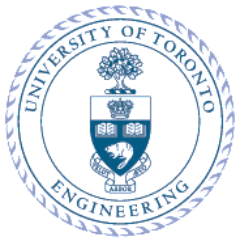
Building High-Performance Smartphones via Non-Volatile Memory: The Swap Approach

Kan Zhong[1], Tianzheng Wang[2], Xiao Zhu[1], Linbo Long[1], Duo Liu[1], Weichen Liu[1], Zili Shao[3] and Edwin H.-M. Sha[1]

[1]College of Computer Science, Chongqing University
[2]Department of Computer Science, University of Toronto
[3]Department of Computing, The Hong Kong Polytechnic University
{liuduo, kzhong1991}@cqu.edu.cn

http://nvm-swap.bitbucket.org/

Oct 15, 2014
New Delhi, India



Outline

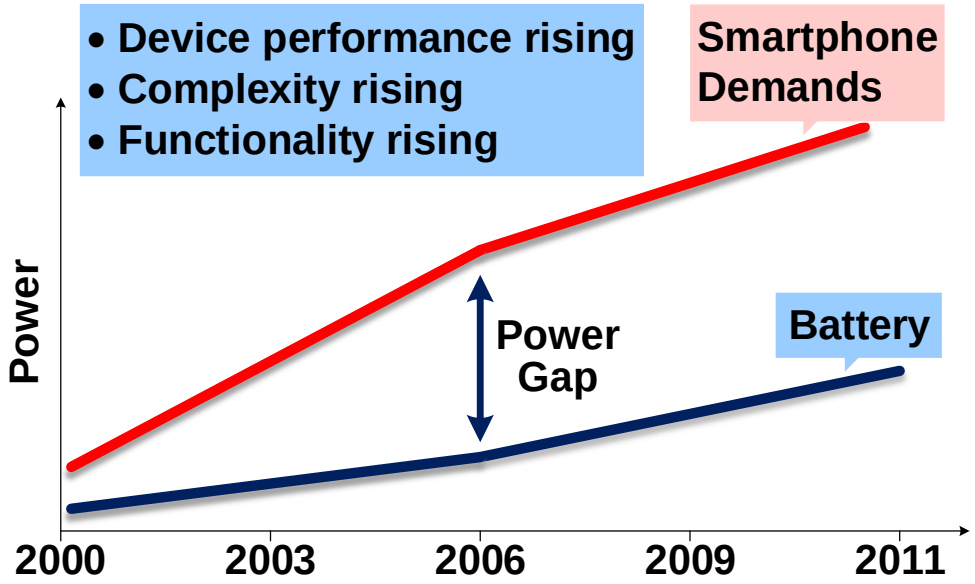
- Background
  - Performance and Energy
  - Swapping
- NVM-Swap: NVM based swapping
  - Copy-on-Write Swap-in (COWS)
  - WL Algorithm: Heap-Wear
- Evaluation
- Conclusion

Background — Performance

- Large DRAM is required for high performance smartphones

Background — Energy

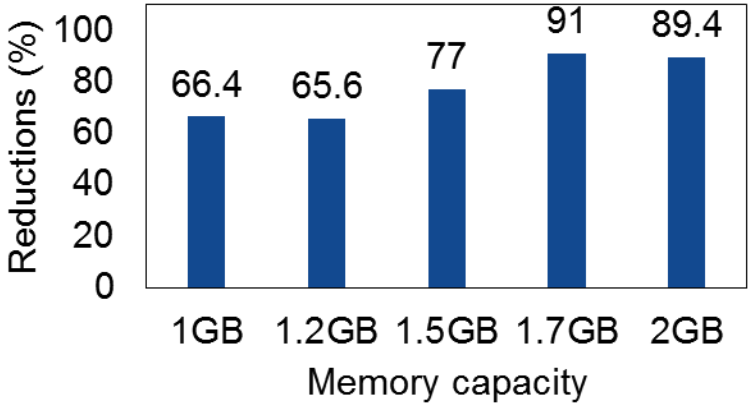
- More DRAM, more energy consumption
  - DRAM consumes up to 34.5% of overall energy
- Rise more pressure on battery lifetime



[1] Rice, A etc., Decomposing power measurements for mobile devices, *IEEE PerCom 2011.*
[2] A. Carroll etc., An analysis of power consumption in a smartphone, *USENIX ATC* 2010.
[3] Avneesh Agrawal, “Trends in Wireless Communications”, available at http://www.ieee infocom.org/2010/docs/Infocom2010_keynote.pdf.

Background — Swapping

- Write inactive pages to swap device
  - Extend main memory space



Reduce around **66% ~ 91%** of process terminations.

- Swapping is not practical in smartphones
  - Poor performance of smartphone internal flash
  - Limited program/erase cycles of flash memory

Why not use flash base swapping

- Swapping is disabled in smartphones

	SLC NAND flash	MLC NAND flash	DRAM
Endurance	**100,000 P/E cycles**	**30,000 P/E cycles**	**>10^{16}**
Read page	**25us**	**75us**	**~ns**
Program page	**200us**	**1600us**	**~ns**
Erase block	**700us**	**5ms**	**-**

- Process will be terminated directly if no memory space left
  - Bad user experience

[1] Micron, “1Gb x8, x16: NADN Flash Memory Features” MT29F1G08ABBDAH4 data sheet, 2010
[2] Micron, “128Gb 256GB, 512Gb Async/Sync Enterprise NAND Features” MT29E128G08CECAB data sheet, 2010

Outline

- Background
  - Performance and Energy
  - Swapping
- NVM-Swap: NVM based swapping
  - Copy-on-Write Swap-in (COWS)
  - WL Algorithm: Heap-Wear
- Evaluation
- Conclusion

Emerging NVM technology

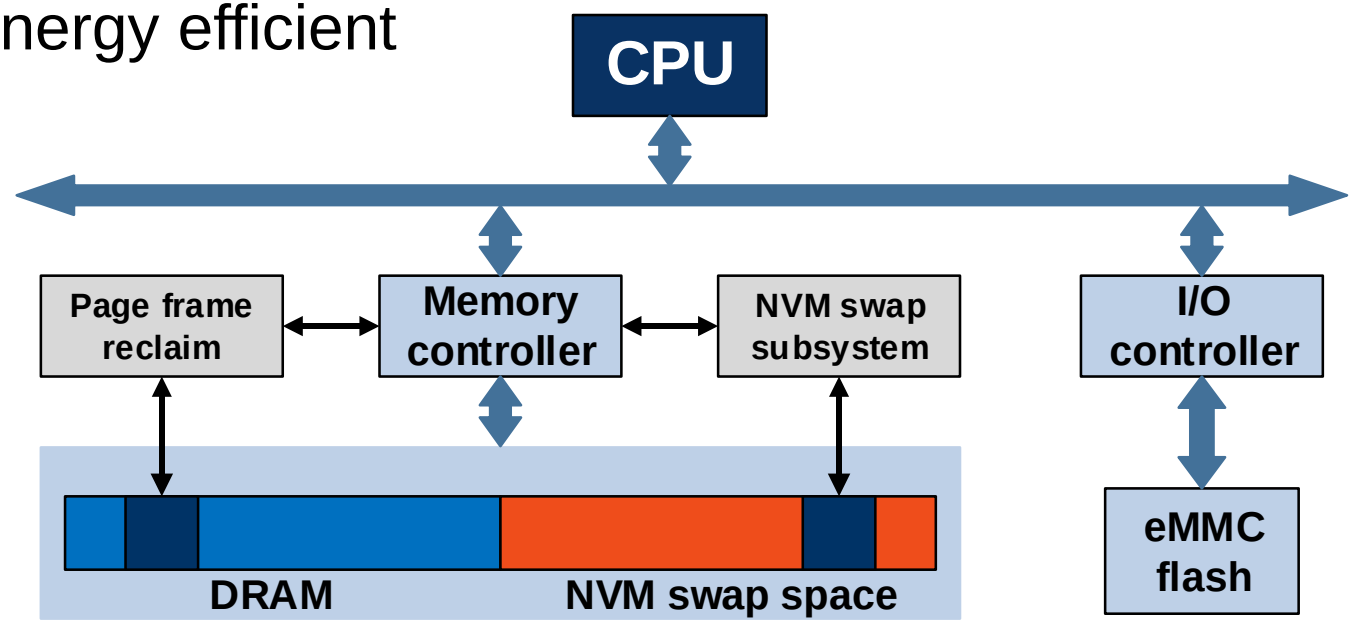
- Emerging Non-Volatile Memory (NVM)
  - Byte-addressable, high density, low standby power etc.
  - Near DRAM performance

	PRAM	RRAM	STT-RAM	DRAM
Endurance	**10^8**	**10^8**	**>10^{15}**	**>10^{16}**
Read time (ns)	**20–50**	**10–50**	**2–20**	**30**
Write / Erase time (ns)	**50 / 120**	**10–50**	**2–20**	**15**
Cell size (F^2)	**6–12**	**6–10**	**6–20**	**6-10**
High voltage required	**1.5-3V**	**1.5-3V**	**<1.5V**	**3V**
Refresh operation	**No**	**No**	**No**	**Yes**

Alexander Driskill-Smith, Latest Advances and Future Prospects of STT-RAM, *NVMW 2010*

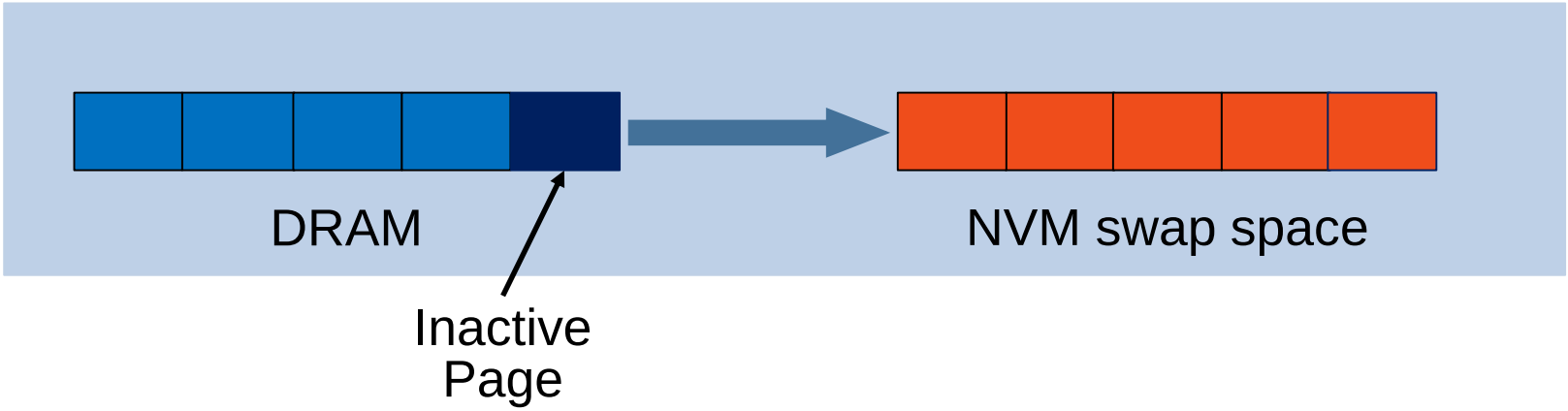
NVM-Swap: NVM based swapping

- Revisiting swapping in smartphones with emerging NVM
  - NVM is attached to the memory bus and used as swap area
  - High-performance
  - Energy efficient



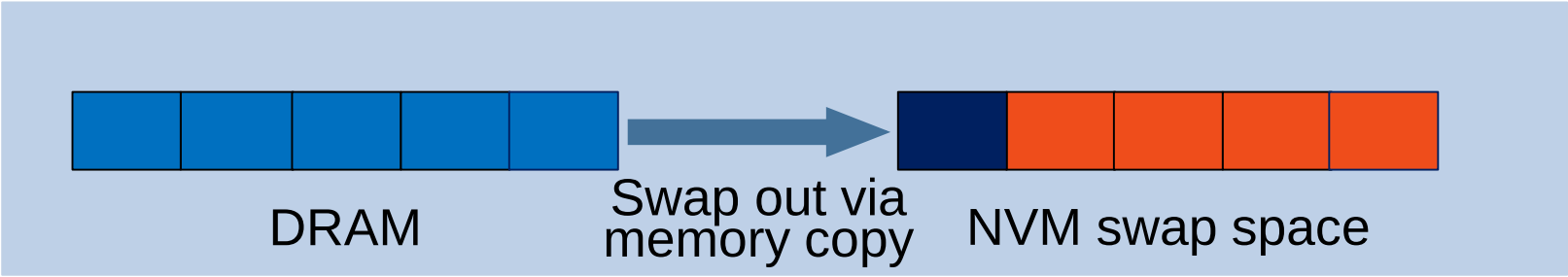
Swap in/out via memory interface

Swap out:



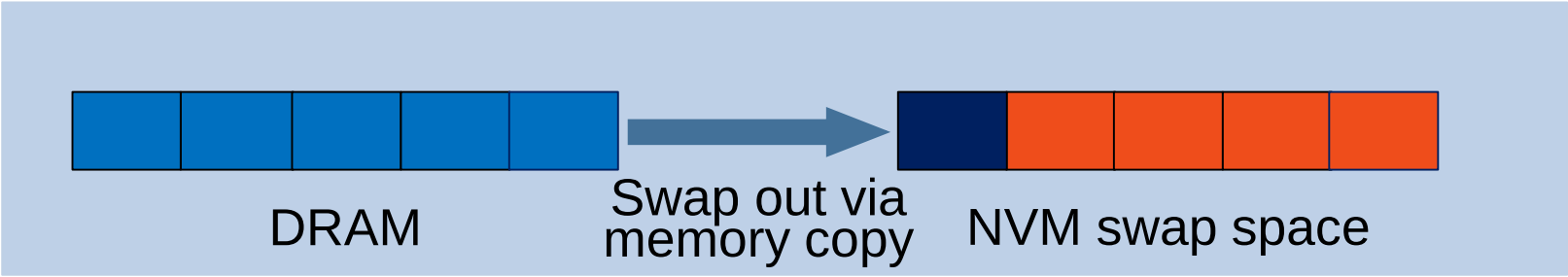
Swap in/out via memory interface

Swap out:

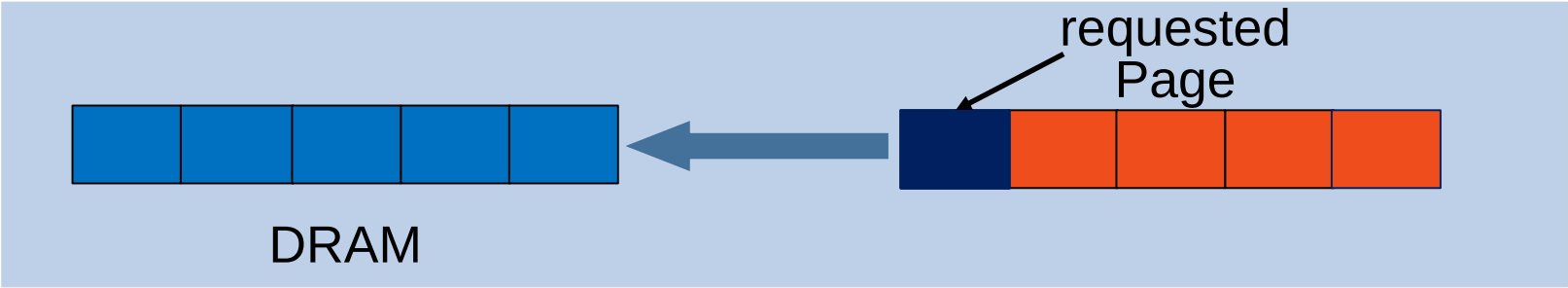


Swap in/out via memory interface

Swap out:

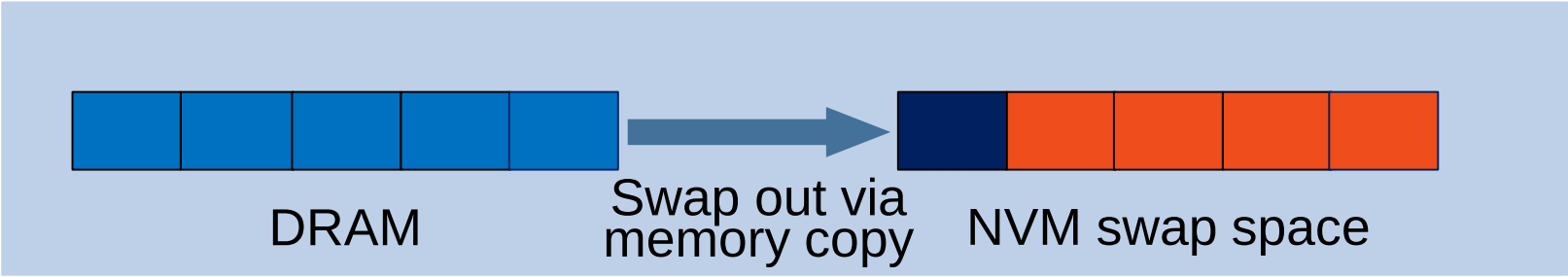


Swap in:

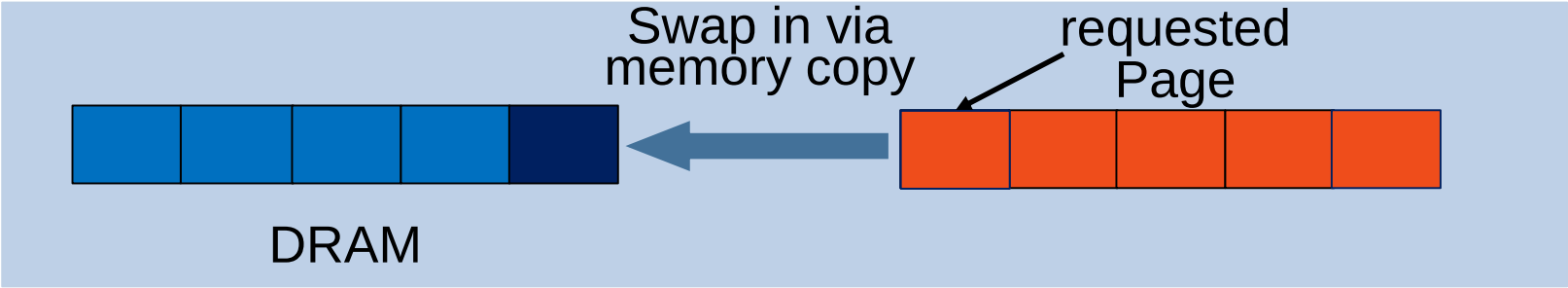


Swap in/out via memory interface

Swap out:

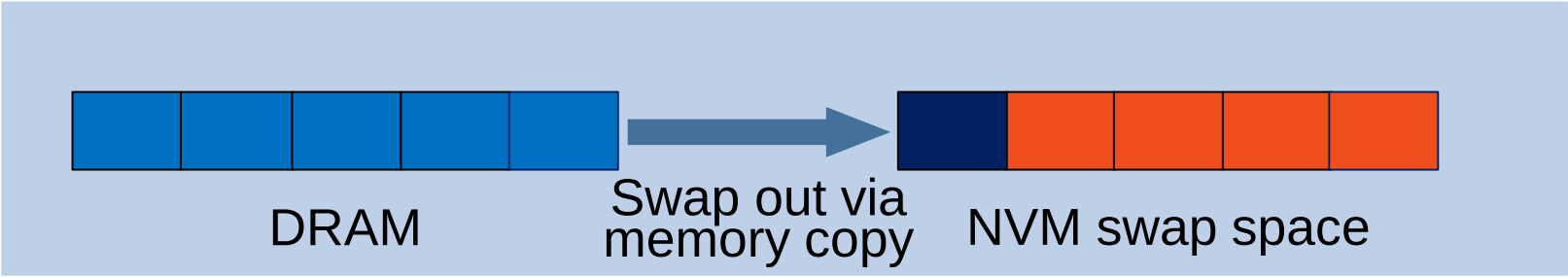


Swap in:

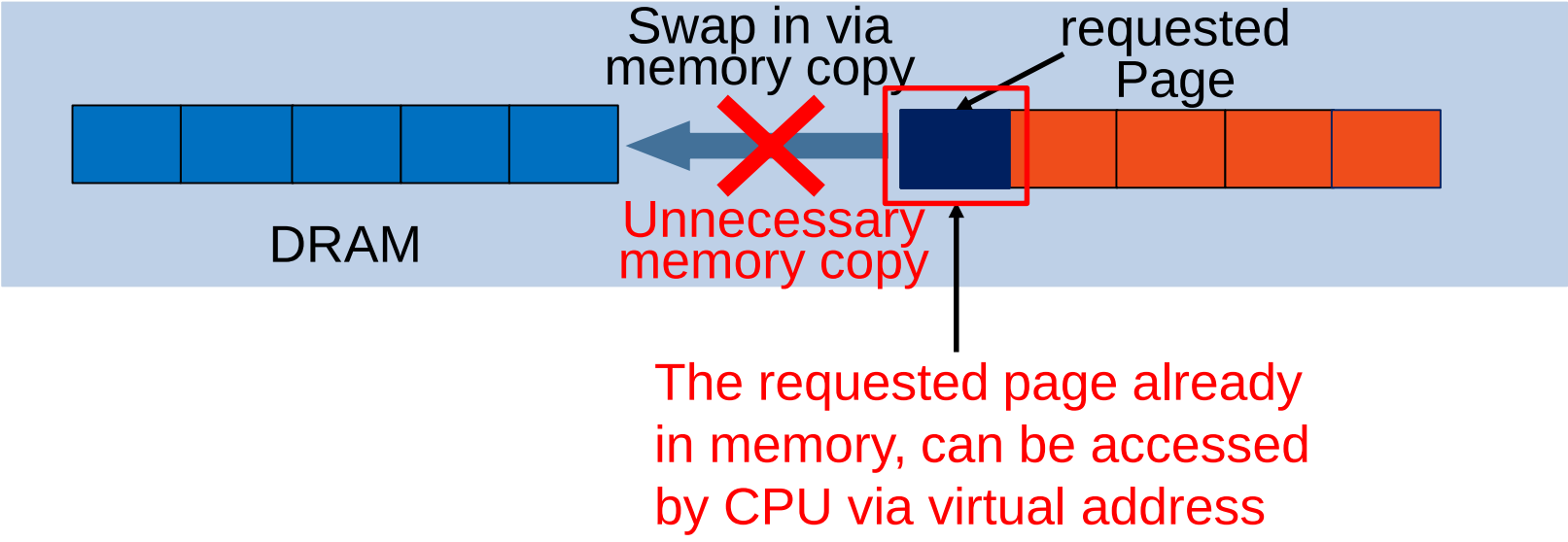


Swap in/out via memory interface

Swap out:

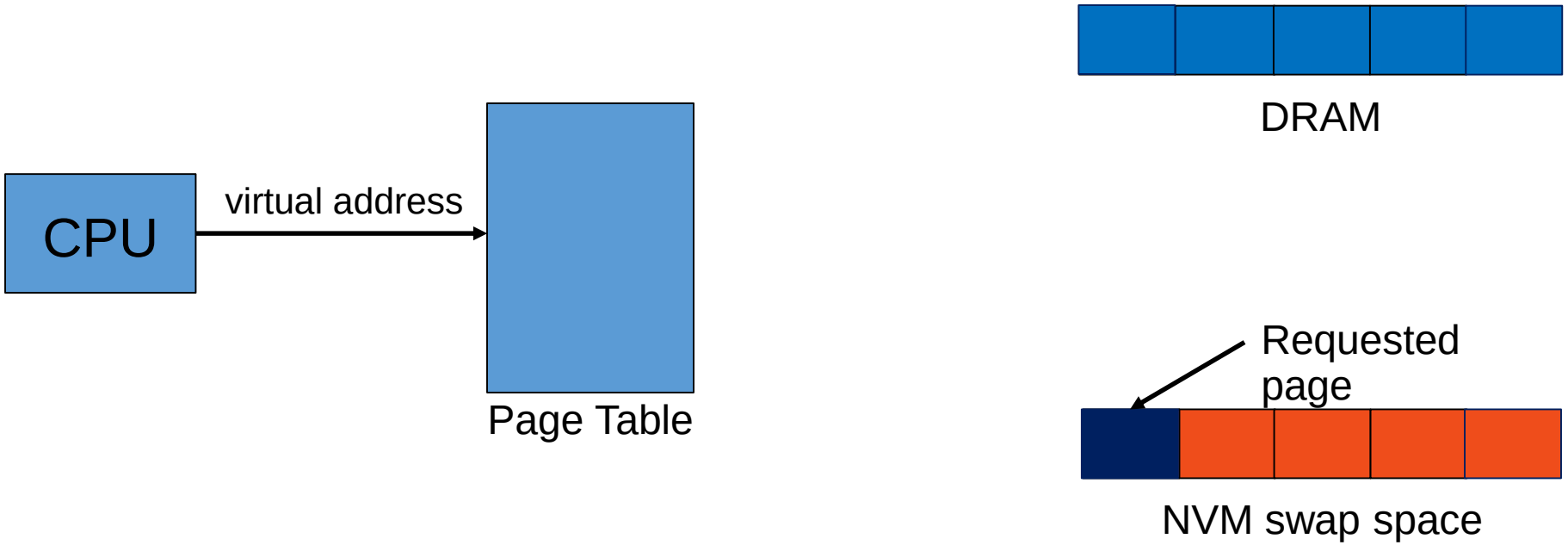


Swap in:



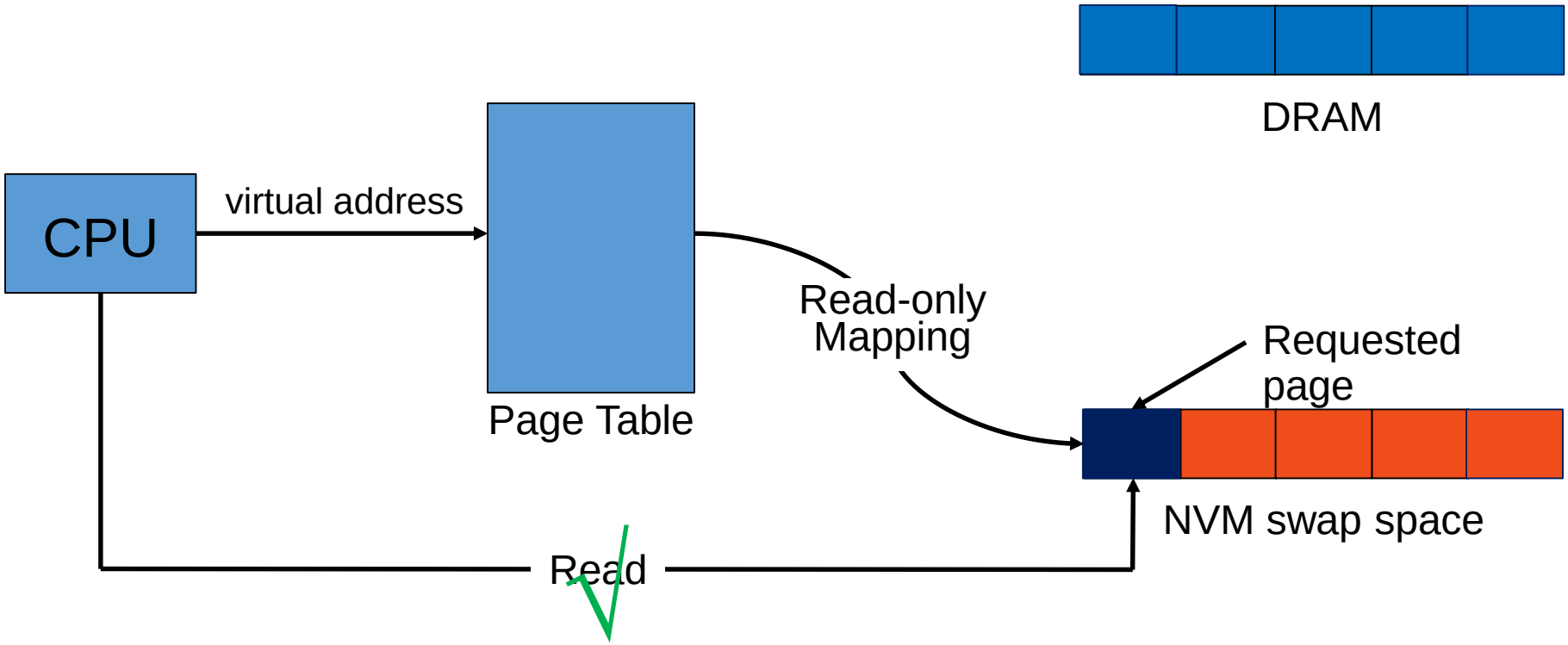
Copy-on-Write Swap-in (COWS)

- COWS: remove unnecessary copy operations



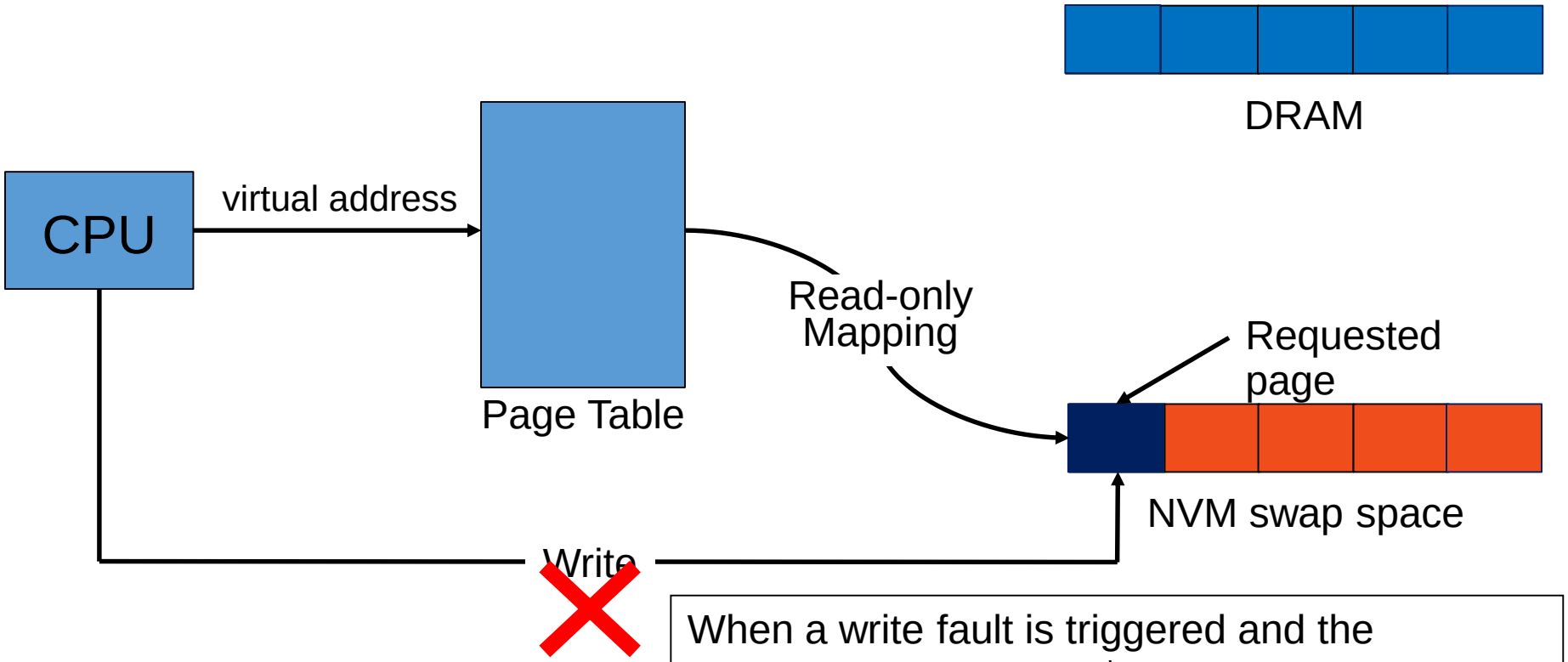
Copy-on-Write Swap-in (COWS)

- COWS: remove unnecessary copy operations



Copy-on-Write Swap-in (COWS)

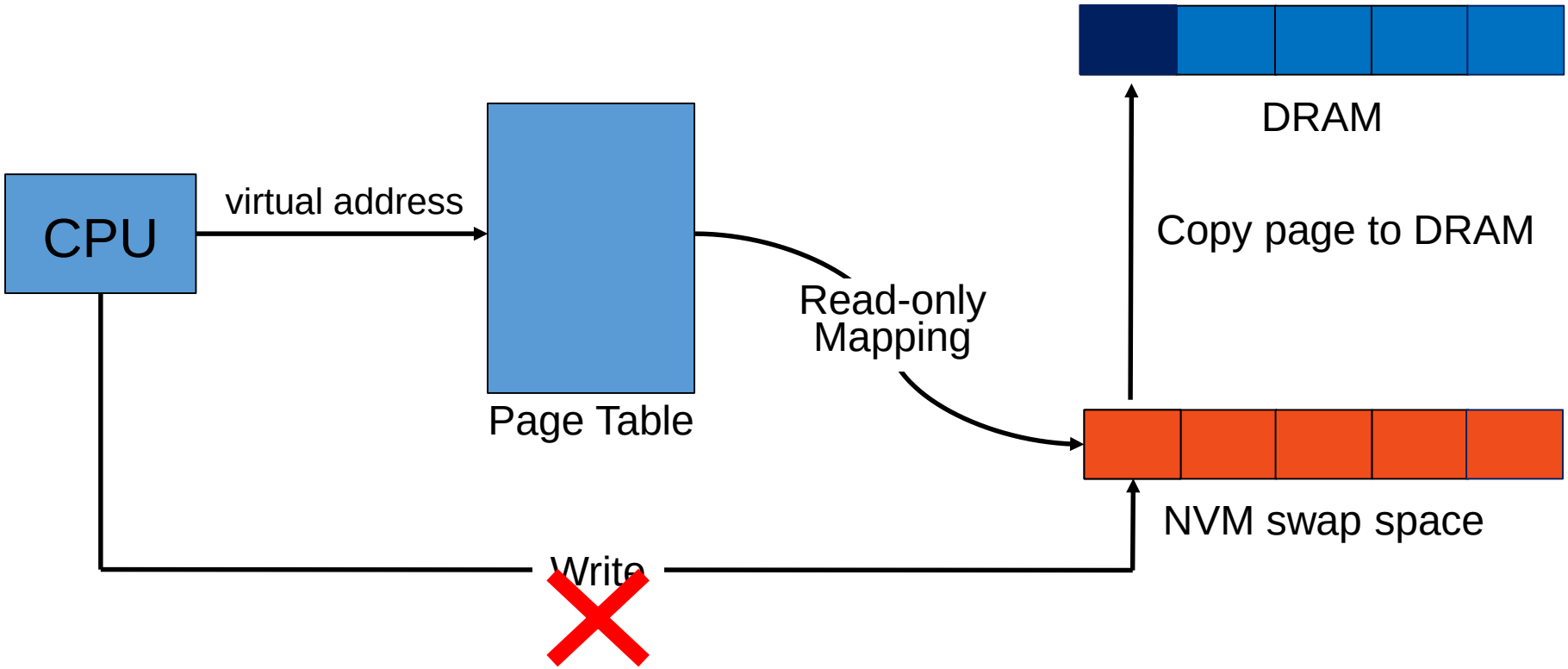
- COWS: remove unnecessary copy operations



When a write fault is triggered and the corresponding PTE's 5th bit is set, we copy the page from NVM swap space back to DRAM

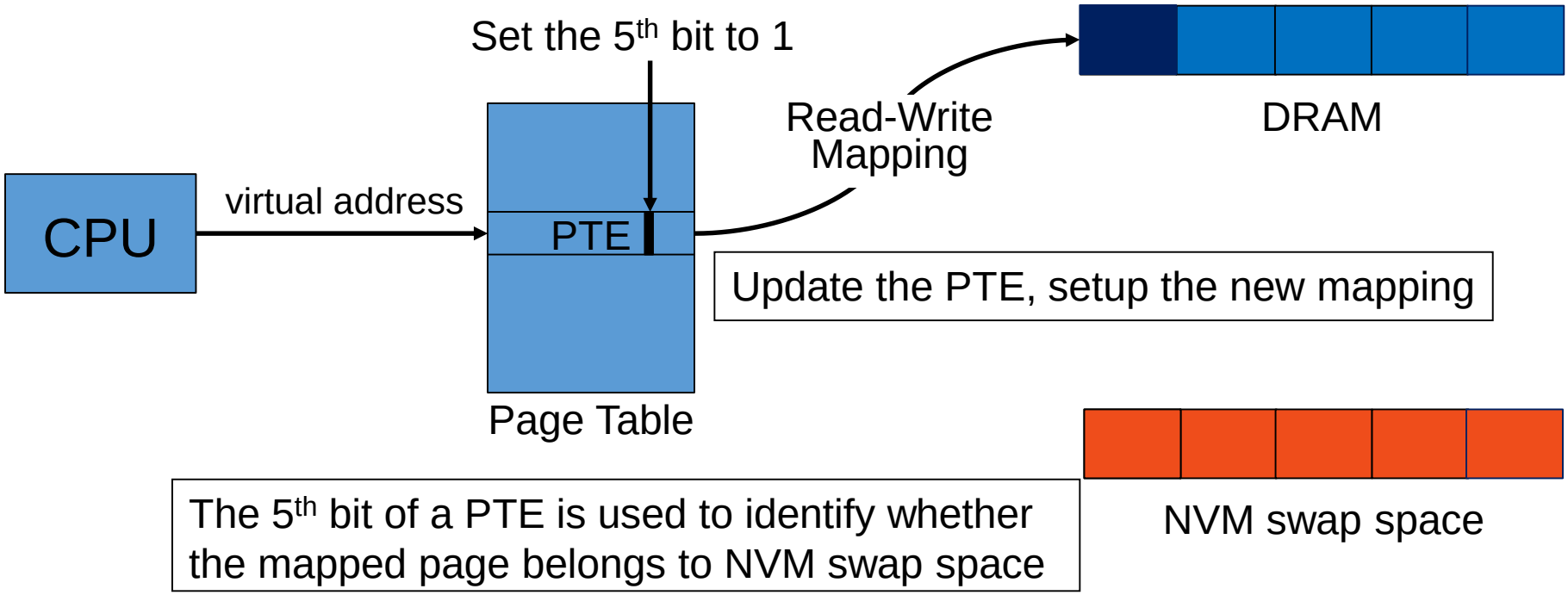
Copy-on-Write Swap-in (COWS)

- COWS: remove unnecessary copy operations



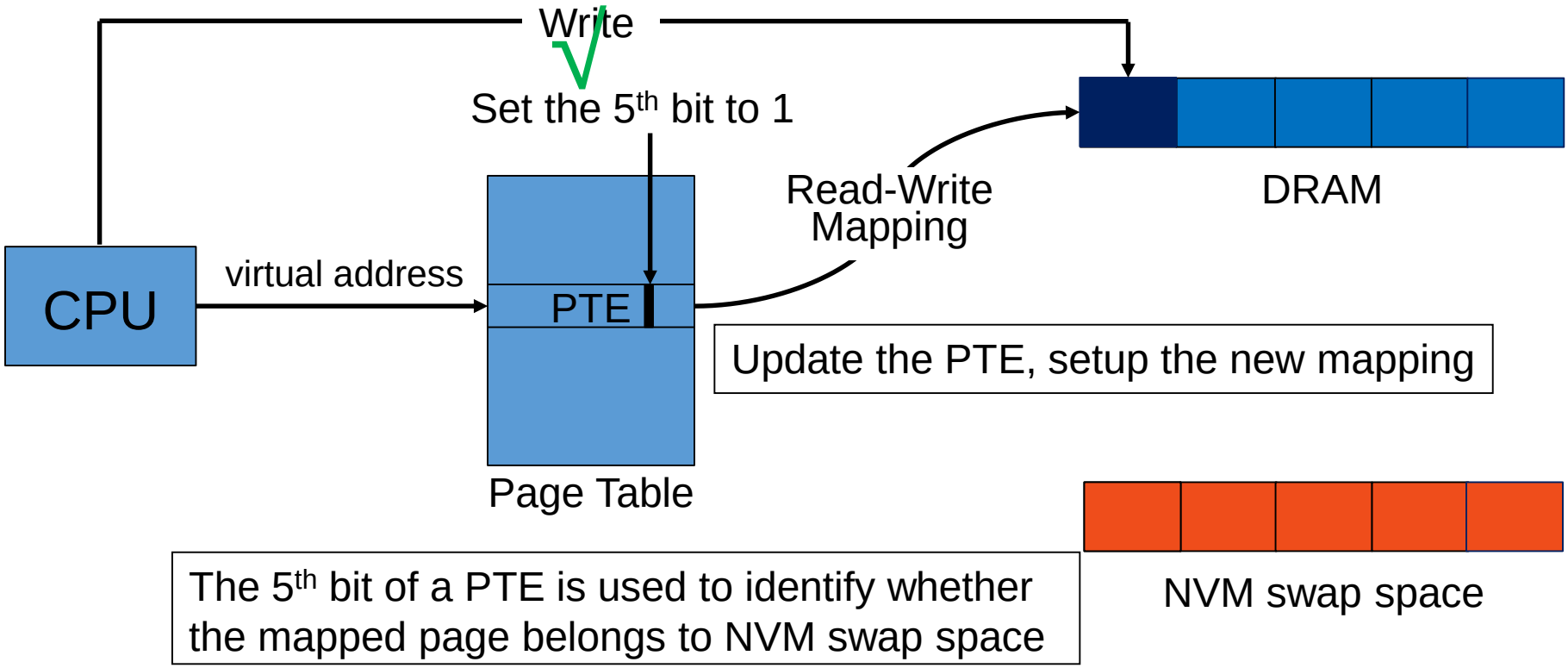
Copy-on-Write Swap-in (COWS)

- COWS: remove unnecessary copy operations



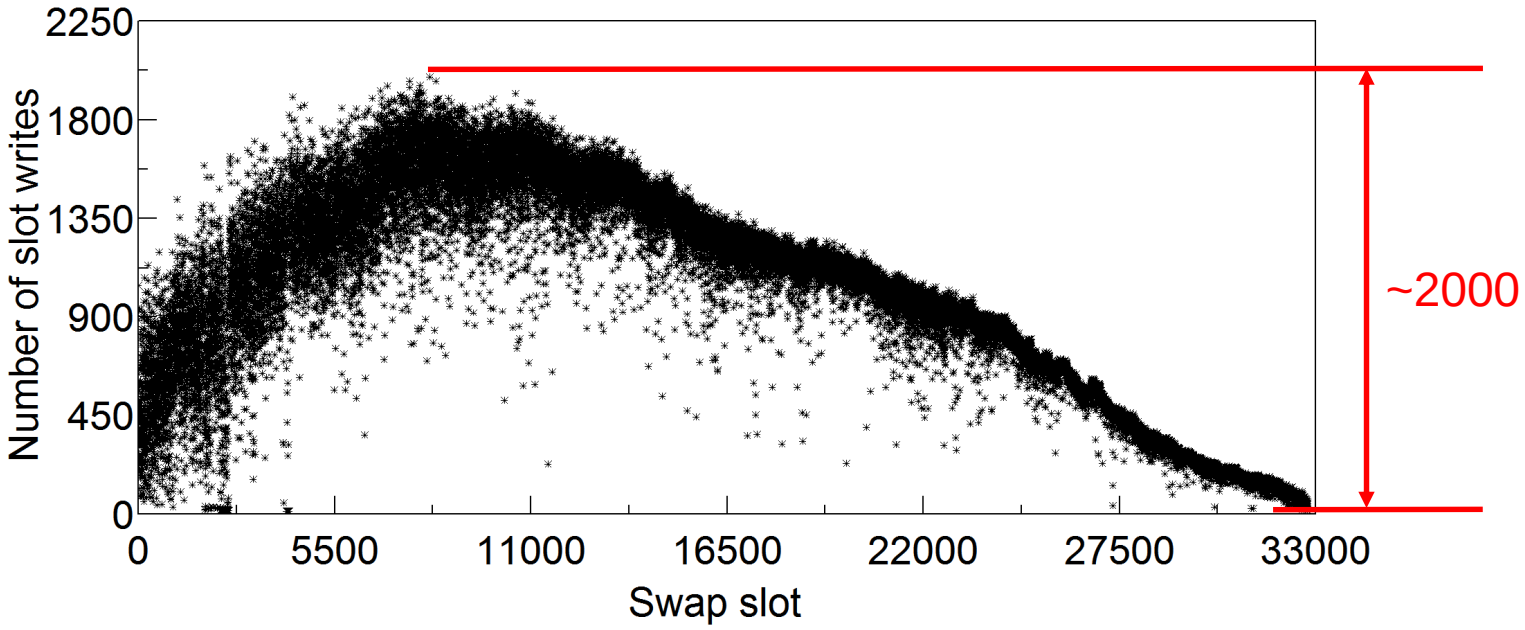
Copy-on-Write Swap-in (COWS)

- COWS: remove unnecessary copy operations



NVM-Swap endurance problem

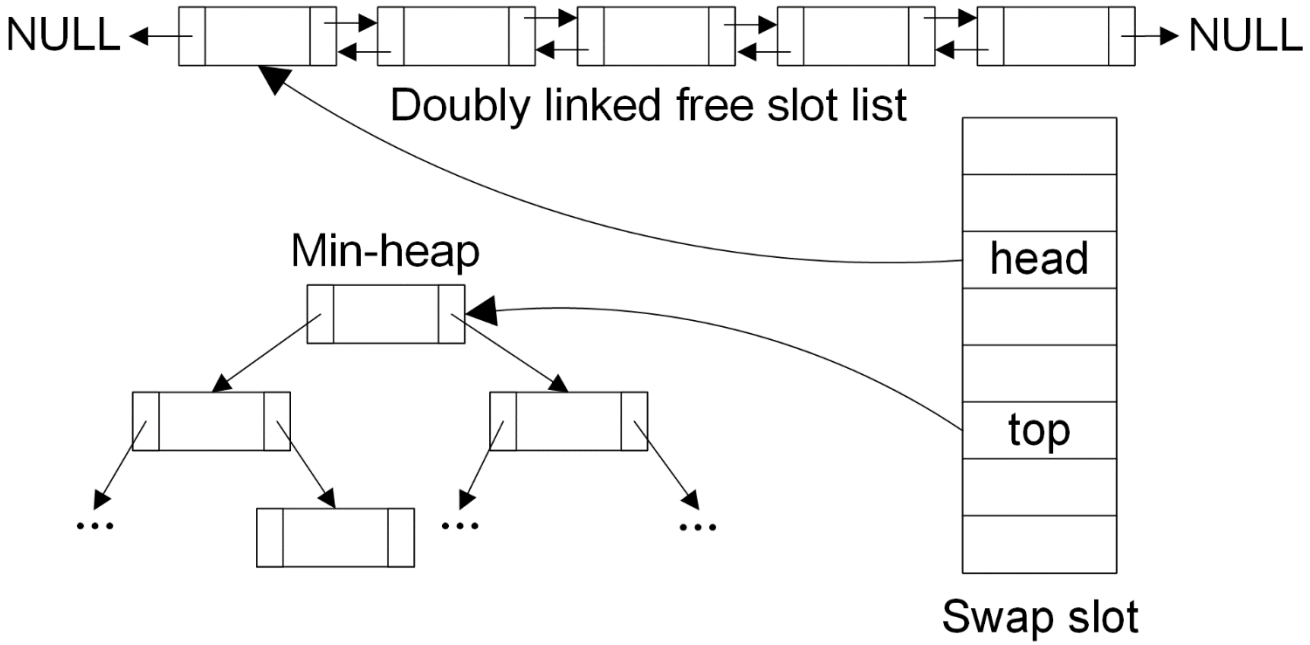
- Unbalance writes in NVM swap area



- Endurance issue of NVM
  - E.g. PCM cell only has 10^8 ~ 10^9 programming cycles
  - Most NVMs are vulnerable to unbalance writes

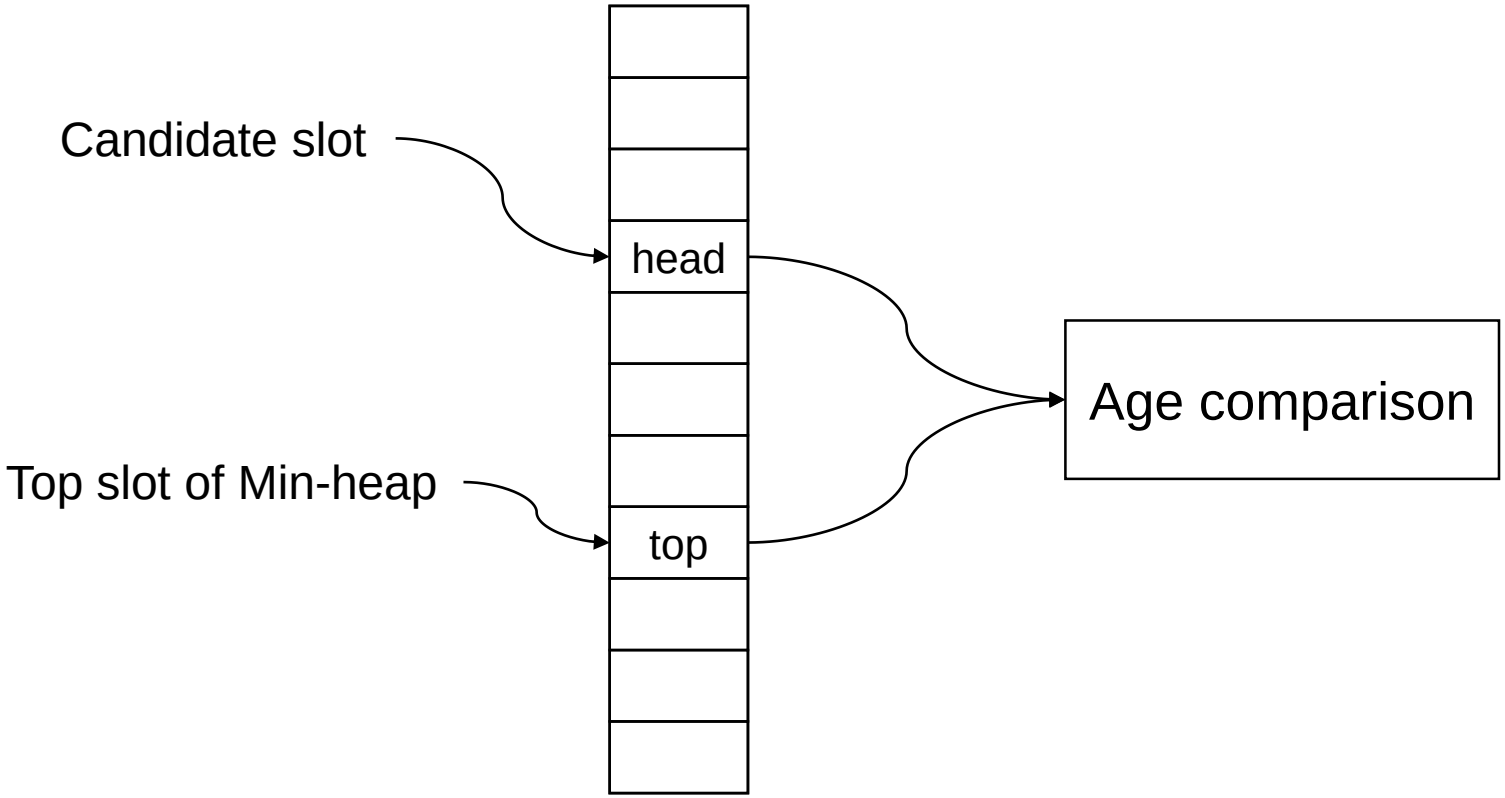
Heap-Wear: NVM-Swap Wear leveling

- Data structure
  - Age counter for each swap slot
  - Doubly linked list
  - Min-heap



Heap-Wear: NVM-Swap Wear leveling

- Age comparison
  - Compare the age of head slot with the top slot

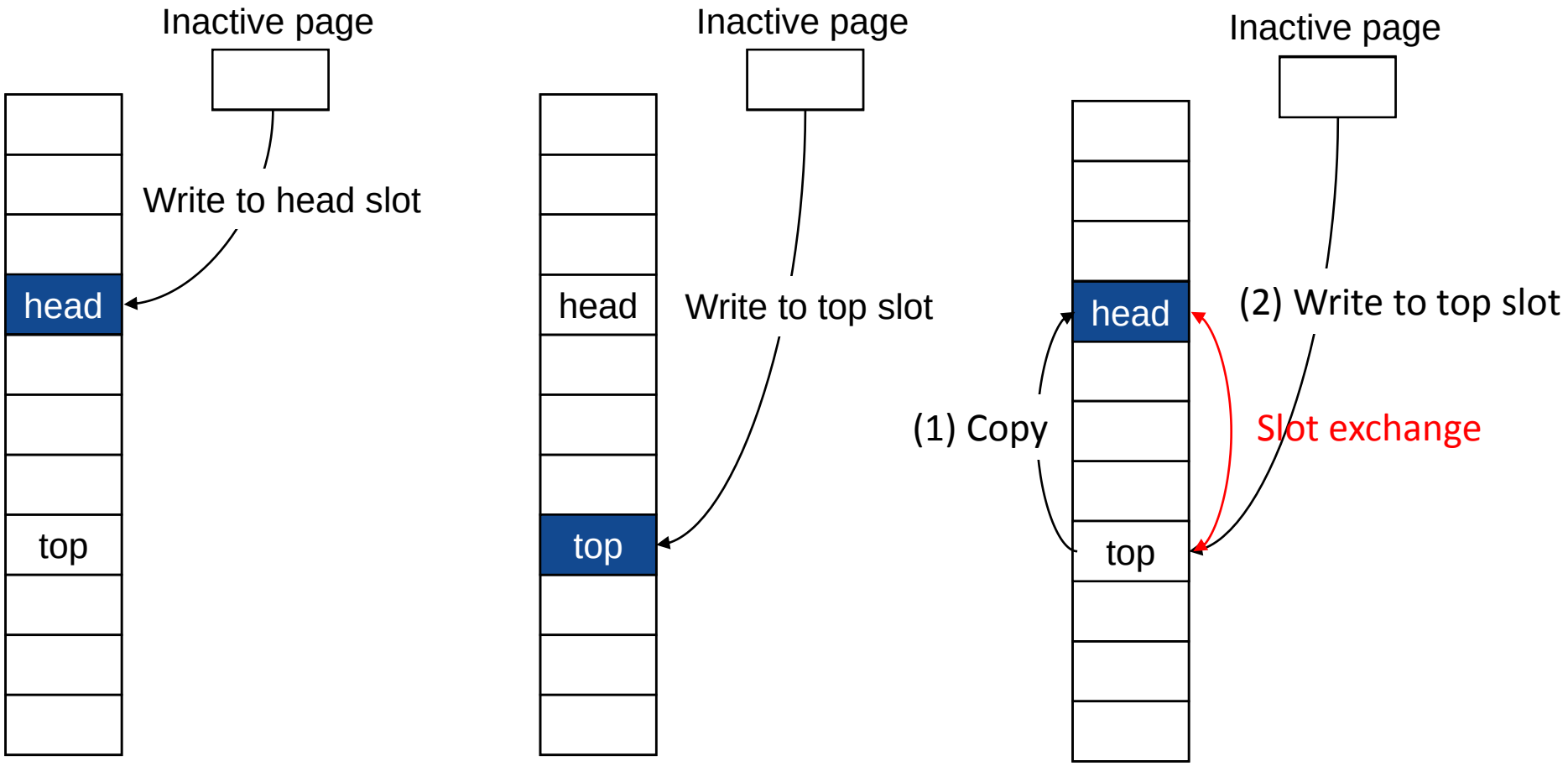


Heap-Wear: NVM-Swap Wear leveling

Case 1: Age difference <= TH

Case 2: Age difference > TH & top slot is free

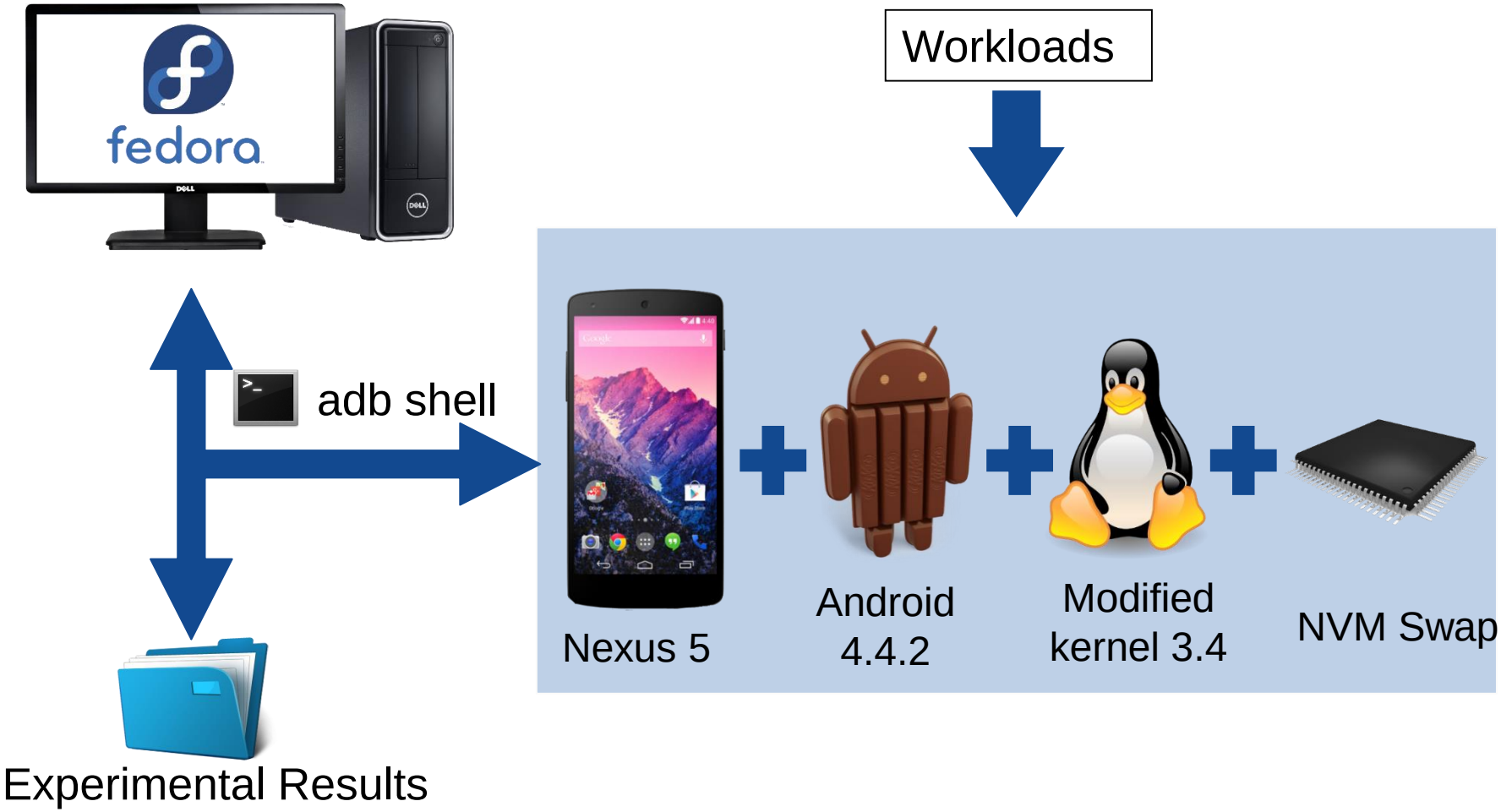
Case 3: Age difference > TH & top slot is in use



Outline

- Background
  - Performance and Energy
  - Swapping
- NVM-Swap: NVM based swapping
  - Copy-on-Write Swap-in (COWS)
  - WL Algorithm: Heap-Wear
- Evaluation
- Conclusion

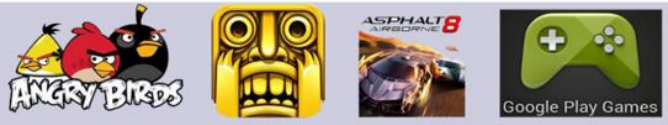
Evaluation — Experimental setup



Evaluation — Metrics

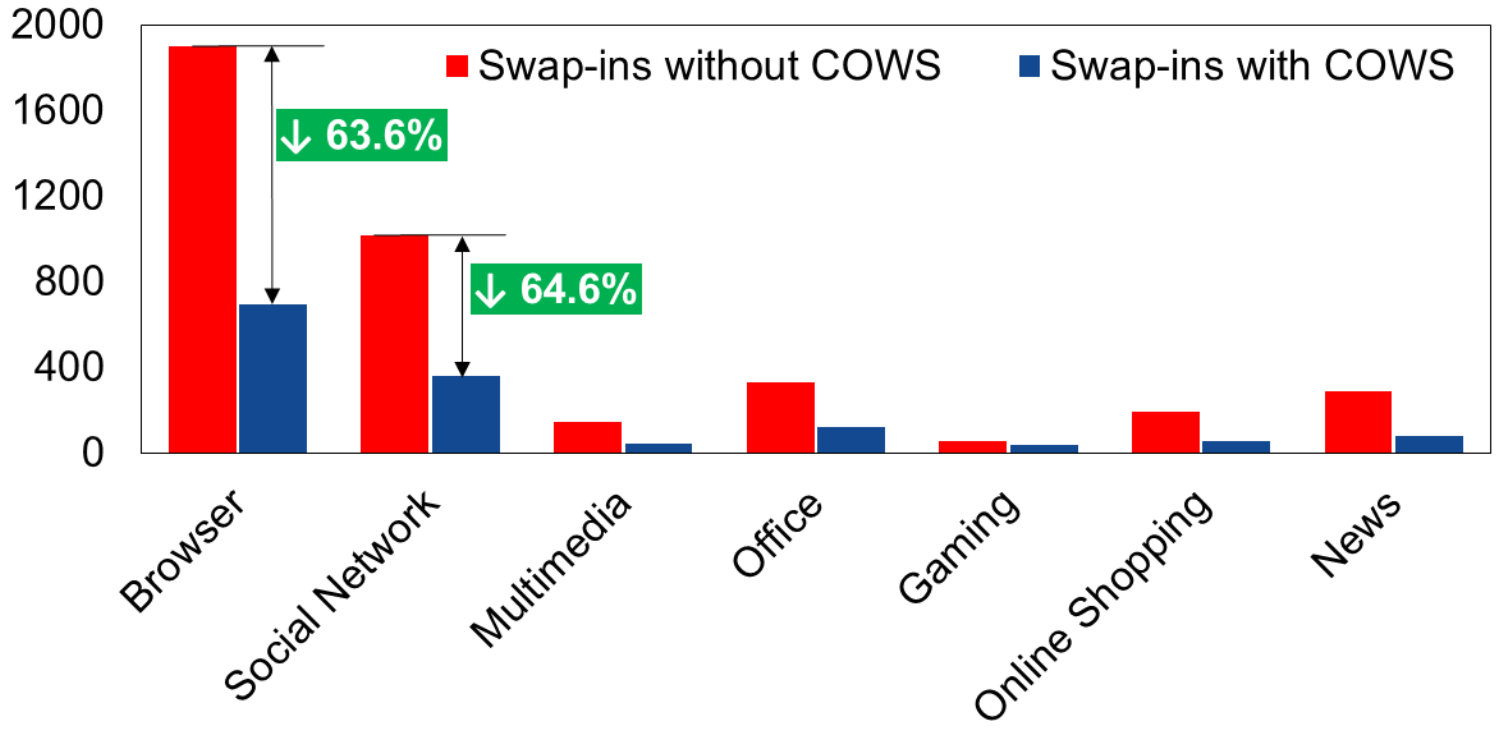
Metrics	Description
Number of memory copy operations	- Measure the effectiveness of COWS - Run various of applications
Wear-leveling	- Evaluate the effectiveness of Heap-Wear - Write 128GB data to swap space in total
Application launch time	- Important performance metric for smartphone users - Use customized applications

Evaluation — Applications

Category	Applications
Browser	
Social network	
Multimedia	
Office	
Gaming	
Shopping	
News	

Evaluation Results

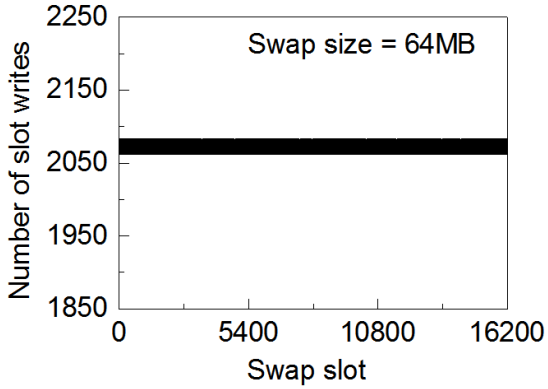
- Memory copy reduction
  - COWS can help reduce around 40% ~ 75% of swap-ins

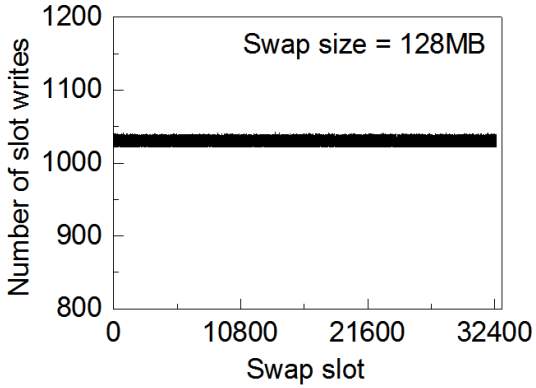


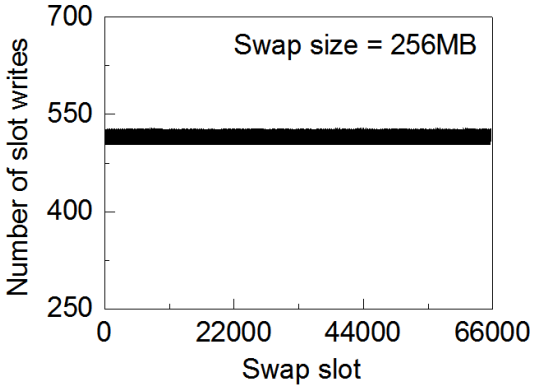
Comparison of Number of memory copy operations

Evaluation Results

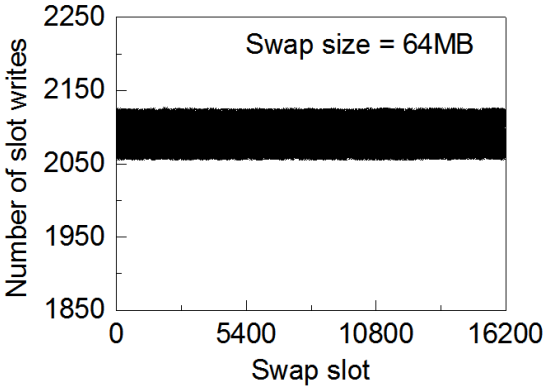
- Wear-leveling

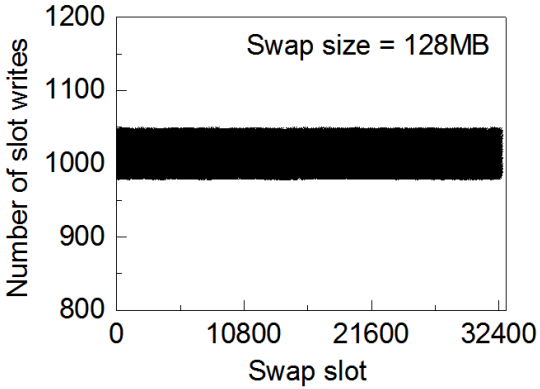


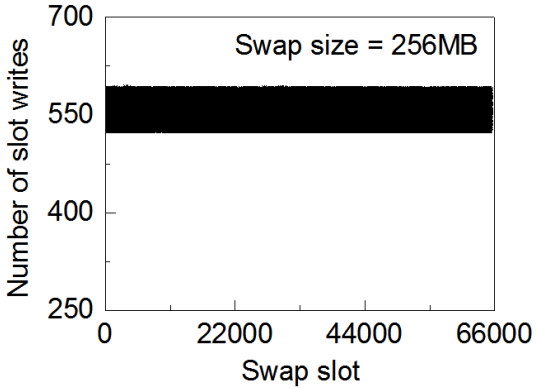




Threshold = 16



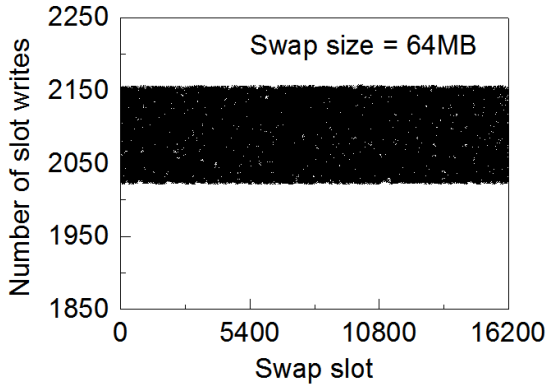


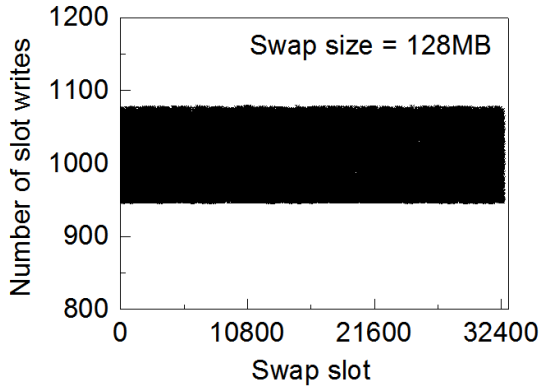


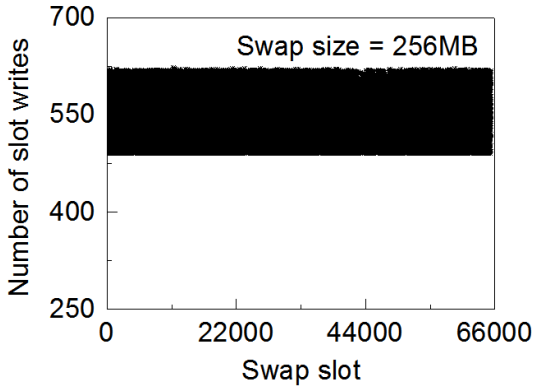
Threshold = 64

Evaluation Results

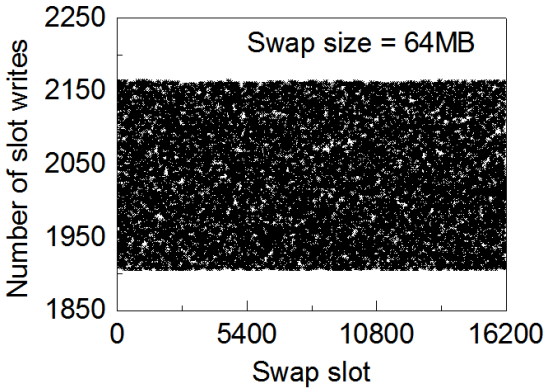
- Wear-leveling

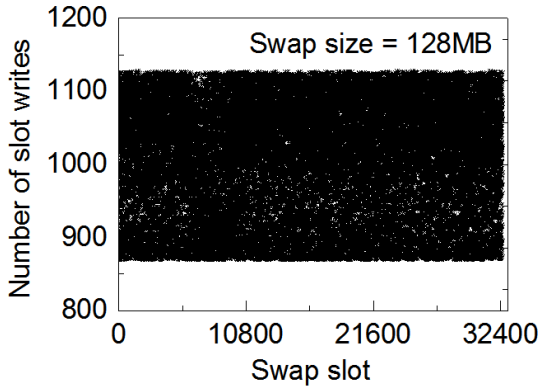


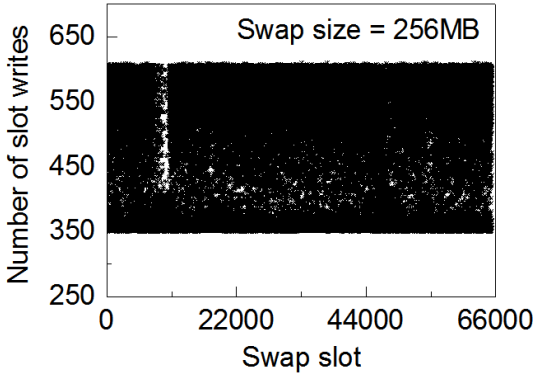




Threshold = 128



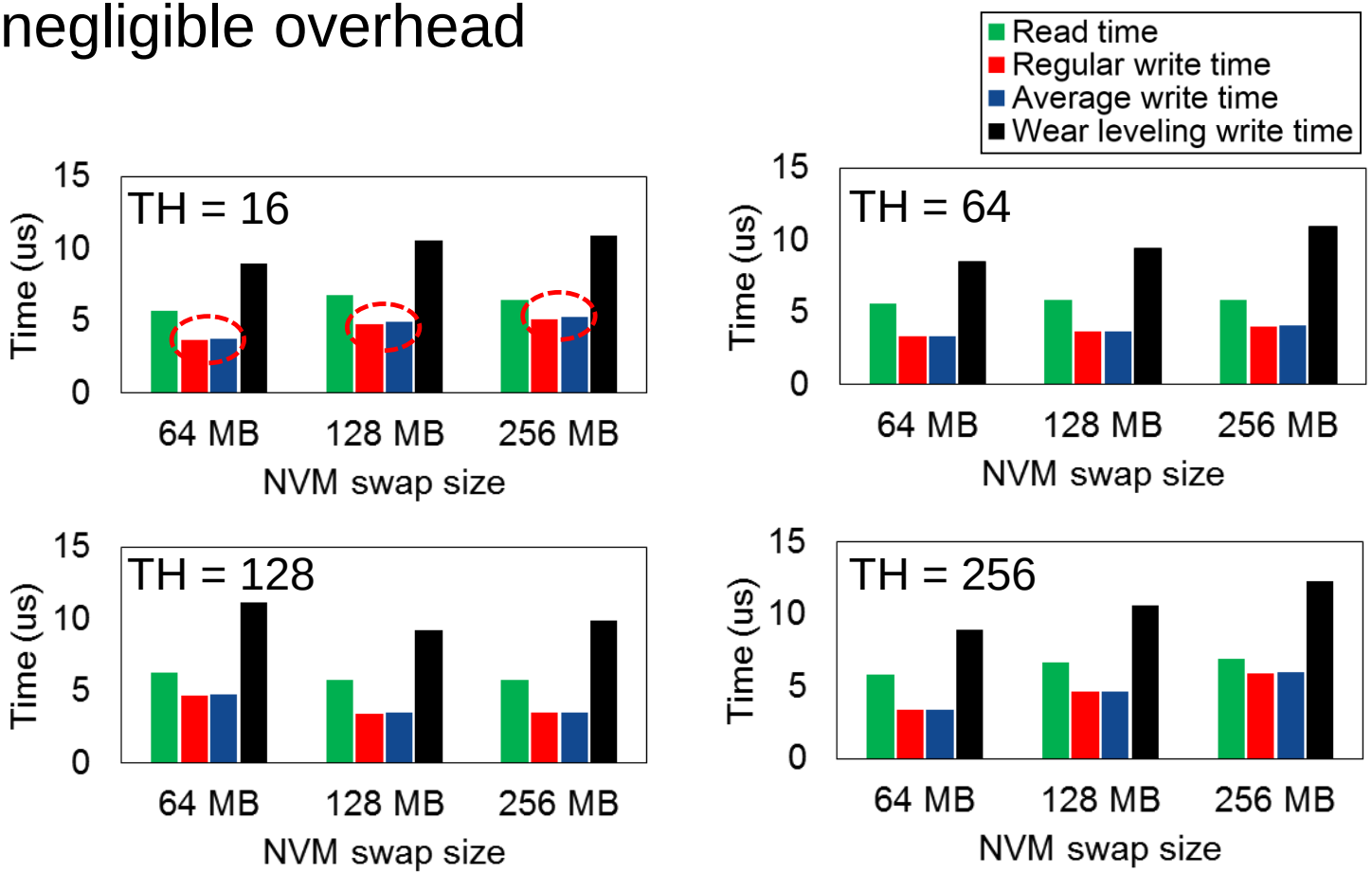




Threshold = 256

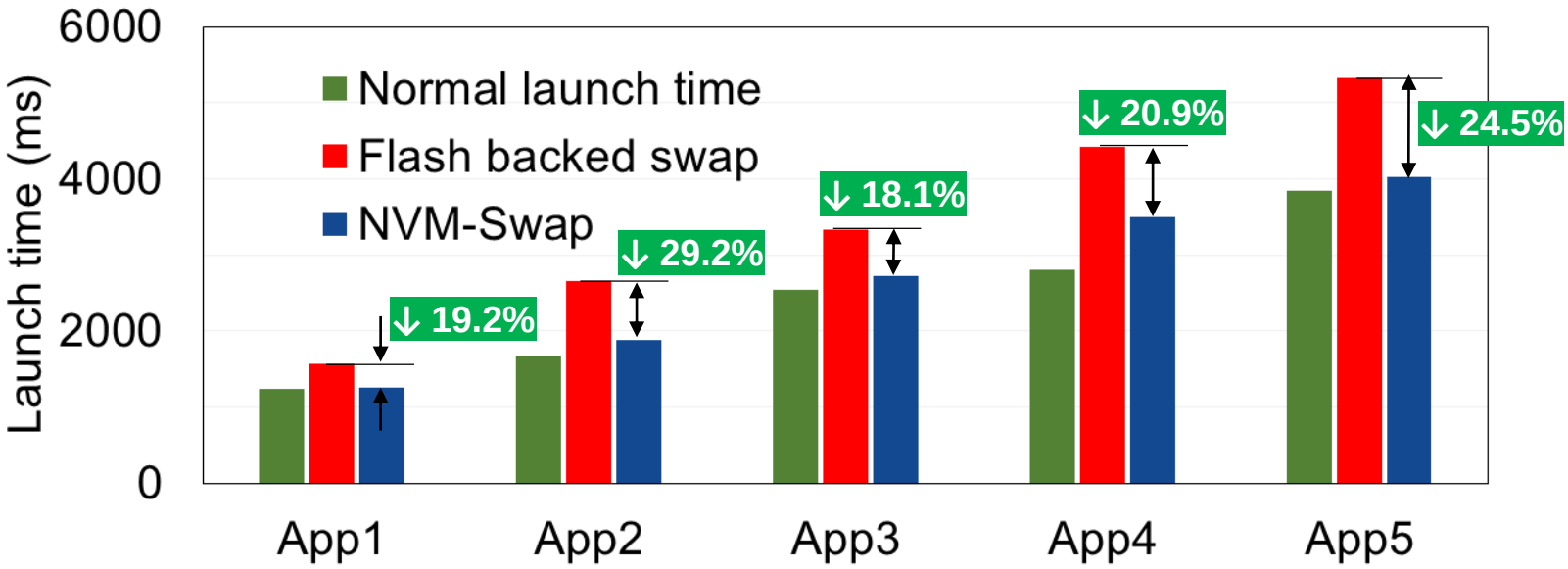
Evaluation Results

- Wear-leveling overhead
  - Distribute the writes evenly across the swap space with negligible overhead



Evaluation Results

- Application launch time
  - NVM-Swap is more than 20% faster on average compared to Flash backed swap



App1-5 simulate application launch by loading a file of size 10MB to 30MB (5MB increment)

Conclusion

- We revisited swapping in smartphones and proposed NVM-Swap to build high-performance smartphones
- COWS: Remove unnecessary memory copy operations
  - More than 50% memory copy operations reduction
- Heap-Wear: WL algorithm for NVM swap space
  - Distribute writes evenly across whole swap space with negligible overhead
- Improve user experience
  - Compared to flash-based swap, application launch time is reduced more than 20% with the help of NVM-Swap

Thank you!

Question?

http://nvm-swap.bitbucket.org/