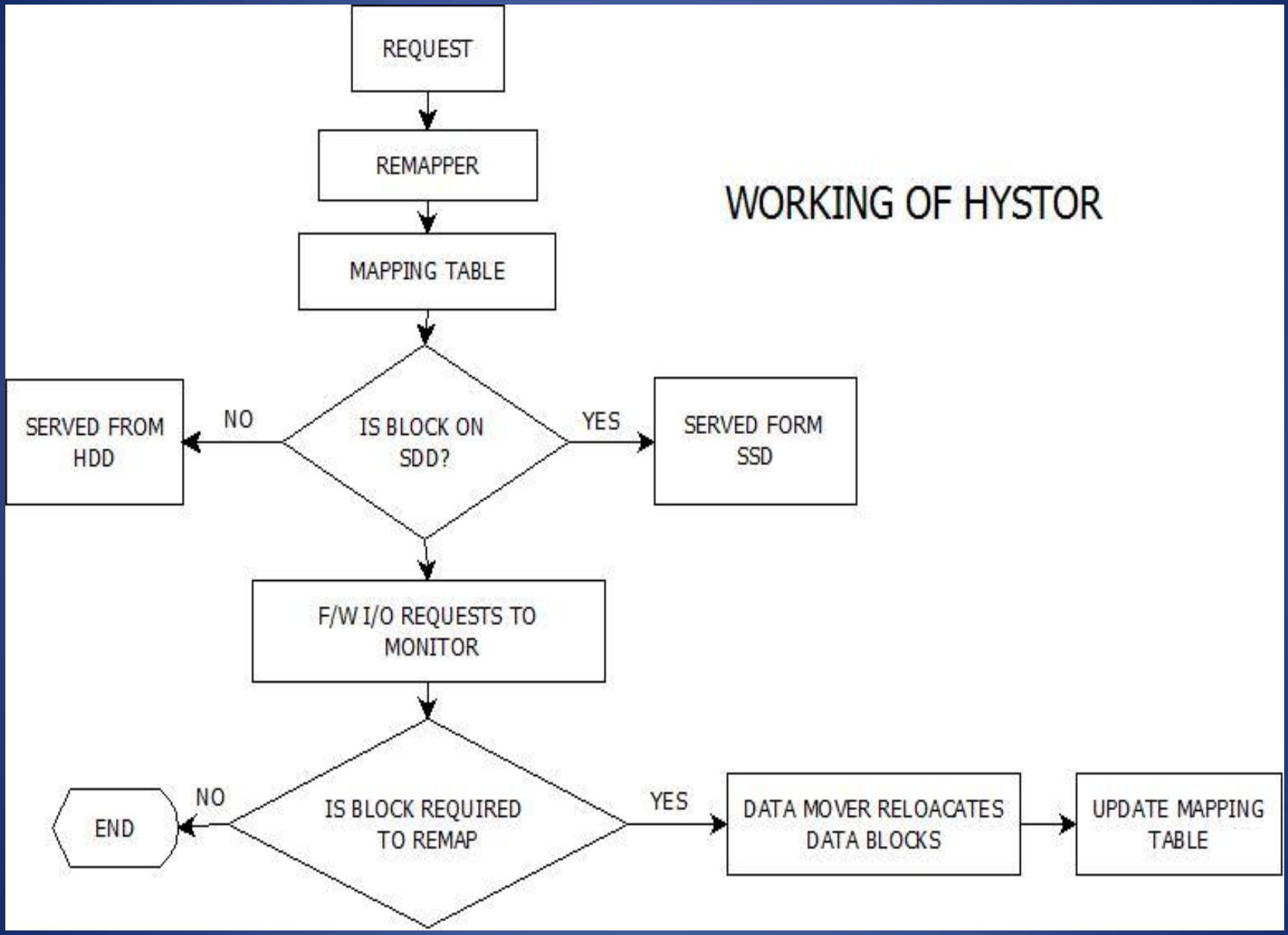
- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.



Architecture of Hystor

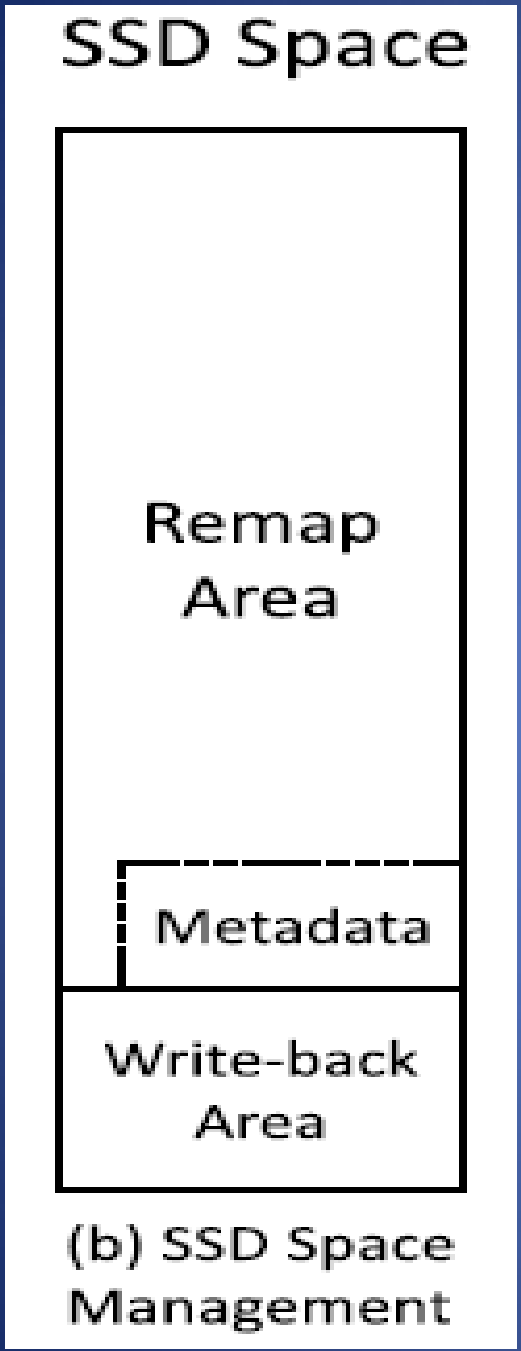
Hystor has three major components: remapper, monitor, data mover.

- Remapper: maintains mapping table to track the original location of blocks on the SSD.
- Monitor: Collects I/O requests and updates the block table to profile workload patterns. It analyzes the data access history, identifies the blocks which require remapping and requests data mover.
- Data Mover: relocate data blocks across storage devices. Update the data table.


WORKING OF HYSTOR
REQUEST
REMAPPER
MAPPING TABLE
IS BLOCK ON SDD?
NO
SERVED FROM HDD
YES
SERVED FORM SSD
F/W I/O REQUESTS TO MONITOR
IS BLOCK REQUIRED TO REMAP
NO
END
YES
DATA MOVER RELOACATES DATA BLOCKS
UPDATE MAPPING TABLE

Logical block Mapping:

- Each logical block is directly mapped to a physical block in the HDD and indexed using logical block number(LBN).
- A logical block is selected to remap to SSD and its physical location is chosen dynamically.
- A mapping table is maintained to keep track of remapped logical blocks only and hence the spatial overhead of it is small and proportional to the SSD size.



- Major role: Storage

  Minor role: write back buffer.
- Two types of blocks are remapped:

  1)high cost data blocks, which are identified by analyzing data access history.

  2)file system metadata blocks.

Managing the write-back area:

- The blocks in the write-back area are managed in two lists: clean list and dirty list.
- Whenever a write request comes, a block from clean list is allocated.
- If the number of dirty blocks in dirty list reaches a certain predefined level, a scrubber is awaken and all these blocks are written to HDD.
- SSDs used in this system are high end SSDs. There MTBF for them is 2 million hours and hence they wont wear out with few number of erase/cycles.

Overview

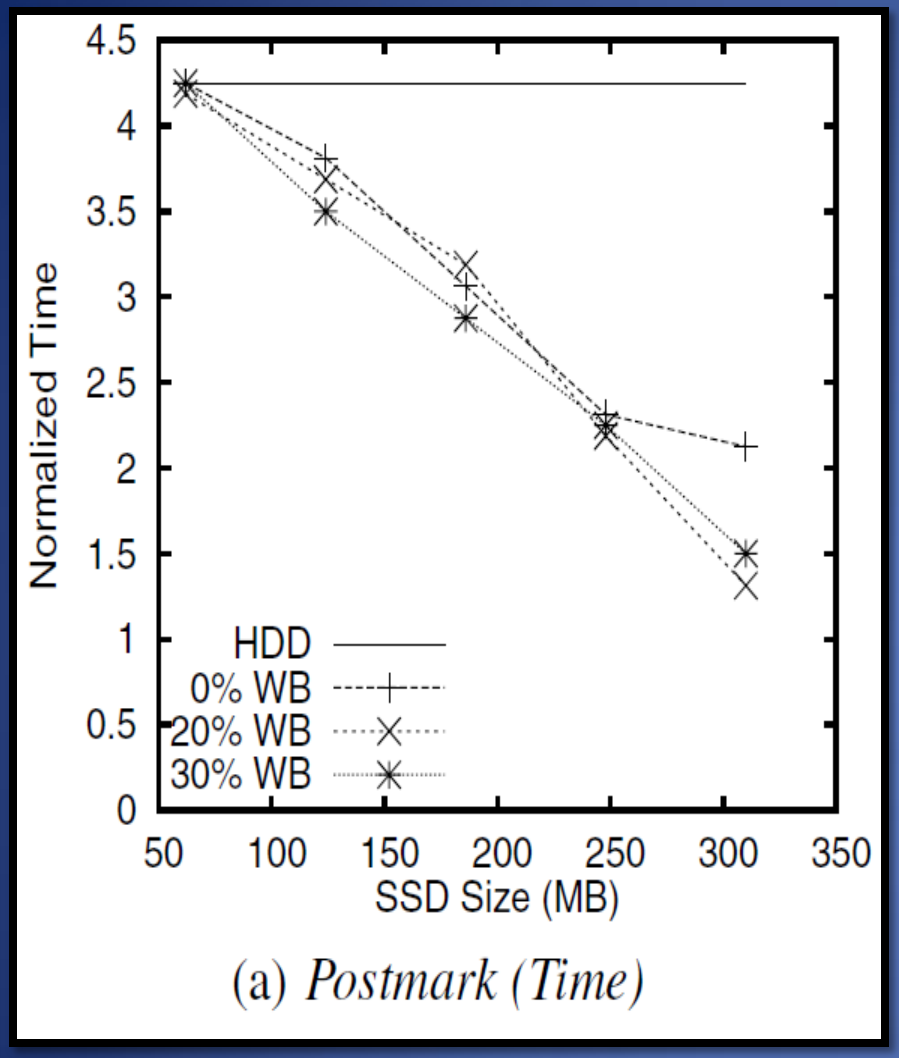
- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.

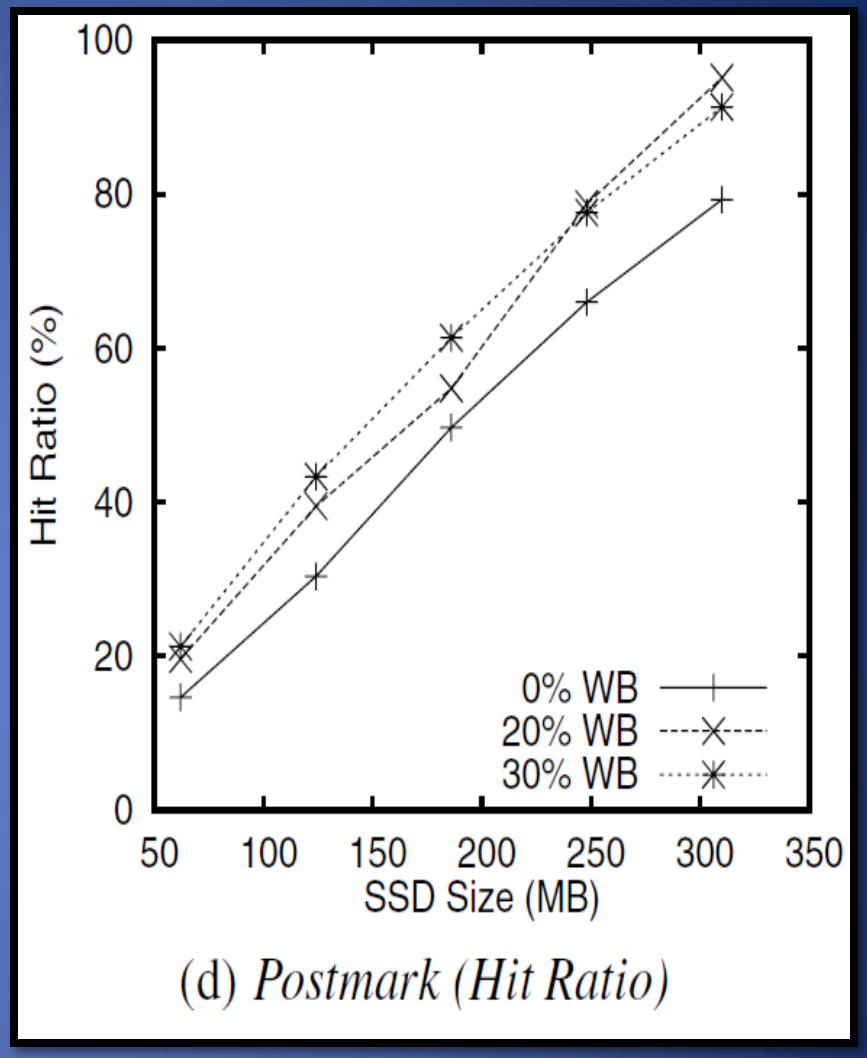
- Prototyped Hystor in the Linux kernel 2.6.25.8 as a stand alone kernel module with 2500 loc. The user level Monitor is implemented with 2400 loc.
- Experimental setup:

	X25-E SSD	CHEETAH HDD
CAPACITY	32GB	73GB
INTERFACE	SATA2	SAS
READ BANDWIDTH	250 MBPS	125 MBPS
WRITE BANDWIDTH	180 MBPS	125MBPS

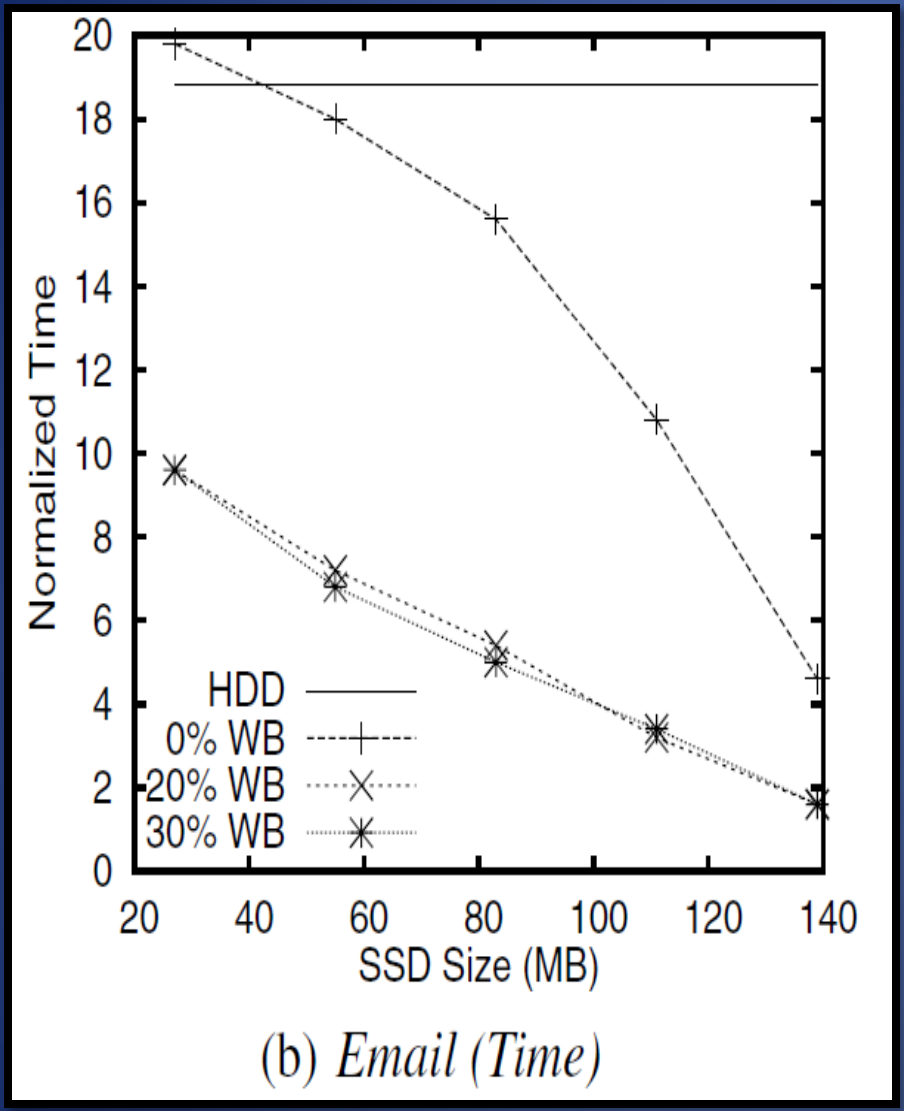
Workloads and Important Terms

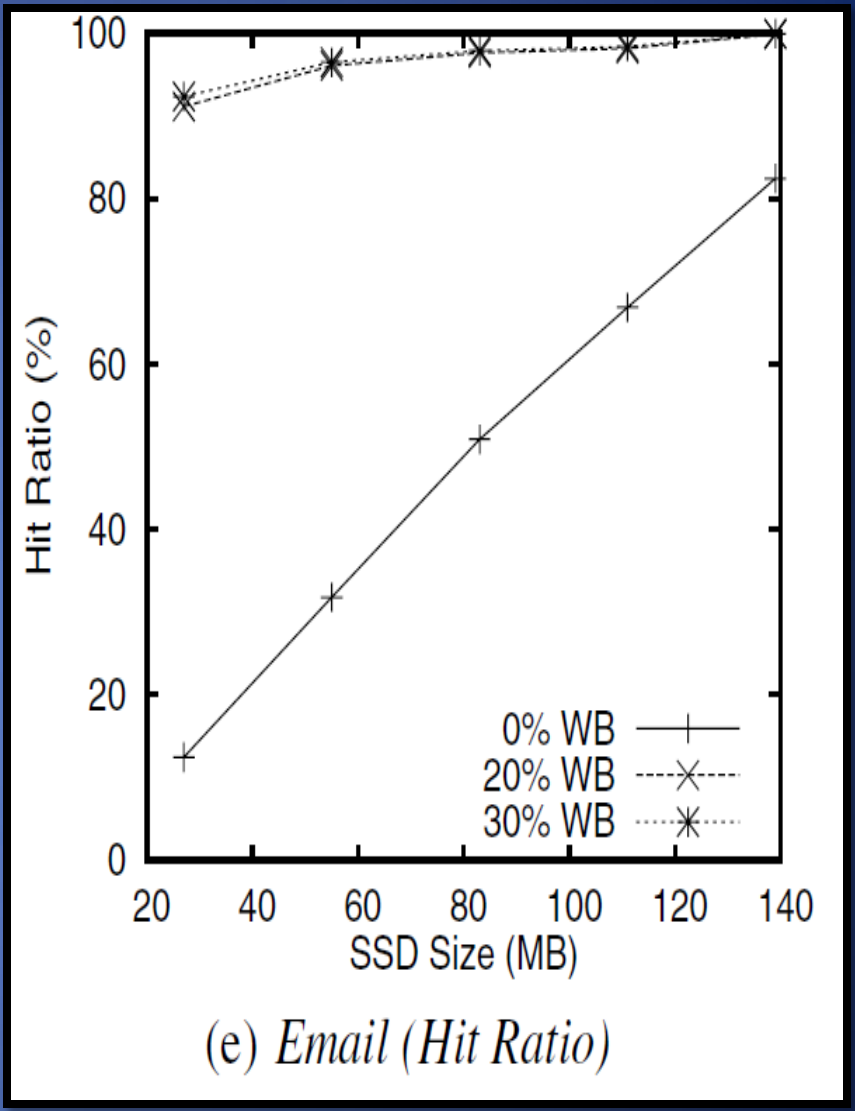
- Three Workloads: Postmark, Email, TPC-H Q1
- Execution times on the Y axis are normalized to that of running on the SSD-only system.
- For comparison purpose, a horizontal line is plotted on the graphs which indicates running on the HDD only system.
- A request to blocks resident in the SSD is considered a hit, otherwise a miss. Hit ratio describes what % of request are served from SSD.


Normalized Time
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
50
100
150
200
250
300
350
SSD Size (MB)
HDD
0% WB
20% WB
30% WB
(a) Postmark (Time)

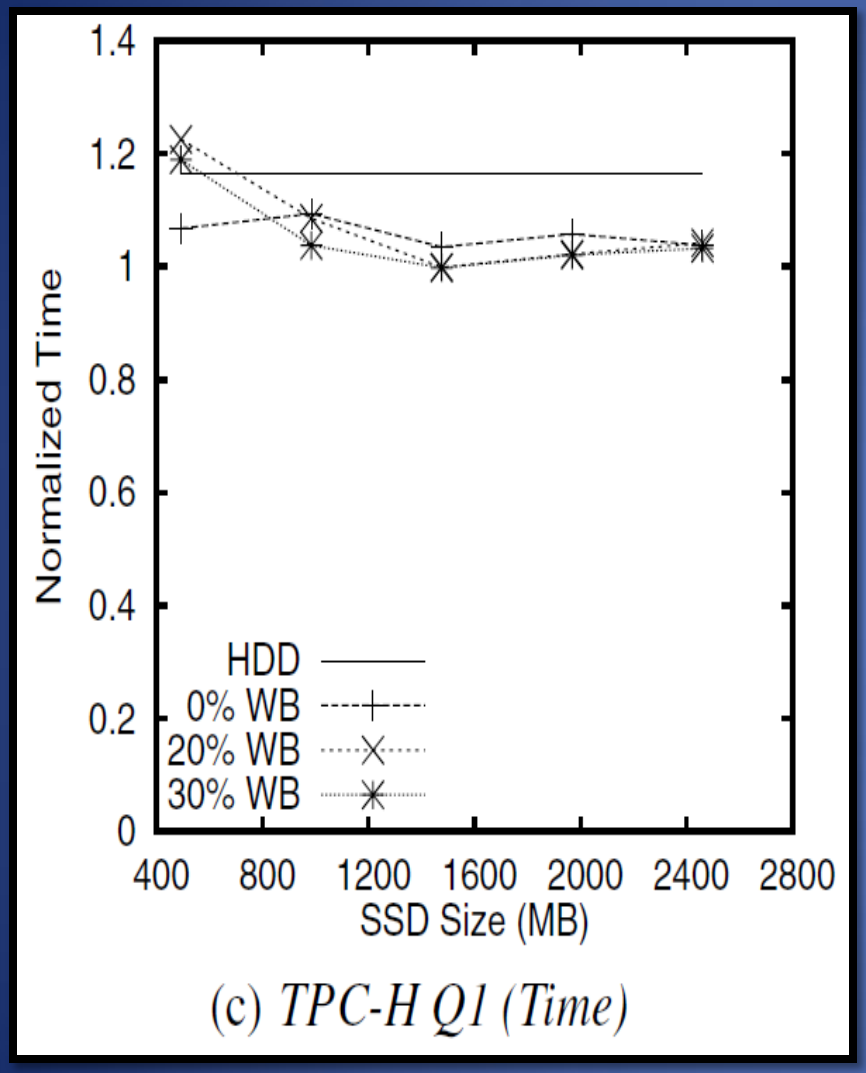

Hit Ratio (%)
100
80
60
40
20
0
50
100
150
200
250
300
350
SSD Size (MB)
0% WB
20% WB
30% WB
(d) Postmark (Hit Ratio)

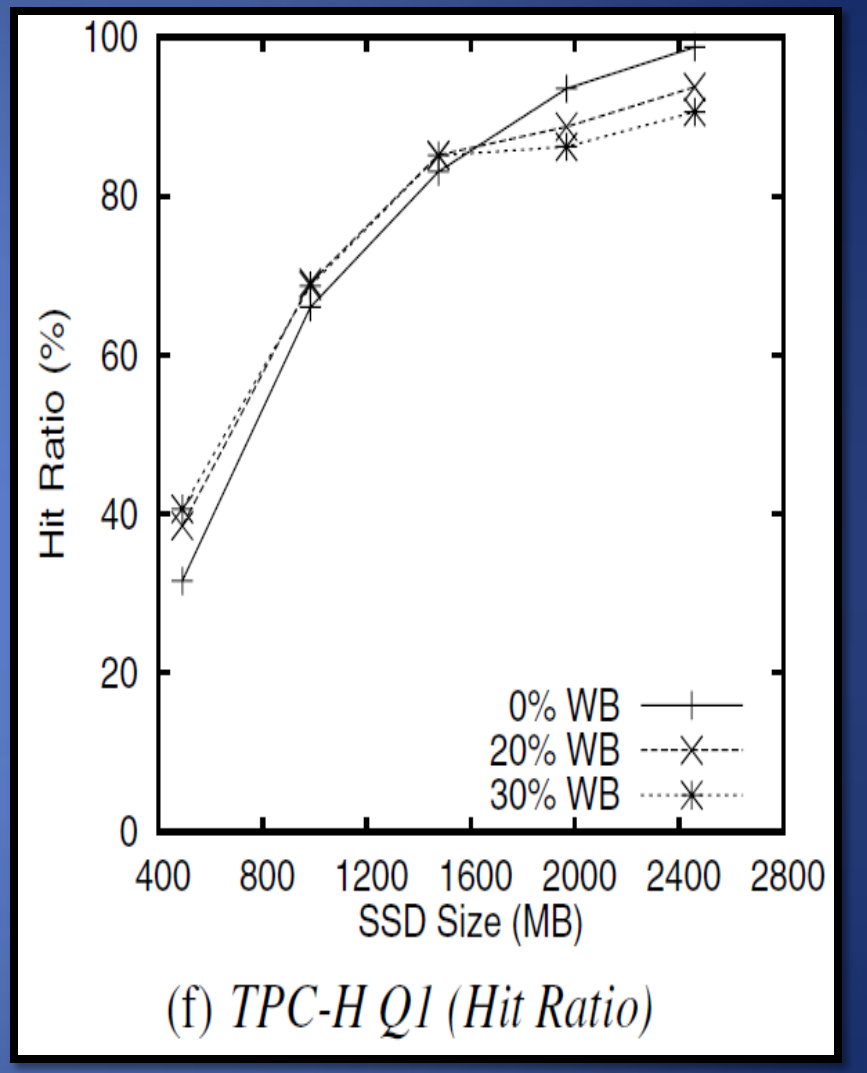
POSTMARK


Normalized Time
SSD Size (MB)
HDD
0% WB
20% WB
30% WB
(b) Email (Time)


Hit Ratio (%)
SSD Size (MB)
0% WB
20% WB
30% WB
(e) Email (Hit Ratio)

Email


Normalized Time
1.4
1.2
1
0.8
0.6
0.4
0.2
0
400
800
1200
1600
2000
2400
2800
SSD Size (MB)
HDD
0% WB
20% WB
30% WB
(c) TPC-H Q1 (Time)


Hit Ratio (%)
100
80
60
40
20
0
400
800
1200
1600
2000
2400
2800
SSD Size (MB)
0% WB
20% WB
30% WB
(f) TPC-H Q1 (Hit Ratio)

TPC – H Q1

Overview

- Introduction.
- Performance advantages of SSD.
- Deciding a metric and encoding it.
- The design of Hystor.
- Evaluation.
- Conclusion.

Conclusion

Complete replacement of HDD by SSD is not beneficial. Hence we need to find the fittest position of SSDs in the existing systems to strike a right balance between performance and cost. In this study, a simple yet effective metric is used to find the best suitable data that can be held in SSD. Use of a SSD as a write back buffer was also effective.

Thank You…