BROMS: Best Ratio of MLC to SLC

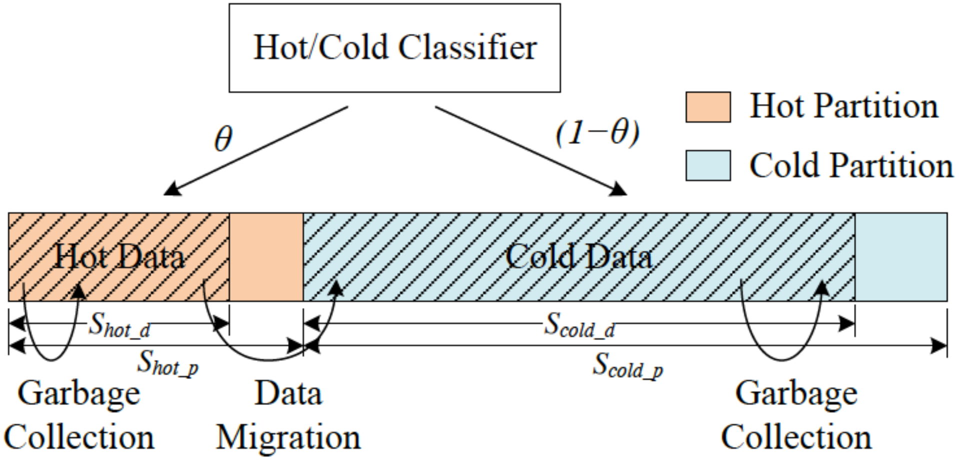
Wei Wang[1], Tao Xie[2], Deng Zhou[1]

[1]Computational Science Research Center, San Diego State University
[2]Computer Science Department, San Diego State University

Partitioned SSD

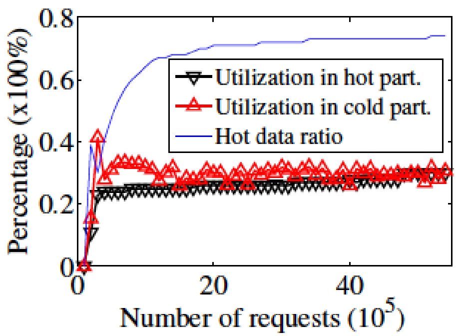
An SSD is typically split into multiple partitions

- Different partitions accommodate different types of data;
- It reduces write amplification;
- It improves performance;

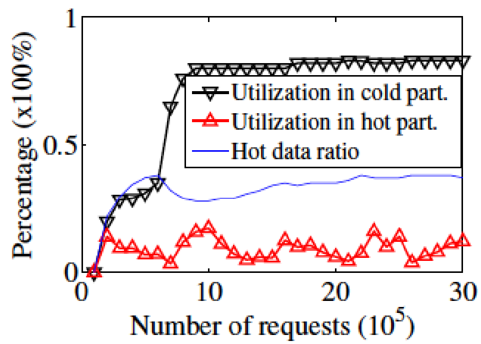


Partition Utilization

We noticed the utilizations of the two partitions may noticeably vary over time.



Fin1 trace [1]



prn0 in MSR traces [2]

The utilization of one partition affects the garbage collection overhead.

[1] J. Boukhobza, I. Khetib, and P. Olivier, "Characterization of OLTP i/o workloads for dimensioning embedded write cache for flash memories: A case study," in MEDI, Berlin, Heidelberg: Springer-Verlag, 2011,

[2] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," Trans. Storage, vol. 4, no. 3, pp. 10:1-10:23, Nov. 2008.

How to partition an SSD?

- To minimize the write amplification (WA);
- To maximize the overall performance.

They are not mutually exclusive.

Garbage Collection Cost

Hot partition:

$$C_{hot\_gc} = \left\lceil \mu_h \cdot \frac{N_p}{2} \right\rceil \cdot C_{hot\_pc} + C_e$$

Number of pages in a block (N_p)

Hot partition utilization (μ_h)

Page copy cost (C_{hot_pc})

Block erase cost (C_e)

Cold partition:

$$\mu = \frac{data\ size}{partition\ capacity}$$

$$C_{cold\_gc} = \left| \mu_c \cdot N_p \right| \cdot C_{cold\_pc} + C_e$$

After a garbage collection, there are $\lfloor (1-\mu_h) \cdot N_p / 2 \rfloor$ and $\lfloor (1-\mu_c) \cdot N_p \rfloor$ free pages left in a free block in hot partition and cold partition, respectively.

Page Programming Cost

Hot partition:

$$C_{hot\_pw} = \frac{C_{hot\_gc}}{\lfloor (1-\mu_h)\cdot N_p / 2 \rfloor} + C_{pf}$$

fast page programming cost

Cold partition:

$$C_{cold\_pw} = \frac{C_{cold\_gc}}{\lfloor (1-\mu_c)\cdot N_p \rfloor} + C_{pa}$$

average page programming cost

The page write cost is a function of utilization.

Partition Utilization

$$\mu = \frac{data\ size}{partition\ capacity}$$

The utilization of a victim block ($\tilde{\mu}$) is typically lower than that of a partition because the victim block has the lowest number of valid pages.

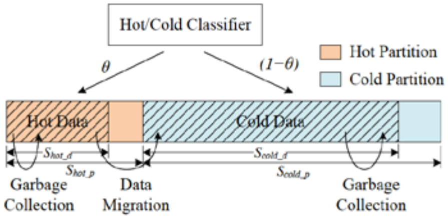
$$\mu = \frac{\tilde{\mu} - 1}{\ln(\tilde{\mu})} \quad [1]$$

[1] Y. Oh and J. Choi et al. Caching less for better performance: Balancing cache size and update cost of flash memory cache in hybrid storage systems. In Proceedings of the 10th USENIX conference on File and Storage Technologies, FAST'12, 2012.,

Data Migration Cost

Cold partition to hot partition:

- Hot data is mis-dispatched onto the cold partition.
- Its update can be re-distributed to the hot partition.
- The migration cost is zero.



Hot partition to cold partition:

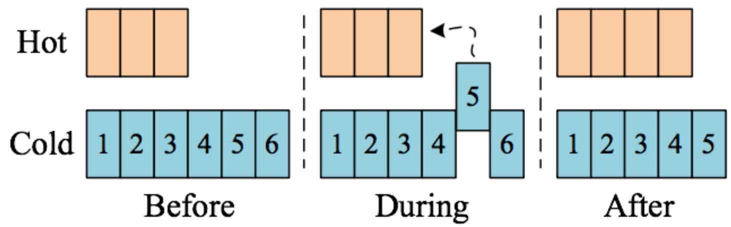
- Cold data is mis-dispatched onto the hot partition.
- Cold data is seldom updated.

$$C_m = C_r + C_{cold\_pw}$$

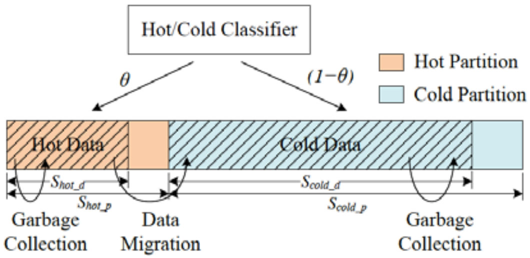
Detecting process consumes processor time, which is trivial compared with flash write operation.

The Programming Cost Model

A Dynamic Partitioning Method



The initial hot and cold partition capacities are: βS_{tot} and $(1-\beta)S_{tot}$.



Assume that hot partition capacity is increased by ΔS. Therefore, cold partition size is decreased by $2\Delta S$.

The Programming Cost Model

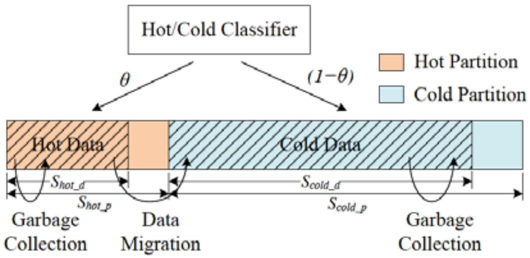
After re-partitioning, utilization of each partition becomes:

$$\mu_h' = \frac{\beta \mu_h S_{tot}}{\beta S_{tot} + \Delta S} \qquad \mu_c' = \frac{(1-\beta)\mu_h S_{tot}}{(1-\beta)S_{tot} - 2\Delta S}$$

$$\mu_c' = \frac{(1-\beta)\mu_h \mu_h'}{(1-\beta)\mu_h' - 2\beta\mu_h + 2\beta\mu_h'}$$

The lower and upper bound

$$\frac{2\beta\mu_h}{1+\beta-(1-\beta)\mu_c} < \mu_h' < 1$$



The Programming Cost Model

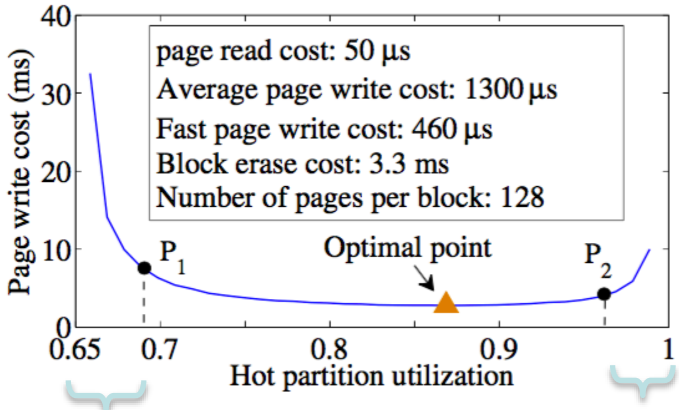
Assume that θ percent of total data are classified as hot data. Also assume that λ percent hot data are mis-dispatched.

$$C'_{overall} = \theta C'_{hot\_pw} + (1-\theta)C'_{cold\_pw} + \lambda\theta C'_m$$

Initial hot and cold partition utilization: $\mu_h = \mu_c = 0.8$;

$\theta = 0.3$;

$\lambda = 0.01$.



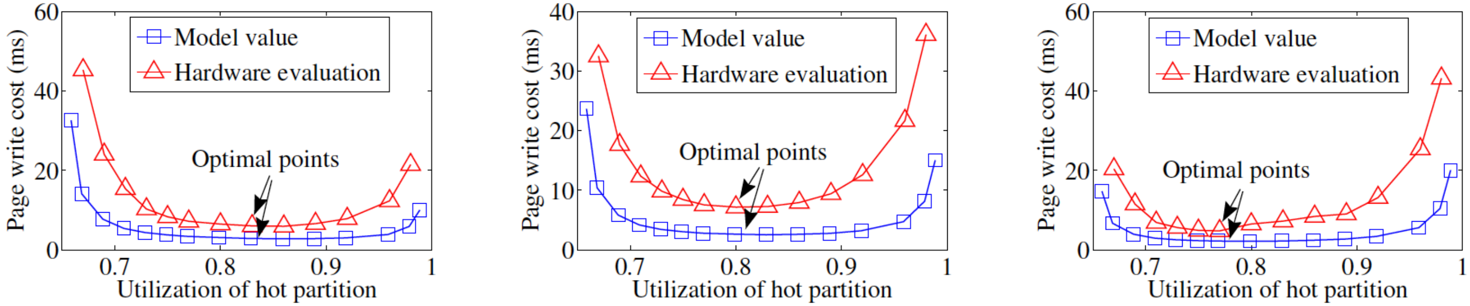
The overhead of GC in cold partition dominants the overall cost.

The overhead of GC in hot partition dominants the overall cost.

Model Verification

ComboFTL [1] on our FPGA evaluation board.

Name	*Total Req.*	*Updates*	≤ 4 *KB Req.*
ST1	10,485,233	80%	30%
ST2	10,485,233	80%	50%
ST3	10,485,233	80%	70%



(a) Evaluation results under ST1. (b) Evaluation results under ST2. (c) Evaluation results under ST3.

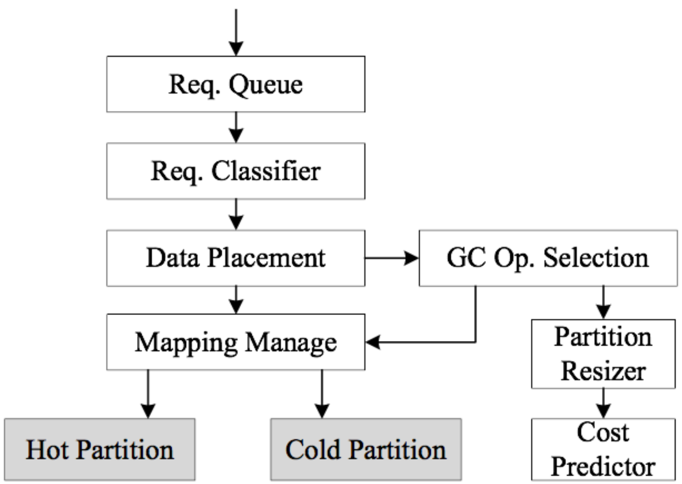
(1) same trend in performance change

(2) the hardware evaluation results are consistently larger

[1] S. Im and D. Shin. ComboFTL: Improving performance and lifespan of MLC flash memory using SLC flash buffer. Journal of Systems Architecture,56(12):641-653, 2010.

The BROMS FTL

Architecture of BROMS



Multiple GC selections:

(1) It does not reclaim any used blocks. It simply grabs an erased from the other partition.

(2) Valid data in the victim block are moved to the other partition.

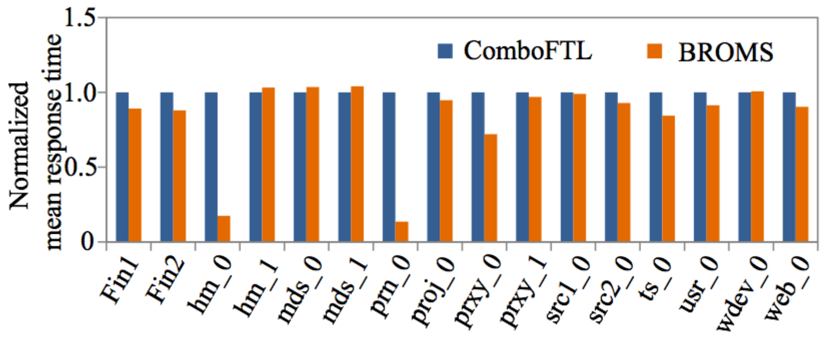
(3) Valid data in the victim block are moved within one partition (i.e., normal GC).

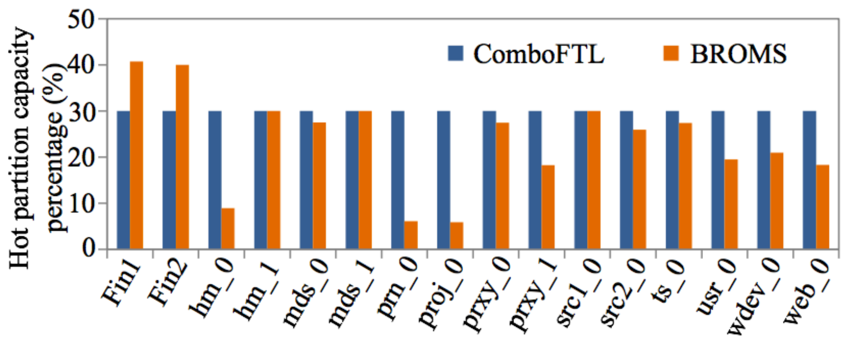
Hardware Evaluation

Two 8 GB MLC flash memory devices;

10% of the total number of blocks forms an over-provisioned space;

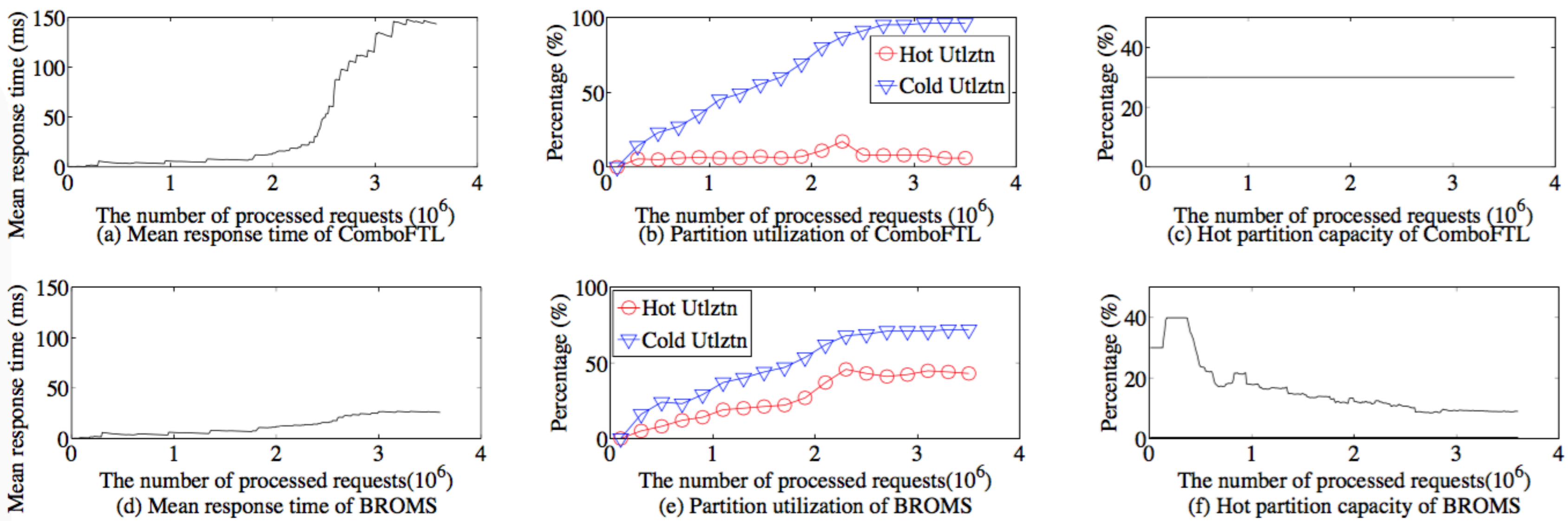
30% of the rest flash capacity is allocated for the hot partition.





Hardware Evaluation

A comparison between ComboFTL and BROMS under *hm_0*



(a) Mean response time of ComboFTL
(b) Partition utilization of ComboFTL
(c) Hot partition capacity of ComboFTL
(d) Mean response time of BROMS
(e) Partition utilization of BROMS
(f) Hot partition capacity of BROMS

Summary

Different workloads running on a fixed partitioning configuration lead to various levels of performance;

We demonstrate that for each workload there always exists a best partition configuration that offers optimal overall performance;

A dynamic partitioning method is developed to help an SSD deliver its best performance;

Our BROMS demonstrates the effectiveness of the dynamic partitioning method.

Thank You!