Curling-PCM: Application-Specific Wear Leveling for Phase Change Memory based Embedded Systems

Duo Liu[1], Tianzheng Wang[2], Yi Wang[3], Zili Shao[3], Qingfeng Zhuge[1], Edwin Sha[1]
[1]Chongqing University
[2]University of Toronto
[3]The Hong Kong Polytechnic University
liuduo@cqu.edu.cn

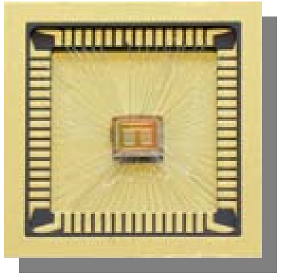
Outline

- Introduction
  - Phase Change Memory (PCM)
  - PCM-based Embedded Systems
- Curling-PCM: Application-specific wear-leveling
  - Full-Curling
  - Partial-Curling
- Evaluation
- Conclusion

PCM (Phase Change Memory)

- Why PCM (Phase Change Memory)?
  - Non-volatile, high density, low standby power…
  - Better than NOR/NAND flash in almost all metrics
  - Performance close to DRAM but with better scalability
  - NOR/DRAM replacement: PCM chips have been shipped by Micron (128Mb SPI/P8P; 1Gb LPDDR).

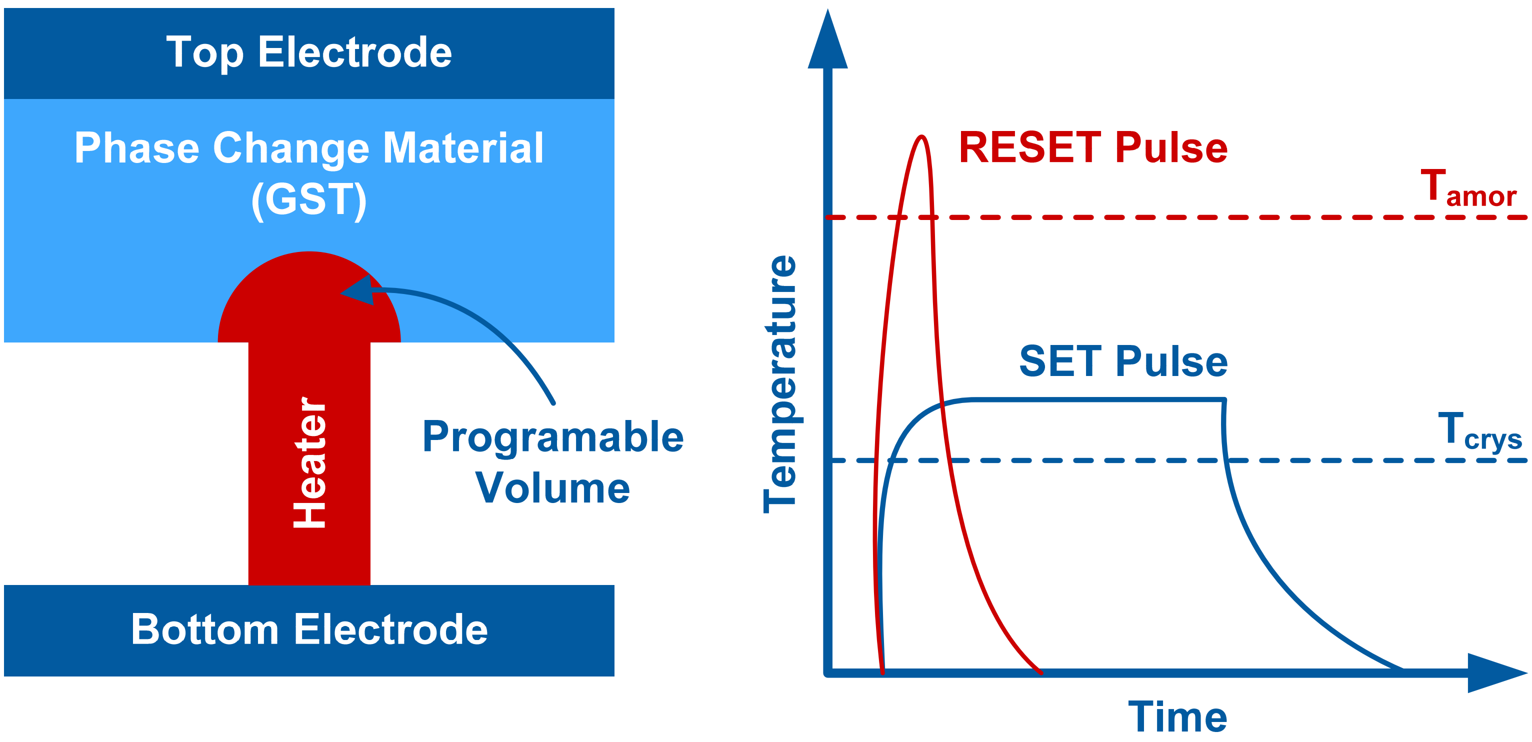
Samsung's PCM



IBM's PCM

How PCM works?

- Two states: amorphous (0) ⟷ crystalline (1)



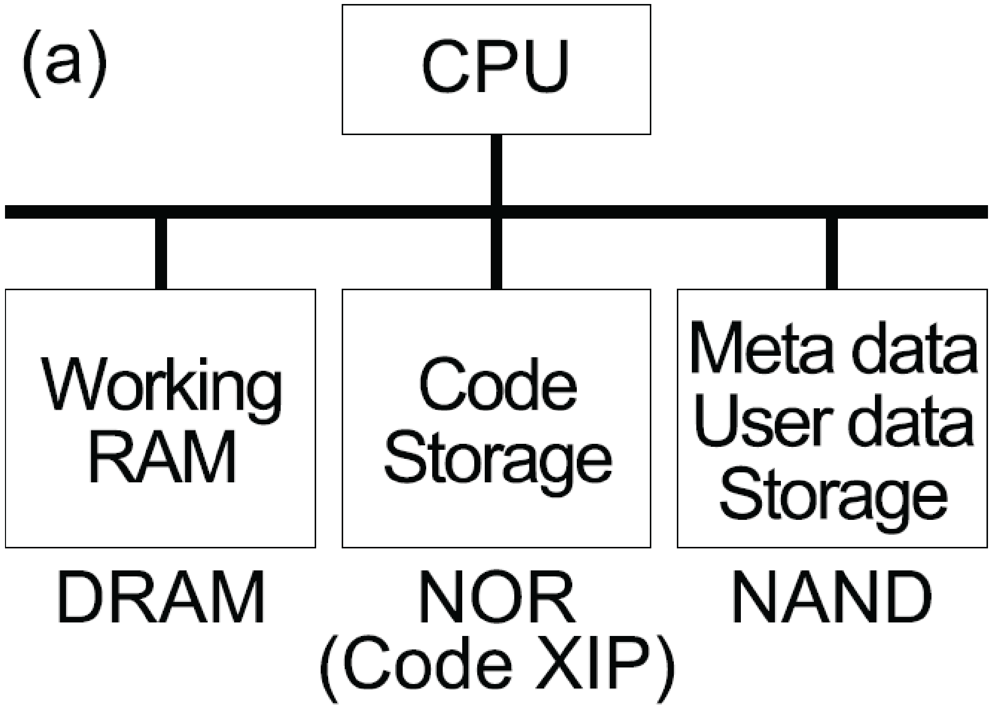
Comparison of DRAM, PCM and NAND

- PCM has limited endurance (10^6~10^8 writes – DRAM: 10^{16})

	DRAM	PCM	NAND
Non-Volatile	No	Yes	Yes
Erase Unit	Bit	Bit	Block
Power	~W/GB	100-500mW/die	~100mW/die
Write Latency	<10ns	50-120ns	~100us
Write Voltage	2.5V	15V	3V
Read Latency	50ns	50-100ns	10-25us
Read Voltage	1.8v	<3V	2V
Endurance	>10^{16}	10^6-10^8	10^4-10^5
Retention	64ms	>10 years	>10 years

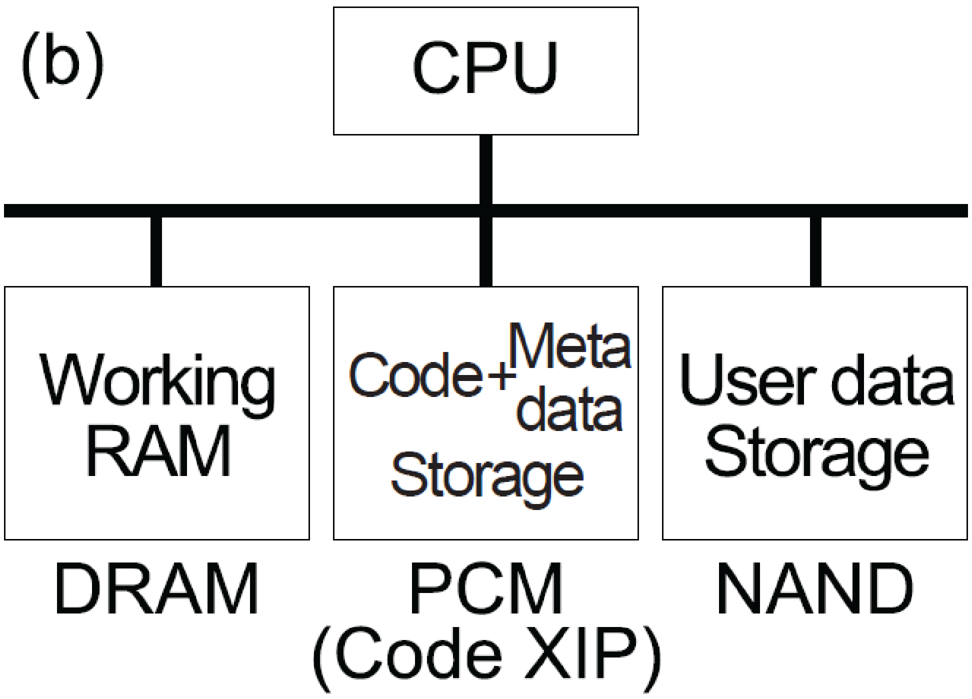
(Data gathered from International technology roadmap for semiconductors - ITRS 2009)

Current Embedded Storage Architecture



DRAM + NOR flash + NAND flash

PCM-based Embedded Systems



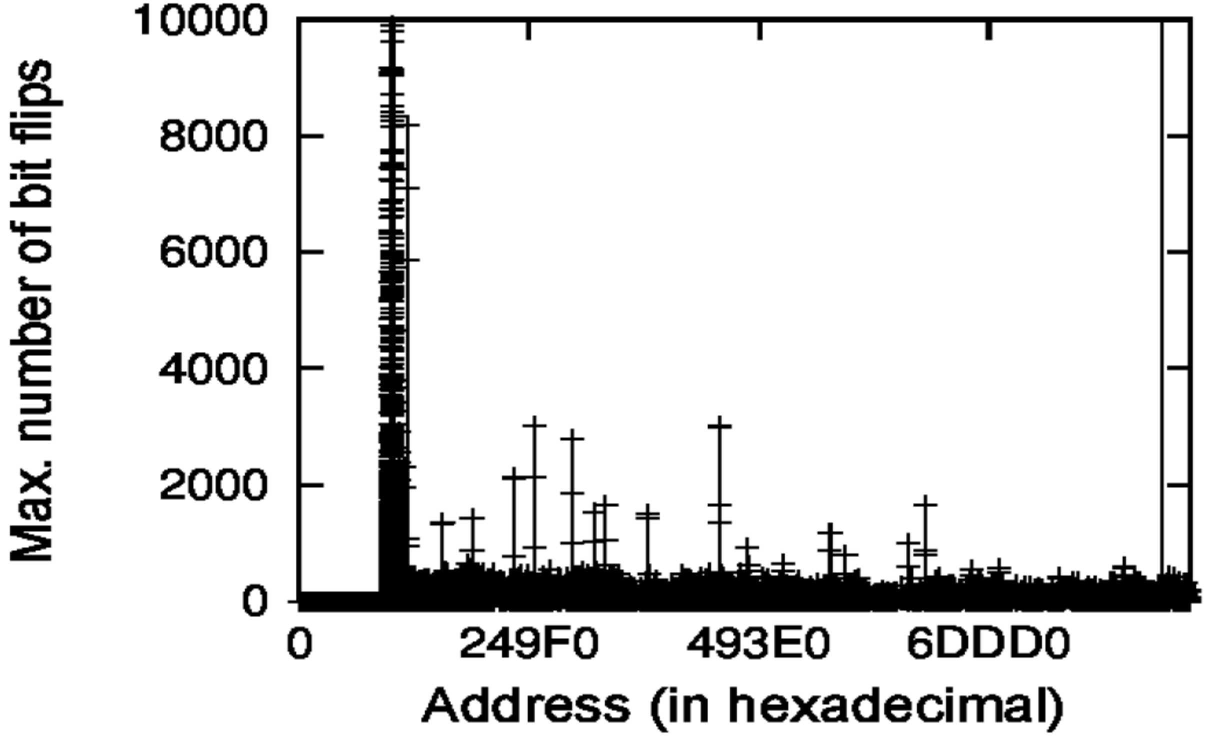
The embedded storage architecture by utilizing PCM as NOR flash replacement with the exploration of its extra space

The Previous Work

- PCM management has been intensively studied in the general-purpose computing field.
  - Start-Gap [MICRO09], Differential write [ISCA 09-Zhang], Security refreshing [ISCA09], PCM SSD[HotStorage11, HPCA10], etc.
- Embedded systems (application-oriented): limited resources to manage PCM.
  - Hybrid SPM with PCM/SRAM [DATE11], Data scheduling/recomputation [DAC10].
  - Reduce energy [DAC11]
  - PCM-FTL [RTSS11]

Motivation

- Distribution of write activities with Start-Gap[1]



[1] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali, "Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling," in MICRO, 2009.

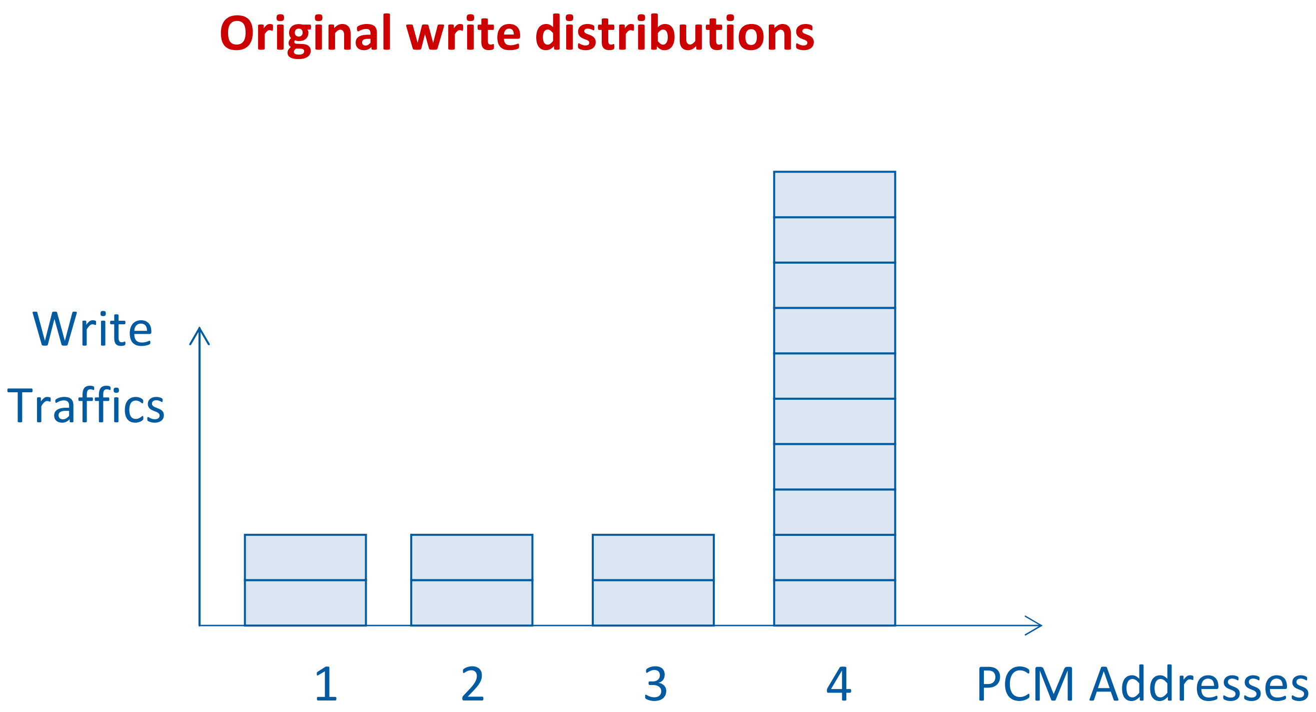
Curling-PCM

- Curling-PCM: evenly distribute write activities by utilizing application-specific features.

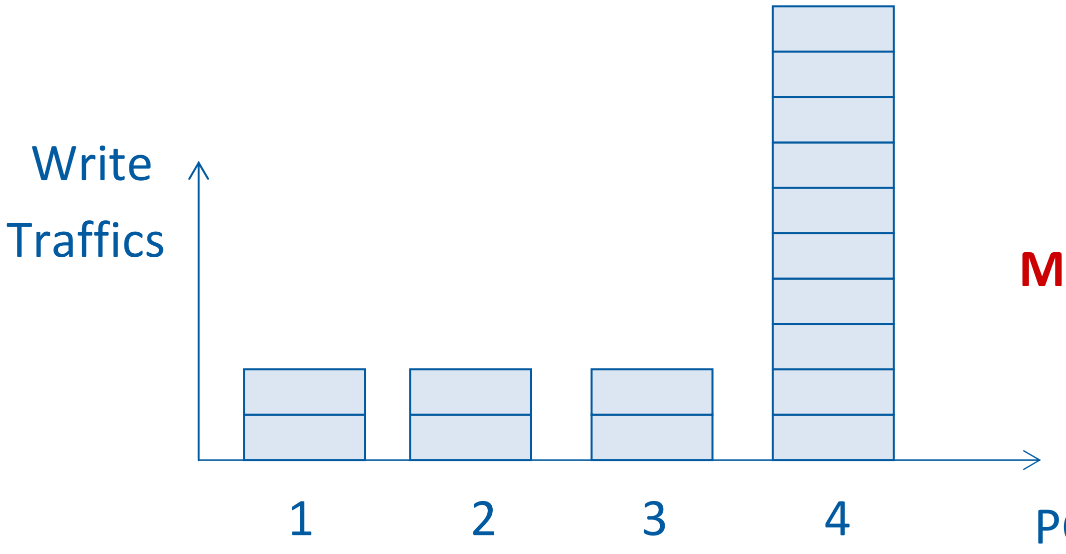
- Basic idea:
  - Identify hot areas by analyzing update frequencies of an application.
  - Periodically move hot areas across PCM (threshold satisfied) so write traffics can be evenly distributed.

Why moves hot areas not cold areas?

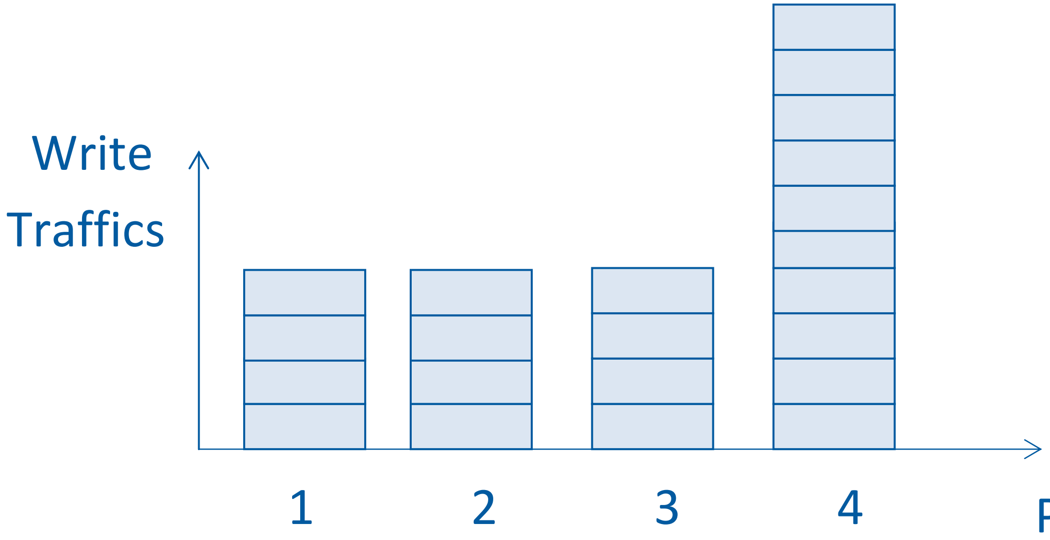
- Moving hot areas can more evenly distribute write traffics than moving cold areas.



Why moves hot areas not cold areas?



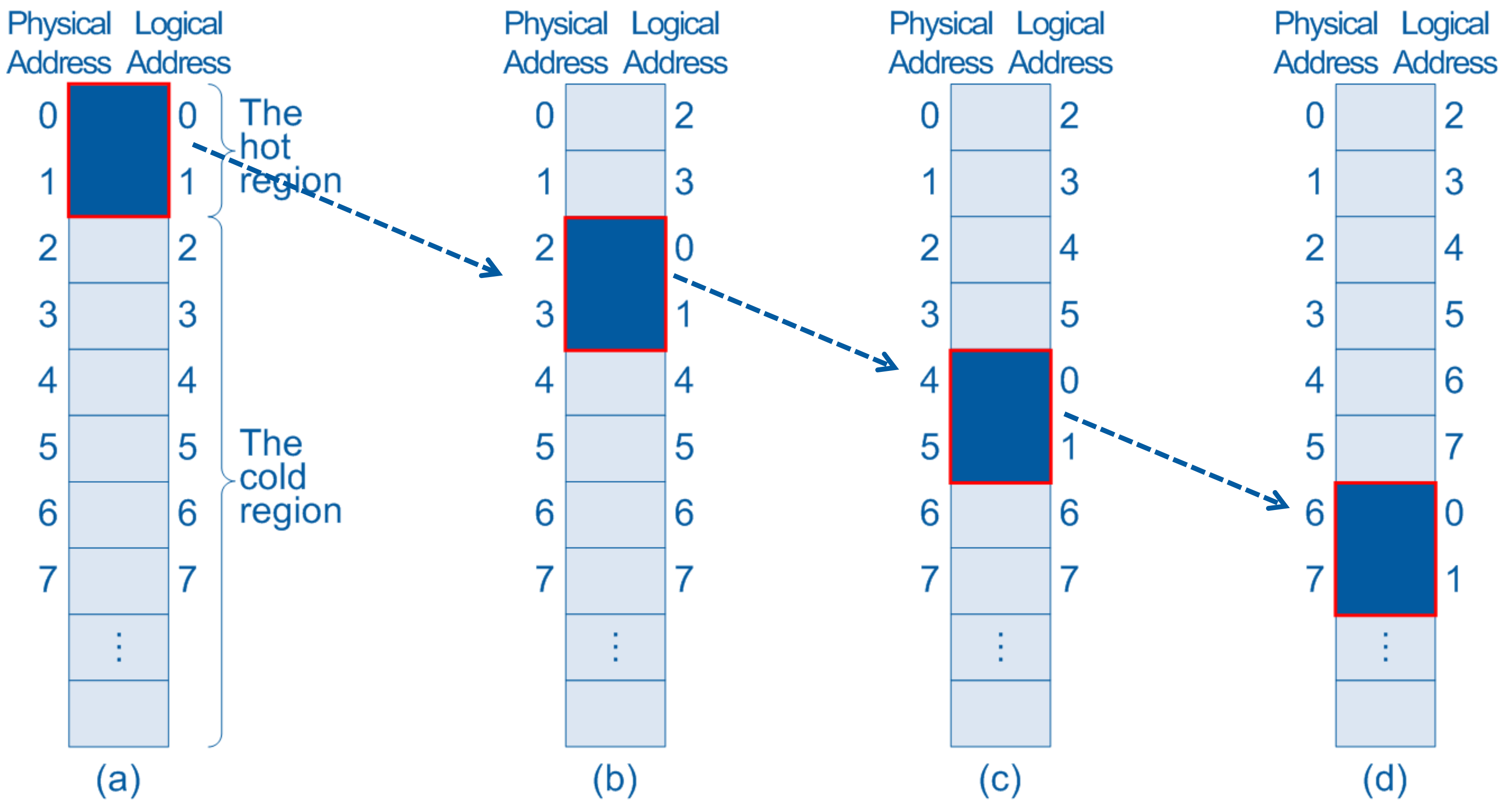
Move empty lines (start-gap)



Move hot areas

Full Curling

- Group hot areas into a hot region and periodically move it



Full Curling Address Translation

- Three registers are needed to handle address translation.

$$PA = \begin{cases} (LA + R\_HStart) \bmod Len \\ \qquad\qquad \text{if } LA \in hot\ region, \\ (LA + R\_HStart + \ Len - R\_CStartL) \bmod Len \\ \qquad\qquad \text{if } LA < R\_CStartL, \\ (LA + R\_HStart + \ HLen - R\_CStartL) \bmod Len \\ \qquad\qquad \text{if } LA \geq R\_CStartL. \end{cases}$$

PA: Physical address
LA: Logical address
R_HStart: Current starting physical address of hot region
R_CStartL: The first logical address following the hot region
Len: Total length of hot and cold regions
HLen: Length of hot region

Full Curling Mapping Example

PA	LA
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

	1st	2nd	3rd	4th	5th	6th	7th
	LA	LA	LA	LA	LA	LA	LA
	2	2	2	0	4	4	4
	3	3	3	1	5	5	5
	0	4	4	4	0	6	6
	1	5	5	5	1	7	7
	4	0	6	6	6	0	2
	5	1	7	7	7	1	3
	6	6	0	2	2	2	0
	7	7	1	3	3	3	1
R_HStart:	**2**	**4**	**6**	**0**	**2**	**4**	**6**
R_CStartL:	**4**	**6**	**2**	**4**	**6**	**2**	**4**

LA	PA	PA	PA	PA	PA	PA	PA
0	2	4	6	0	2	4	6
1	3	5	7	1	3	5	7
2	0	0	0	6	6	6	4
3	1	1	1	7	7	7	5
4	4	2	2	2	0	0	0
5	5	3	3	3	1	1	1
6	6	6	4	4	4	2	2
7	7	7	5	5	5	3	3

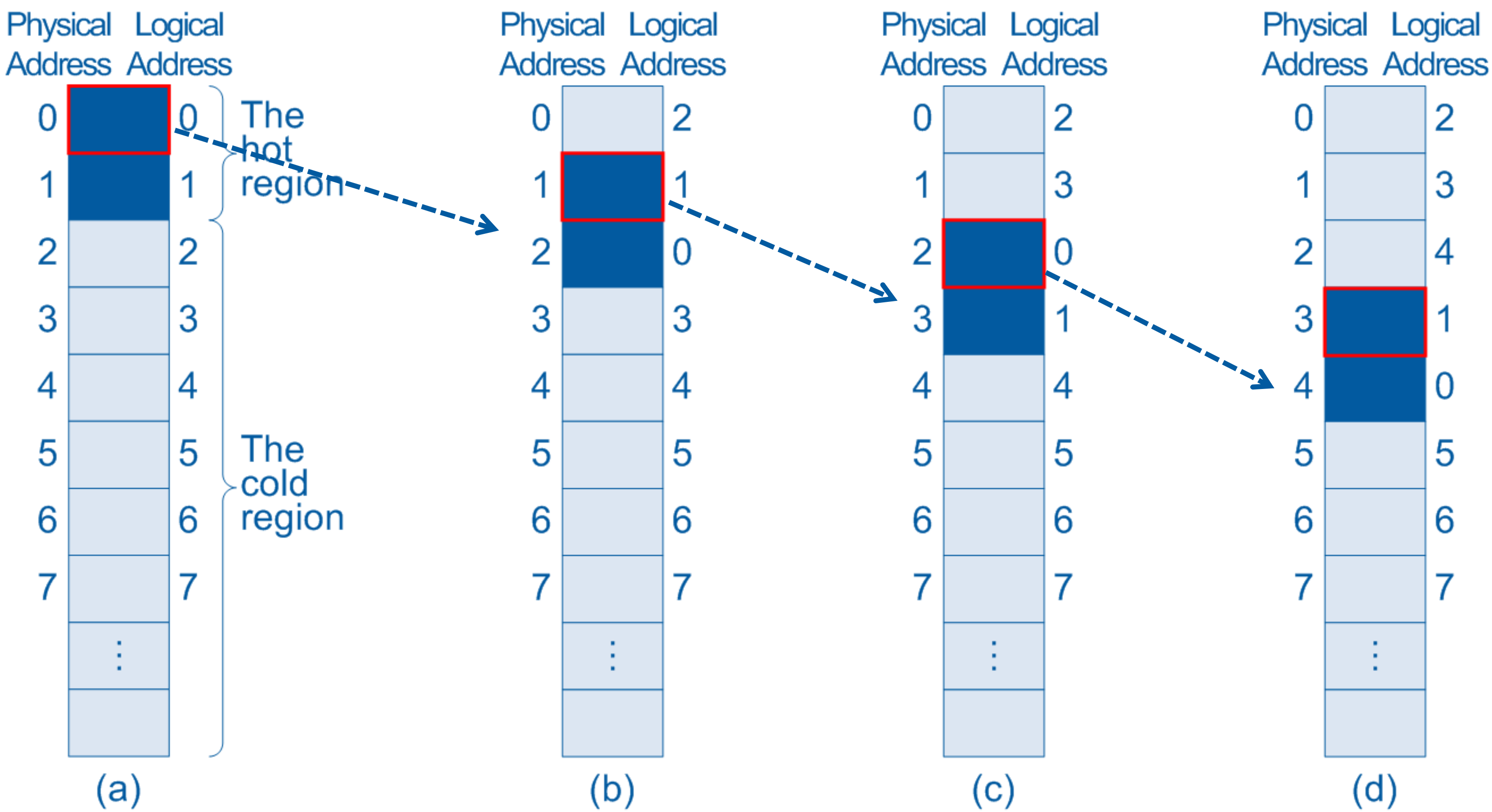
Partial Curling

- Full Curling moves all hot entries when handling a request, leading to *long* response time.

- Partial Curling
  - Divides hot region into smaller sub-regions
  - Move one sub-region following each request
  - Amortize overheads to multiple requests

Partial Curling

- Divide the movement of hot region into small steps, and each step can be interleaved with the service of read/write requests.



Partial Curling

Algorithm III.1 $PartialCurlingTranslate$(LA)

Input: LA - logical address of PCM entry
Output: PA - physical address of PCM entry

```
Let movedEntries = number of moved PCM entries so far.
Let prevOffset = previous starting hot region physical address.
Let newOffset = new starting hot region physical address.
Let hotRegion_StartLA = starting logical address of hot region.
if LA ≤ movedEntries + hotRegion_StartLA then
    // the entry has already been moved, use new parameters.
    Obtain PA by conducting address translation according to
    translation equations with newOffset.
else
    // the entry has not been moved, use the previous parameters.
    Obtain PA by conducting address translation according to
    translation equations with prevOffset.
end if
return PA
```

Partial Curling Mapping Example

PA	LA
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

	1st	2nd	3rd	4th	5th	6th	7th
	LA	**LA**	**LA**	**LA**	**LA**	**LA**	**LA**
	0	0	0	0	2	2	4
	1	1	1	1	1	3	3
	4	4	4	4	4	4	2
	3	5	5	5	5	5	5
	2	2	6	6	6	6	6
	5	3	3	7	7	7	7
	6	6	2	2	0	0	0
	7	7	7	3	3	1	1
R_HStart:	**4**	**6**	**0**	**2**	**4**	**6**	**0**
	PA	**PA**	**PA**	**PA**	**PA**	**PA**	**PA**
LA 0	0	0	0	0	6	6	6
1	1	1	1	1	1	7	7
2	4	4	6	6	0	0	2
3	3	5	5	7	7	1	1
4	2	2	2	2	2	2	0
5	5	3	3	3	3	3	3
6	6	6	4	4	4	4	4
7	7	7	7	5	5	5	5

Evaluation

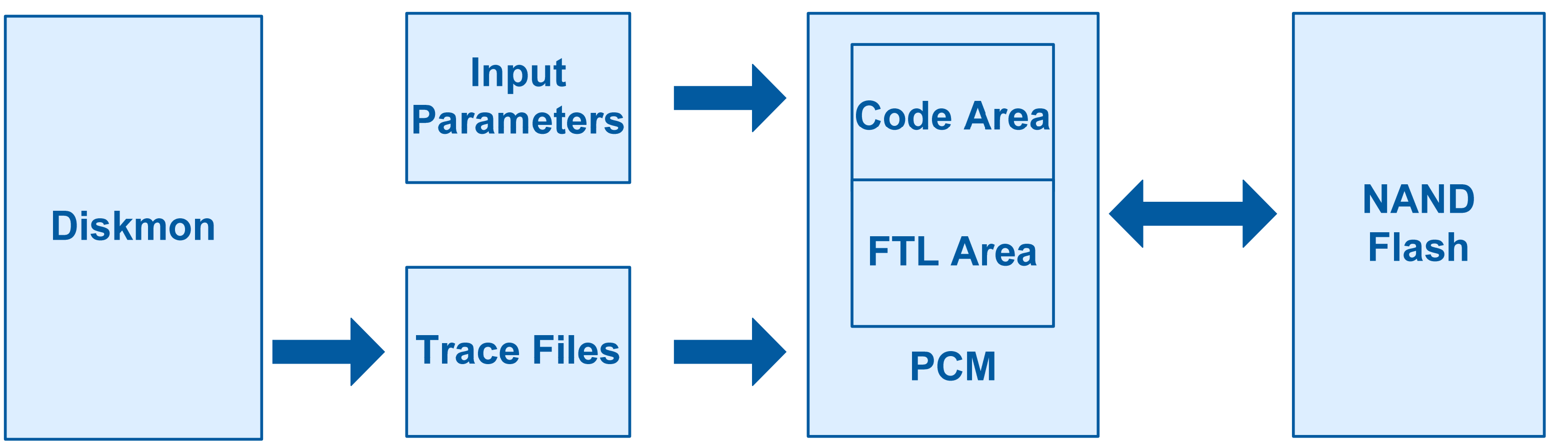
- Applications – the extra space of PCM is used to manage NAND flash.
- PCM is used to store the address mapping table of FTL in practice. This is used in our experiments.
- We compare three schemes: PCM-FTL, Start-Gap and our scheme—Curling-PCM.

Evaluation

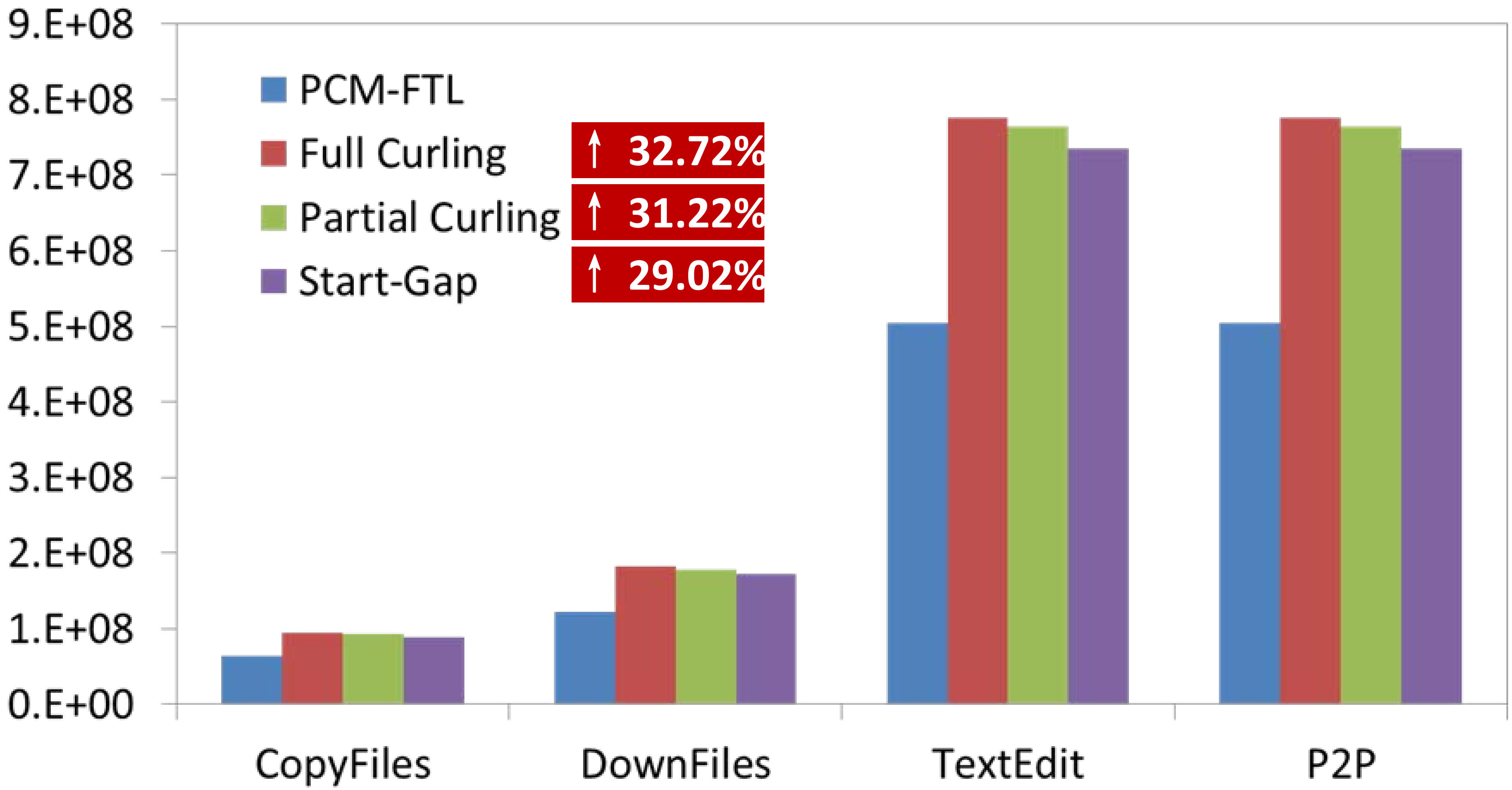
- PCM-FTL [RTSS-11]:
  - A two-level mapping mechanism— the page-level mapping for infrequent updates, and a tiny buffer with block-level mapping for sequential updates.
  - The tiny buffer becomes very hot.
- Start-Gap [Micro-09]
  - Employ an additional empty line as “gap” and move it periodically.
  - In our experiments, one line is 4 bytes, and the threshold is set to 100 writes (same as Micro-09 paper).
- Curling-PCM:
  - The tiny buffer and a few page table entries are hot so they are grouped as “the hot region”.
  - The hot region has 2000 lines, and the threshold is 20,000 writes.

Evaluation

- Experimental Setup
  - Simulation Platform: Linux 2.6.29
    - PCM chip (32 Mb)
    - NAND flash memory (1GB)
- Traces:
  - CopyFiles, DownFiles, TextEdit, P2P
- Metrics:
  - Maximum & Total number of writes of PCM cells

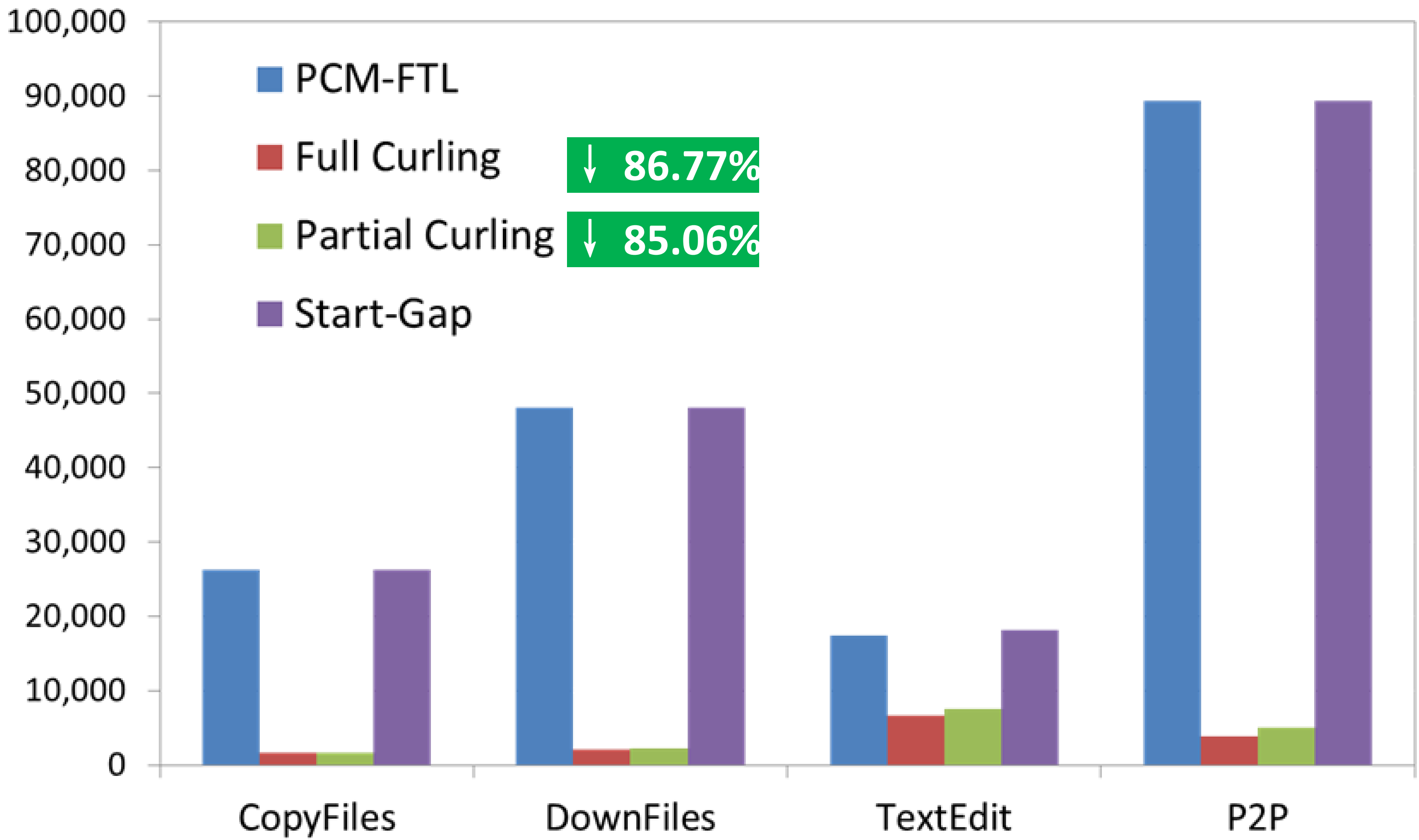


Experimental Results-1



COMPARISON OF THE TOTAL NUMBER OF BIT FLIPS

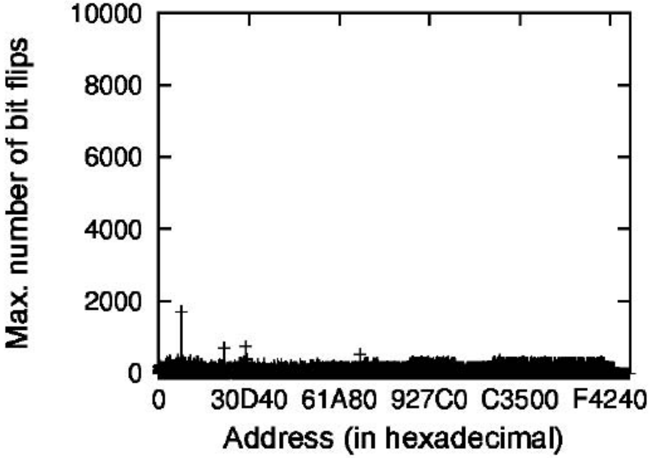
Experimental Results-2



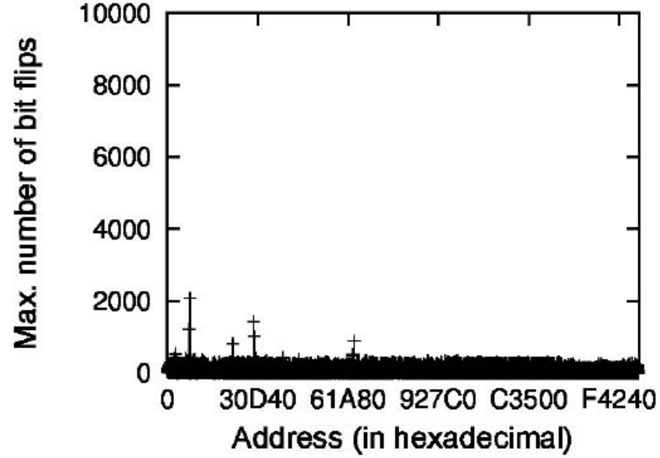
COMPARISON OF THE MAX NUMBER OF BIT FLIPS

Experimental Results-3

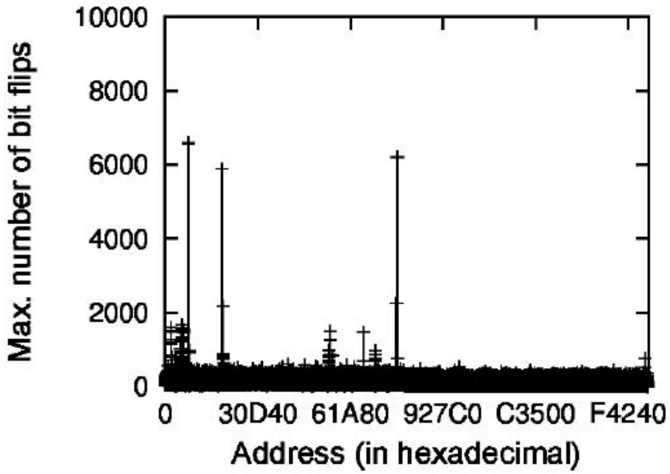
Distribution of the maximum number of bit flips by applying Curling-PCM with full curling



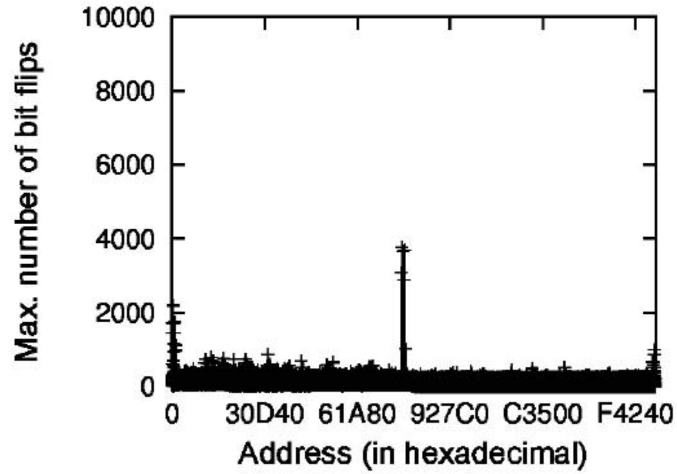
(a) CopyFiles



(b) DownFiles



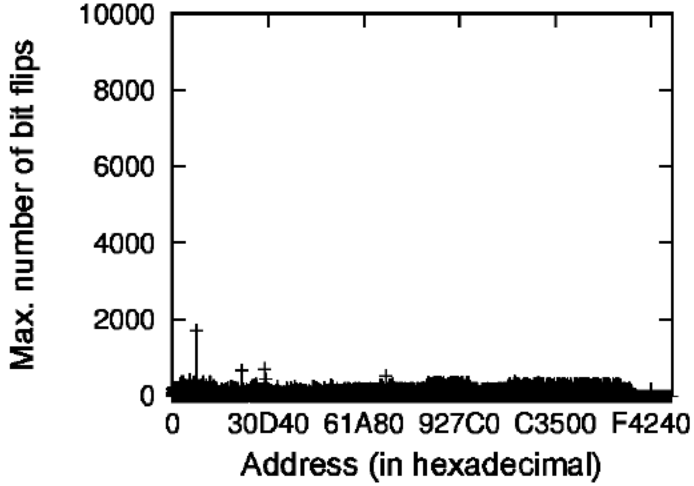
(c) TextEdit



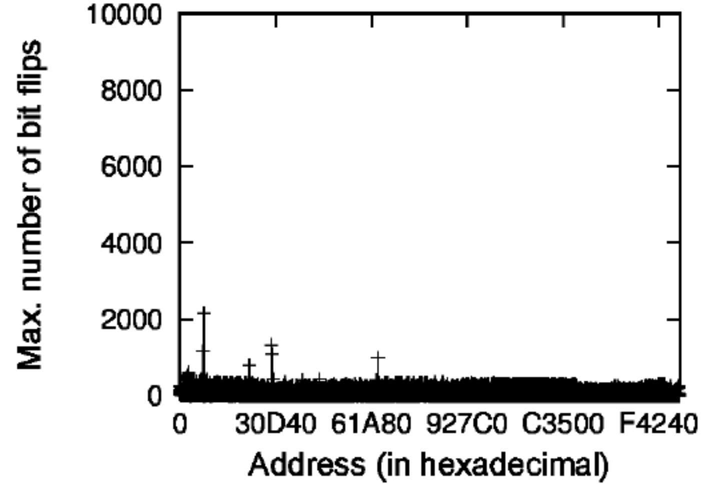
(d) P2P

Experimental Results-4

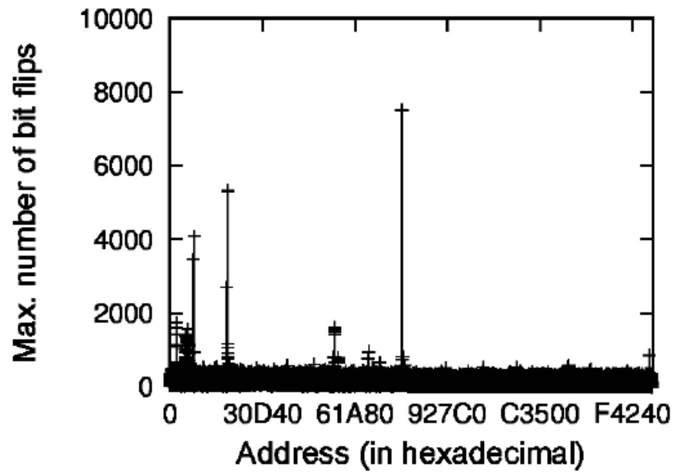
Distribution of the maximum number of bit flips by applying Curling-PCM with partial curling



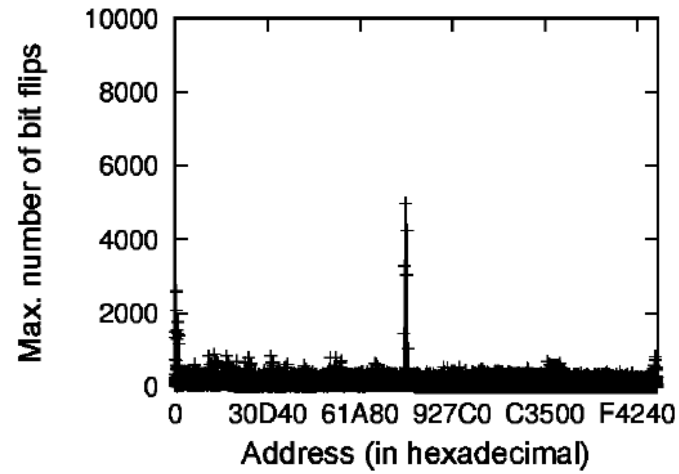
(a) CopyFiles



(b) DownFiles



(c) TextEdit



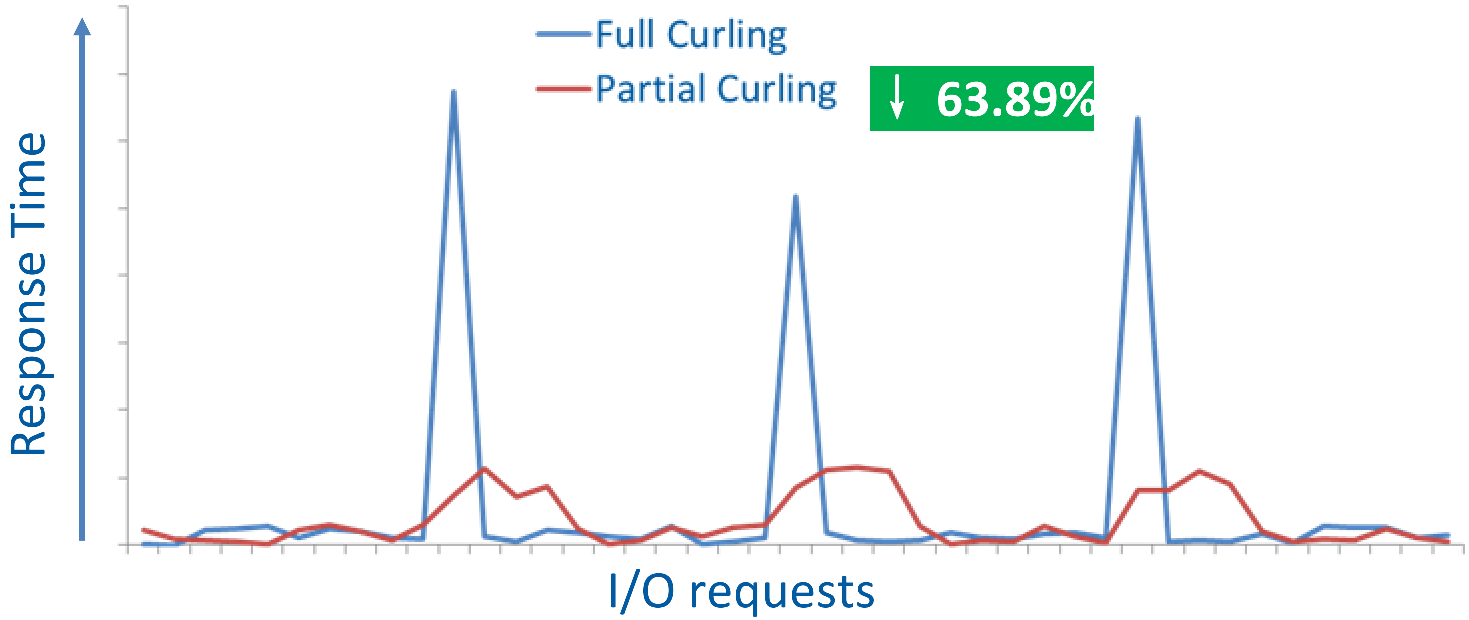
(d) P2P

Experimental Results-5

COMPARISON OF RESPONSE TIME VARIATION.

	Full Curling			Partial Curling			
Trace	Min (μs)	Max (μs)	Variation	Min (μs)	Max (μs)	Variation	Reduction over Full Curling
CopyFiles	0.13	2236.10	9380.94	0.13	961.80	2121.41	77.39%
DownFiles	0.05	3870.30	8140.94	0.05	1875.30	1866.43	77.07%
TextEdit	0.38	3160.80	9169.41	0.38	1729.20	2366.22	74.19%
P2P	0.18	18282.25	15249.94	0.18	16287.25	11145.33	26.92%
Average	0.18	6887.36	10485.31	0.18	5213.39	4374.85	63.89%

$$\text{Variation} = \frac{1}{Total\ Num\ Req} \times \sum_{i=1}^{Total\ Num\ Req} [T_i - \bar{T}]^2$$



Conclusion

- We have proposed an application-specific wear leveling technique, called Curling-PCM, to evenly distribute write activities across the PCM chip for better endurance.

- Experimental results show that Curling-PCM can effectively distribute write activities evenly and improve the lifetime of PCM chips compared to previous work.

Thank you!