

# FADaC: A Self-Adapting Data Classifier For Flash Memory

Kevin Kremer, André Brinkmann  
Johannes Gutenberg University Mainz

SYSTOR 2019

JOHANNES GUTENBERG  
UNIVERSITÄT MAINZ



# Data classification

<b>LPN</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	...
<b># accesses</b>	5	42	1	0	22	23	111	121	...



# SSD components

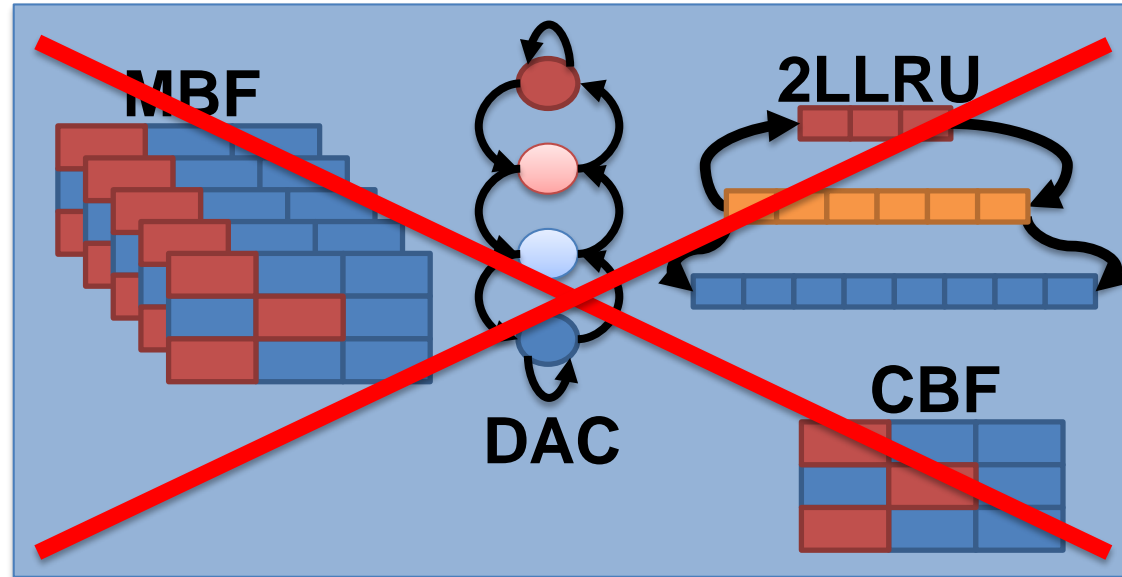
---

Controlling unit

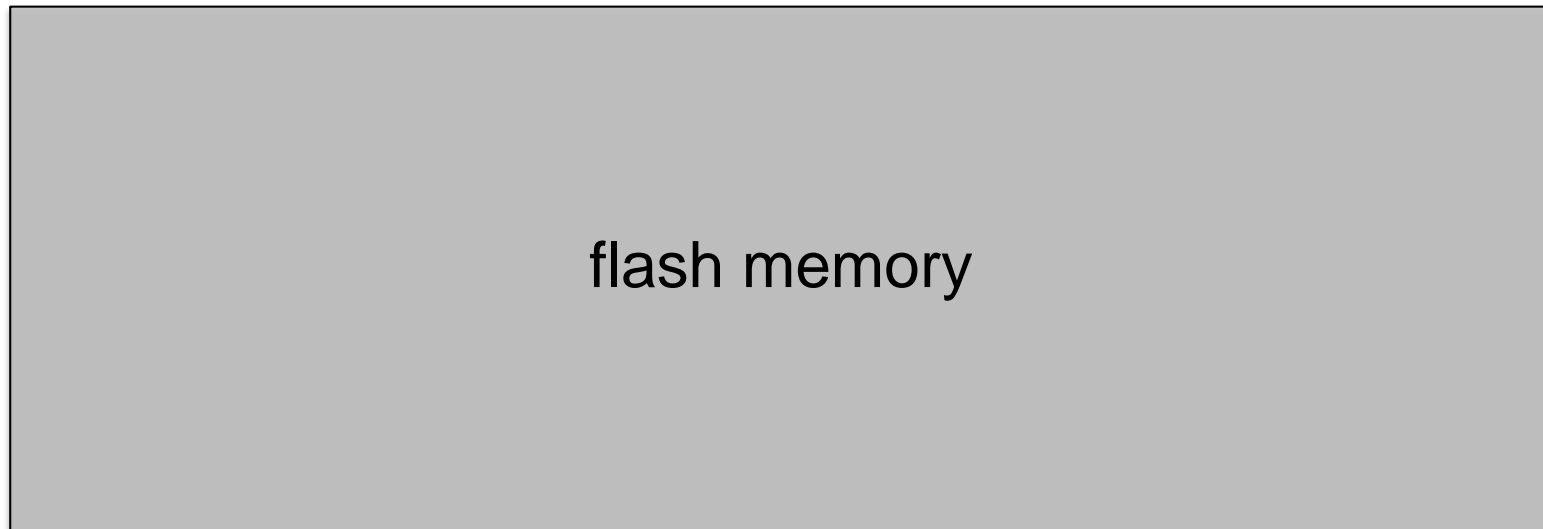


# SSD components

Controlling unit



Storage unit



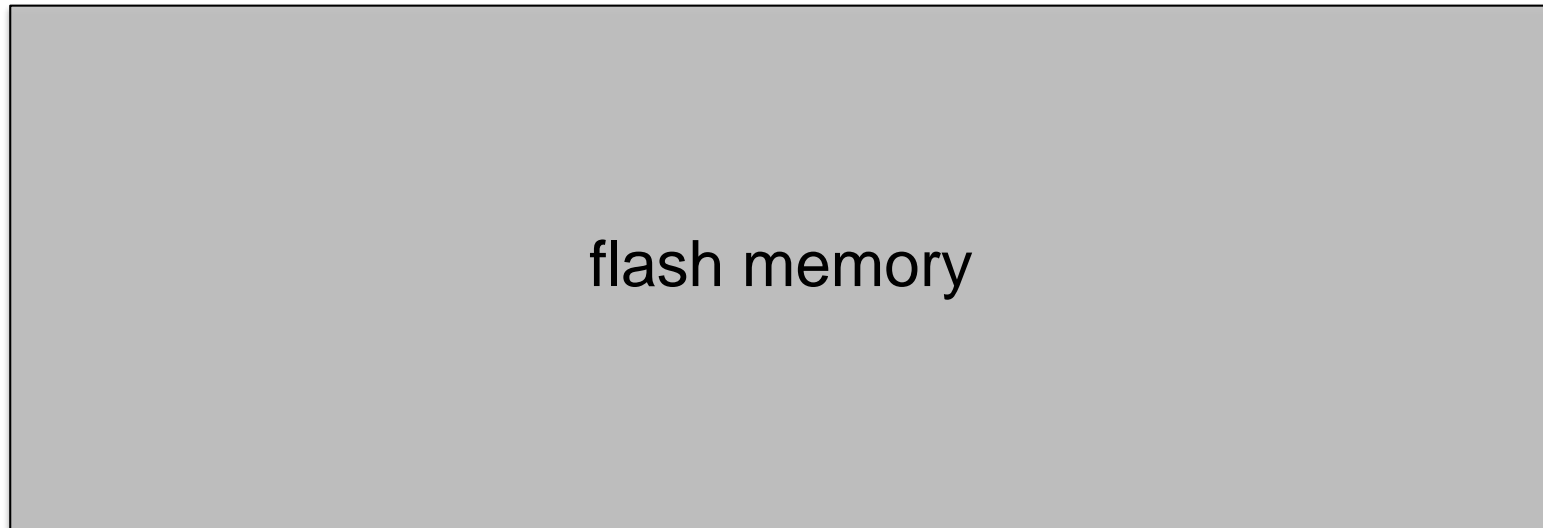
# SSD components

Controlling unit



- exact values
- improve classification quality
- independent from DRAM

Storage unit

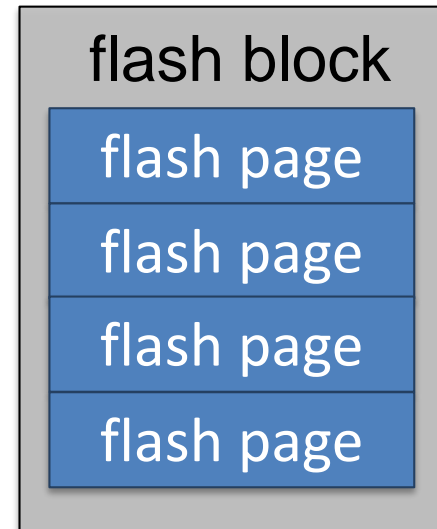
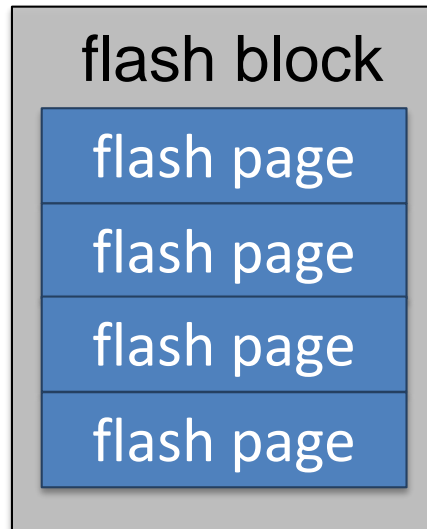
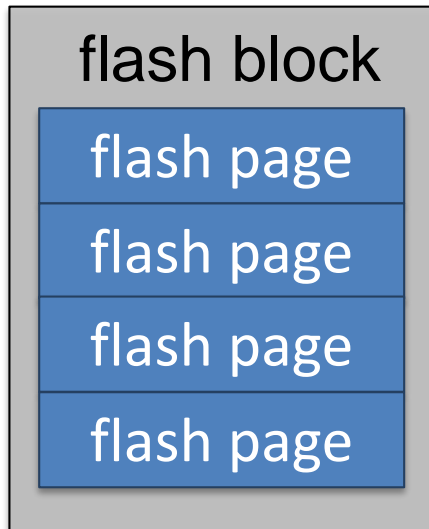


# SSD components

Controlling unit

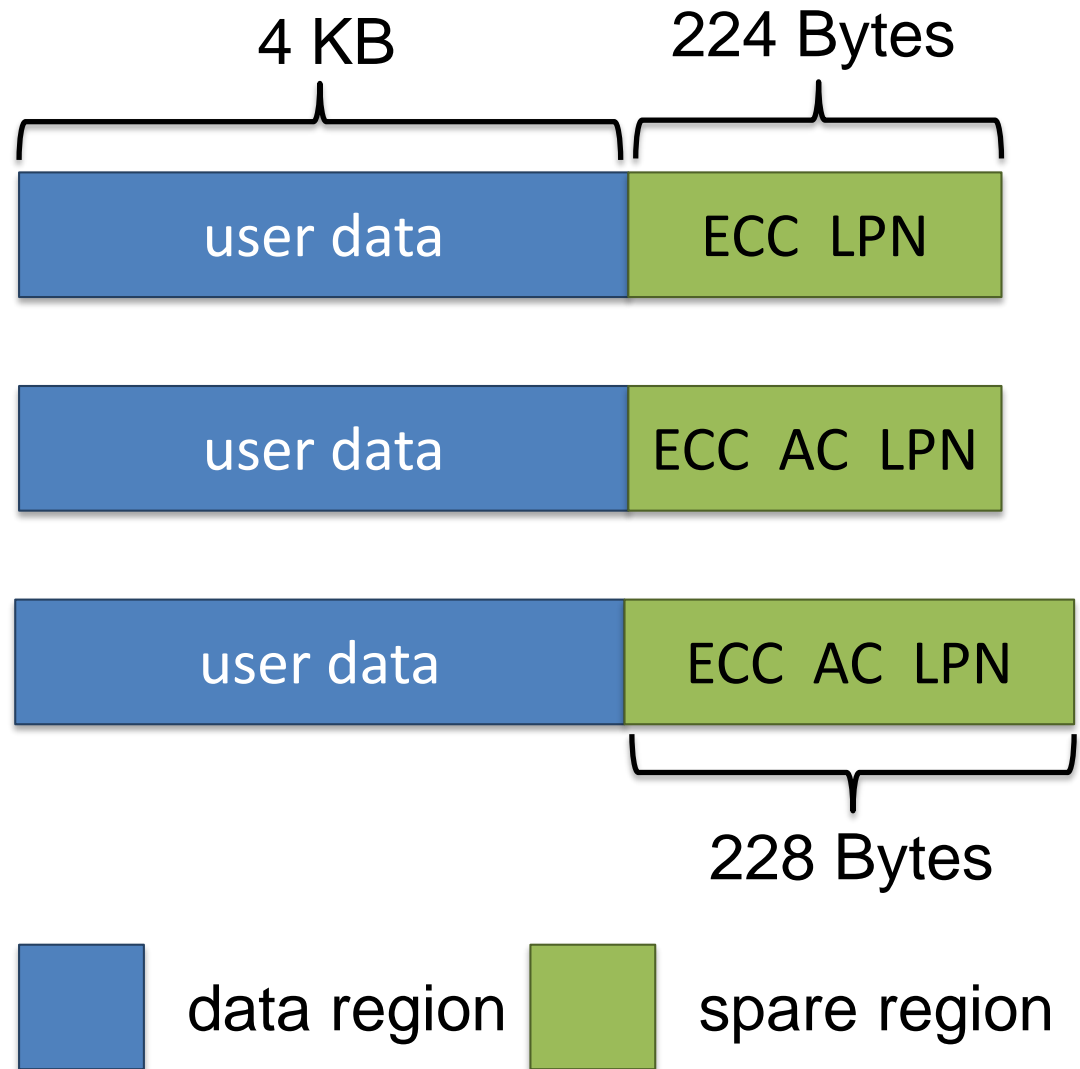


Storage unit

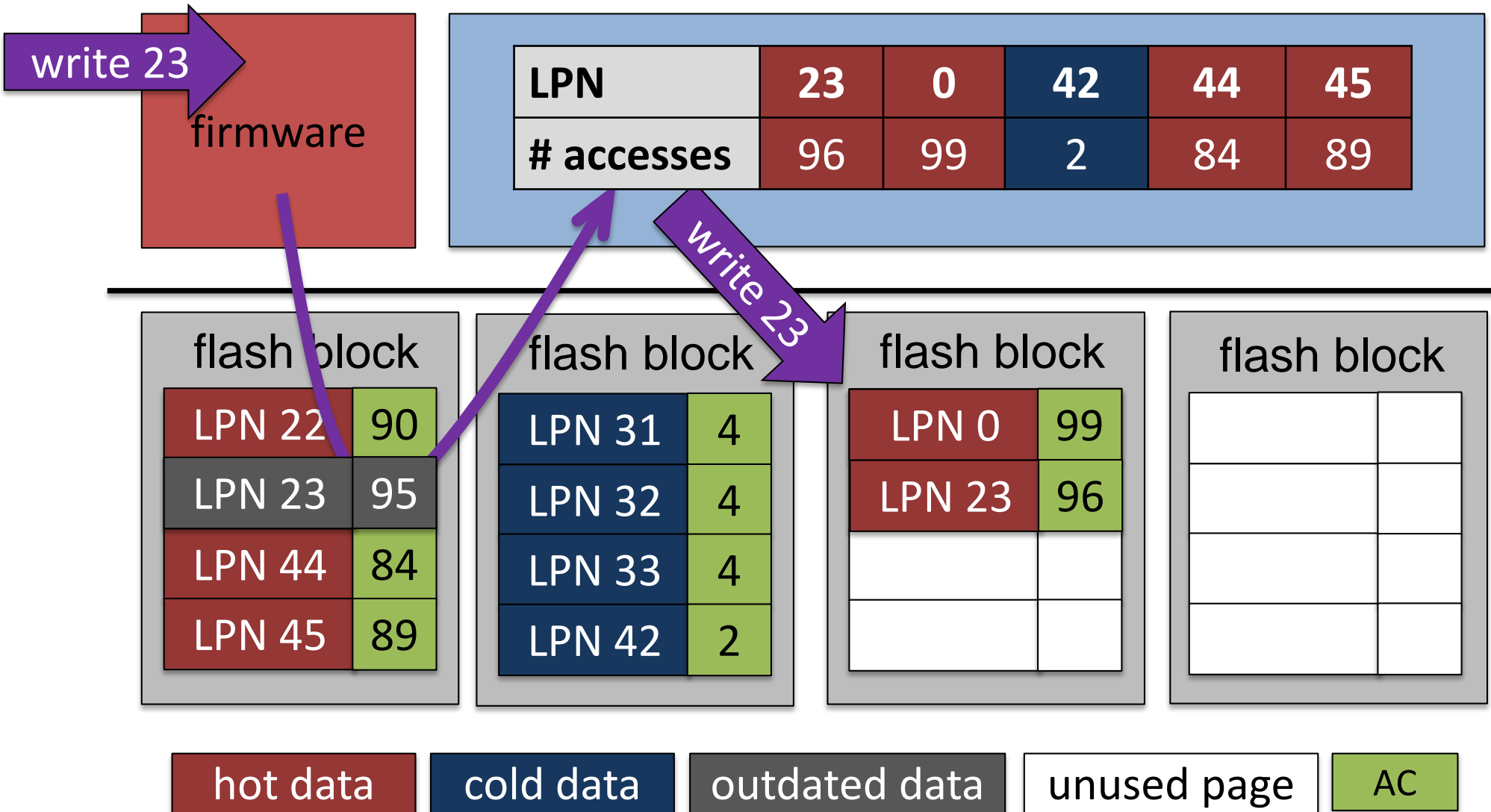


# Flash layout

- user data
- spare region for meta data
- reduce ECC
- enlarge spare region



# SSD components





# SSD components

firmware

LPN	45	44	23	0	42
# accesses	90	85	96	99	2

flash block

LPN 22	90
LPN 23	95
LPN 44	84
LPN 45	89

flash block

LPN 31	4
LPN 32	4
LPN 33	4
LPN 42	2

flash block

LPN 0	99
LPN 23	96
LPN 44	85
LPN 45	90

flash block


hot data

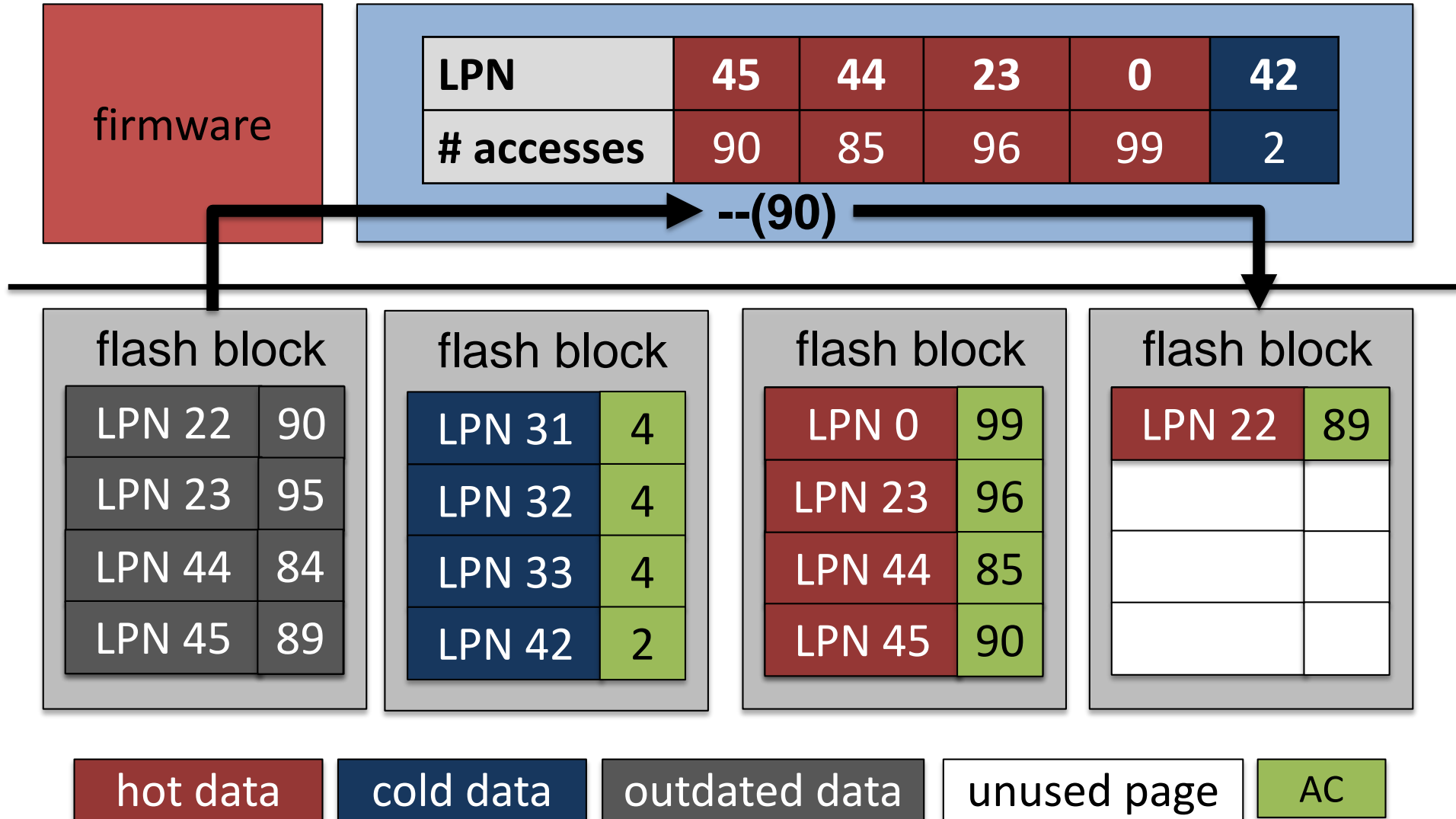
cold data

outdated data

unused page

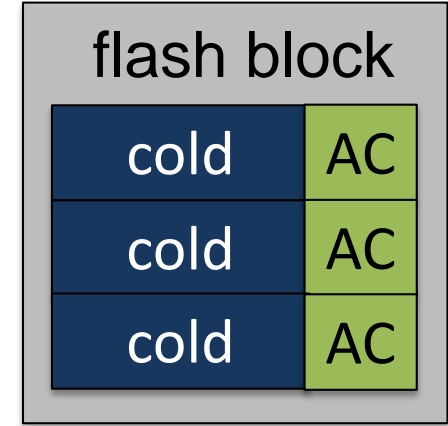
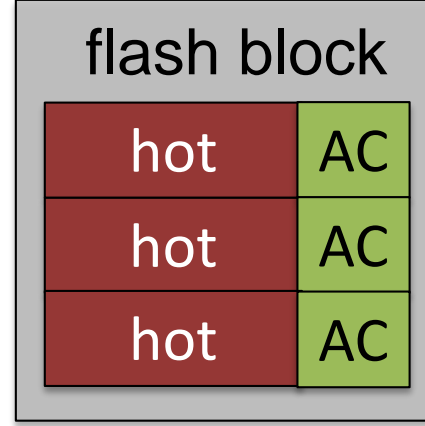
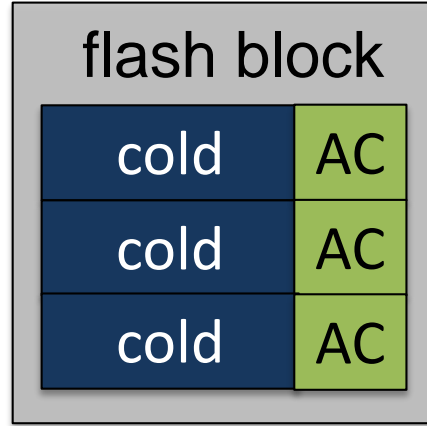
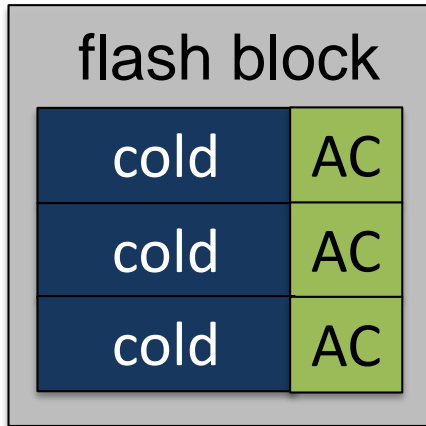
AC

# SSD components

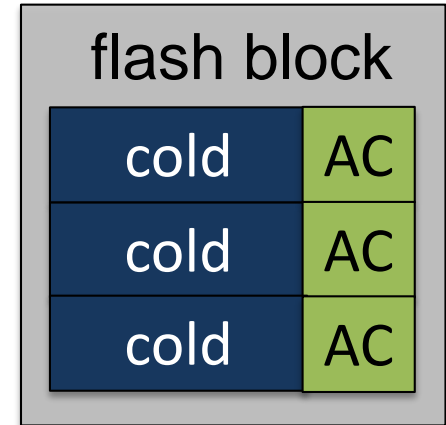
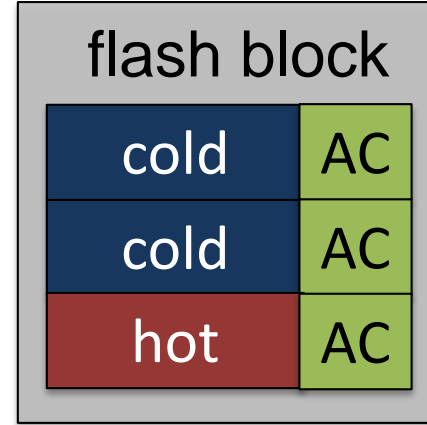
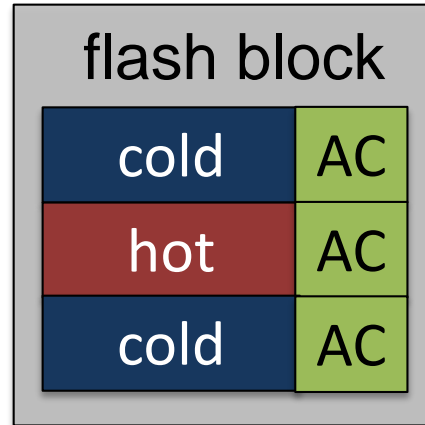
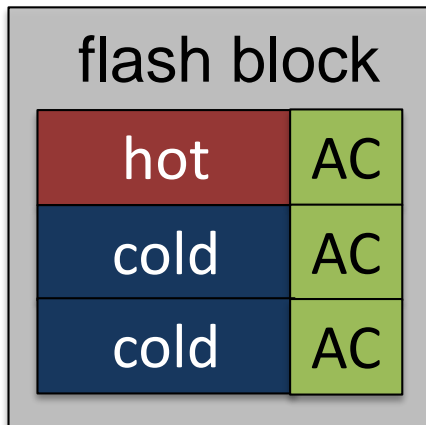


# SSD components

with separation

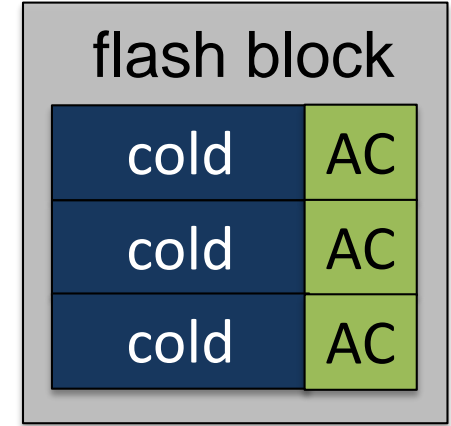
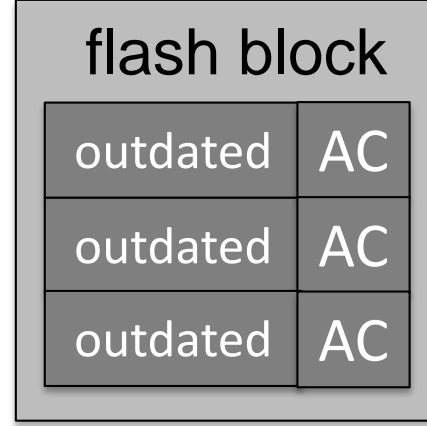
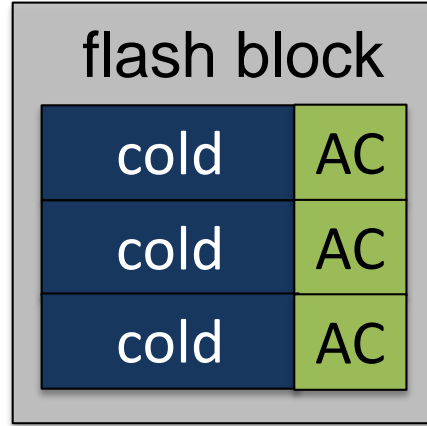
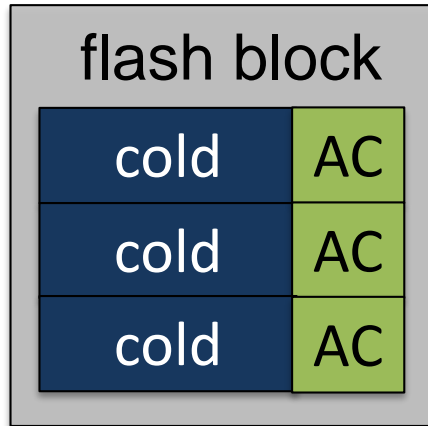


no separation

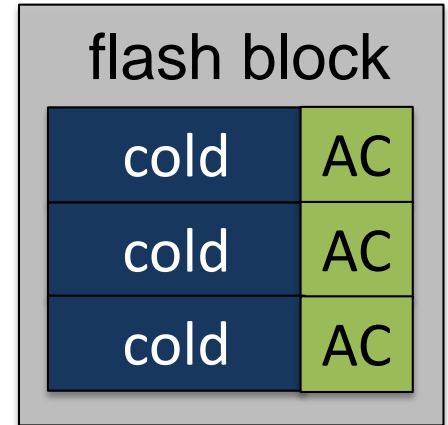
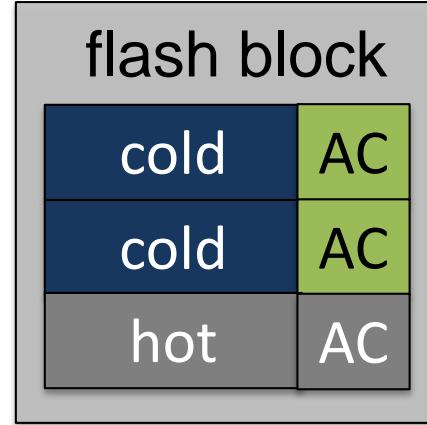
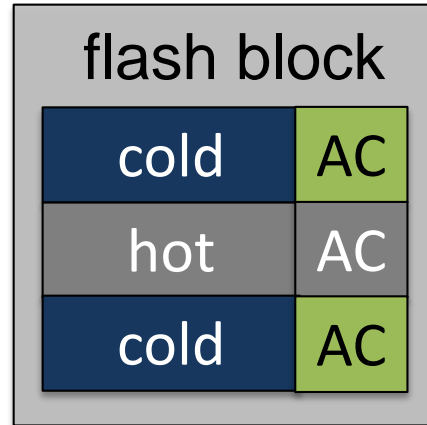
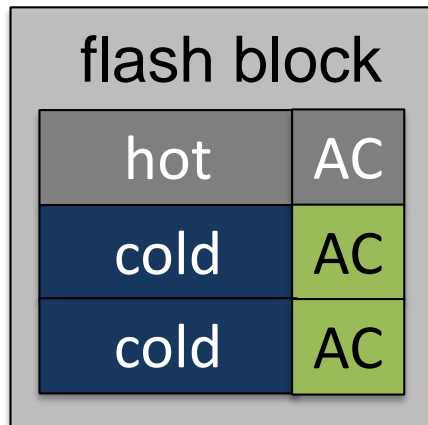


# SSD components

with separation

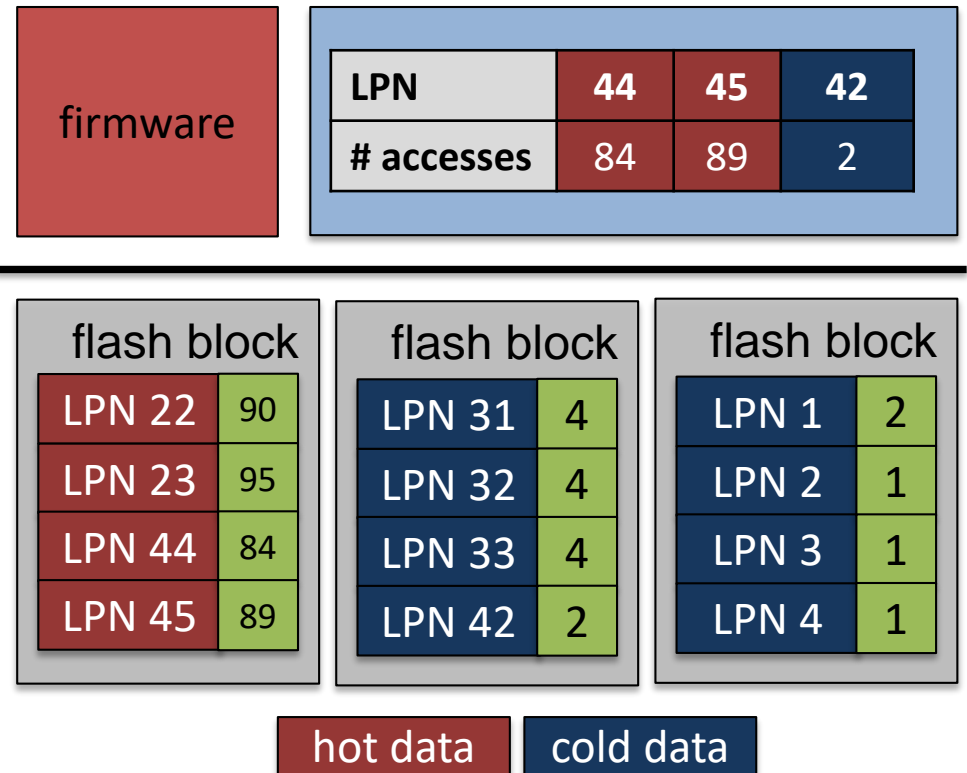


no separation



# Flash layout

- store access counters in a page's spare region
- increase on write
  - on cache miss, needs a fetch first
- decrease on GC
  - no additional cost
- use DRAM as a cache only
- no additional writes



# Classification threshold

flash block

hot	90
hot	95
hot	89

flash block

cold	20
cold	20
cold	20

flash block

hot	30
hot	30
hot	30

flash block

hot	30
hot	33
hot	32

$$avg_t = avg_{t-1} + \frac{(request\ size_t)}{\#\ pages} = 21.63$$

flash block

	0
	0
	0

flash block

	0
	0
	0

flash block

	0
	0
	0

flash block

	0
	0
	0

# Classification threshold

flash block

static	90
static	95
static	89

flash block

static	20
static	20
static	20

flash block

static	30
static	30
static	30

flash block

static	30
static	33
static	32

$$avg_t = avg_{t-1} + \frac{(request\ size_t)}{\#\ pages} = 25.25$$

flash block

cold	1
cold	1
cold	1

flash block

cold	2
cold	2
cold	2

flash block

cold	1
cold	1
cold	1

flash block

cold	25
cold	25
cold	25

# Classification threshold

flash block

hot	90
hot	95
hot	89

flash block

cold	20
cold	20
cold	20

flash block

cold	30
cold	30
cold	30

flash block

cold	30
cold	33
cold	32

$$favg_t = favg_{t-1} + \frac{(request\ size_t)}{\#\ used\ pages} = 43.25$$

flash block

	0
	0
	0

flash block

	0
	0
	0

flash block

	0
	0
	0

flash block

	0
	0
	0



# Classification threshold

flash block

static	90
static	95
static	89

flash block

static	20
static	20
static	20

flash block

static	30
static	30
static	30

flash block

static	30
static	33
static	32

$$favg_t = favg_{t-1} + \frac{(request\ size_t) - favg_{t-1}}{\#\ used\ pages} = 24.3$$

flash block

cold	1
cold	1
cold	1

flash block

cold	2
cold	2
cold	2

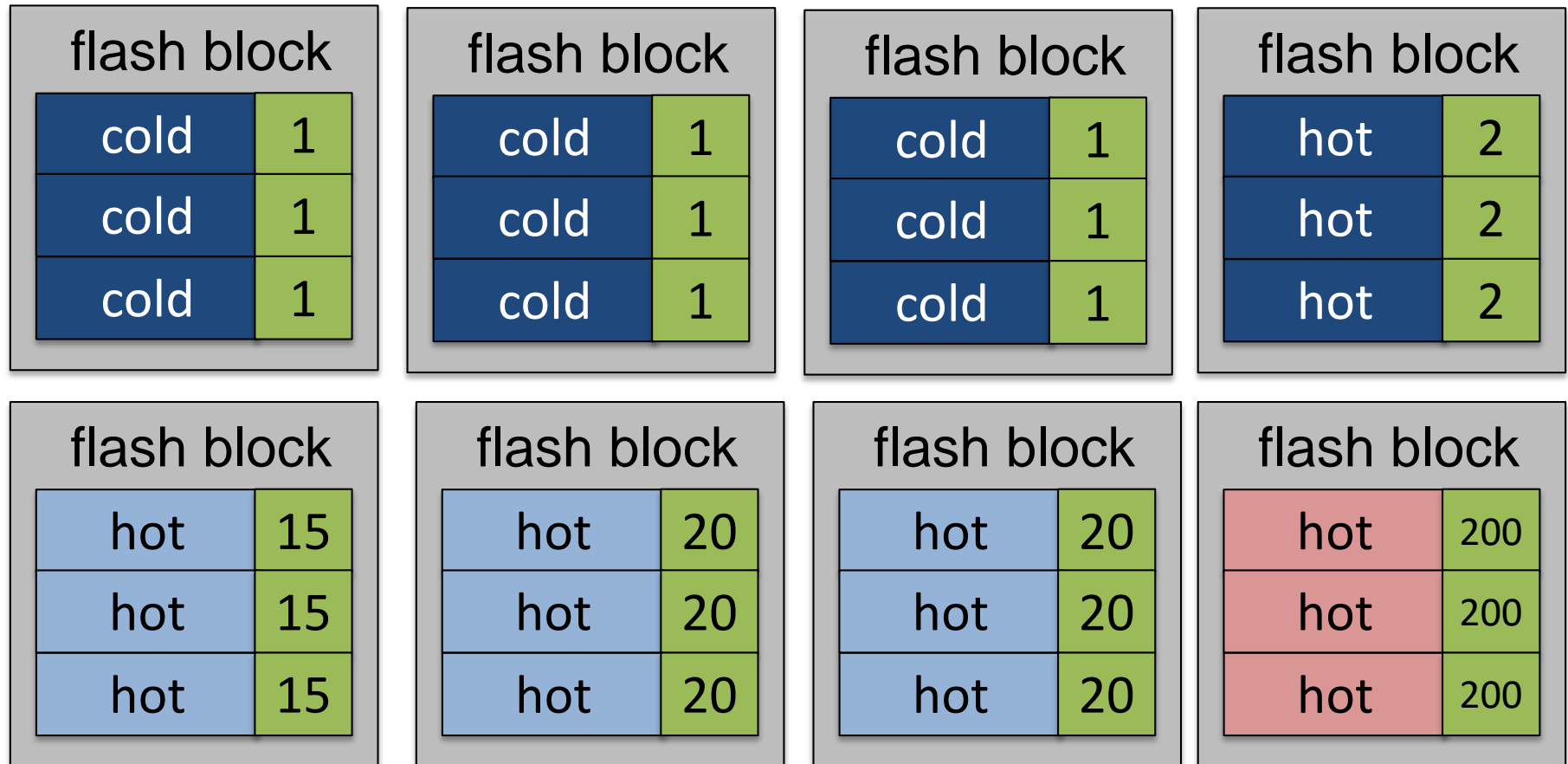
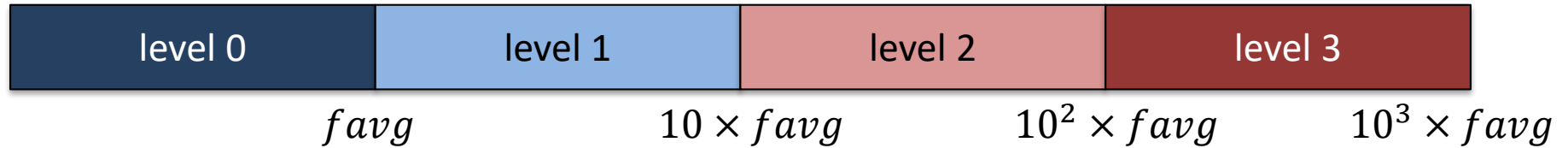
flash block

cold	1
cold	1
cold	1

flash block

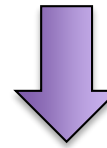
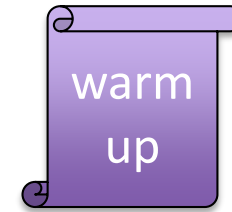
hot	25
hot	25
hot	25

# Classification threshold



# Evaluation

- 950 GB flash memory  
326 MB DRAM
- 7% overprovisioning
- warm up phase



LPN	44	45	42
# accesses	84	89	2

flash block	
LPN 22	90
LPN 23	95
LPN 44	84
LPN 45	89

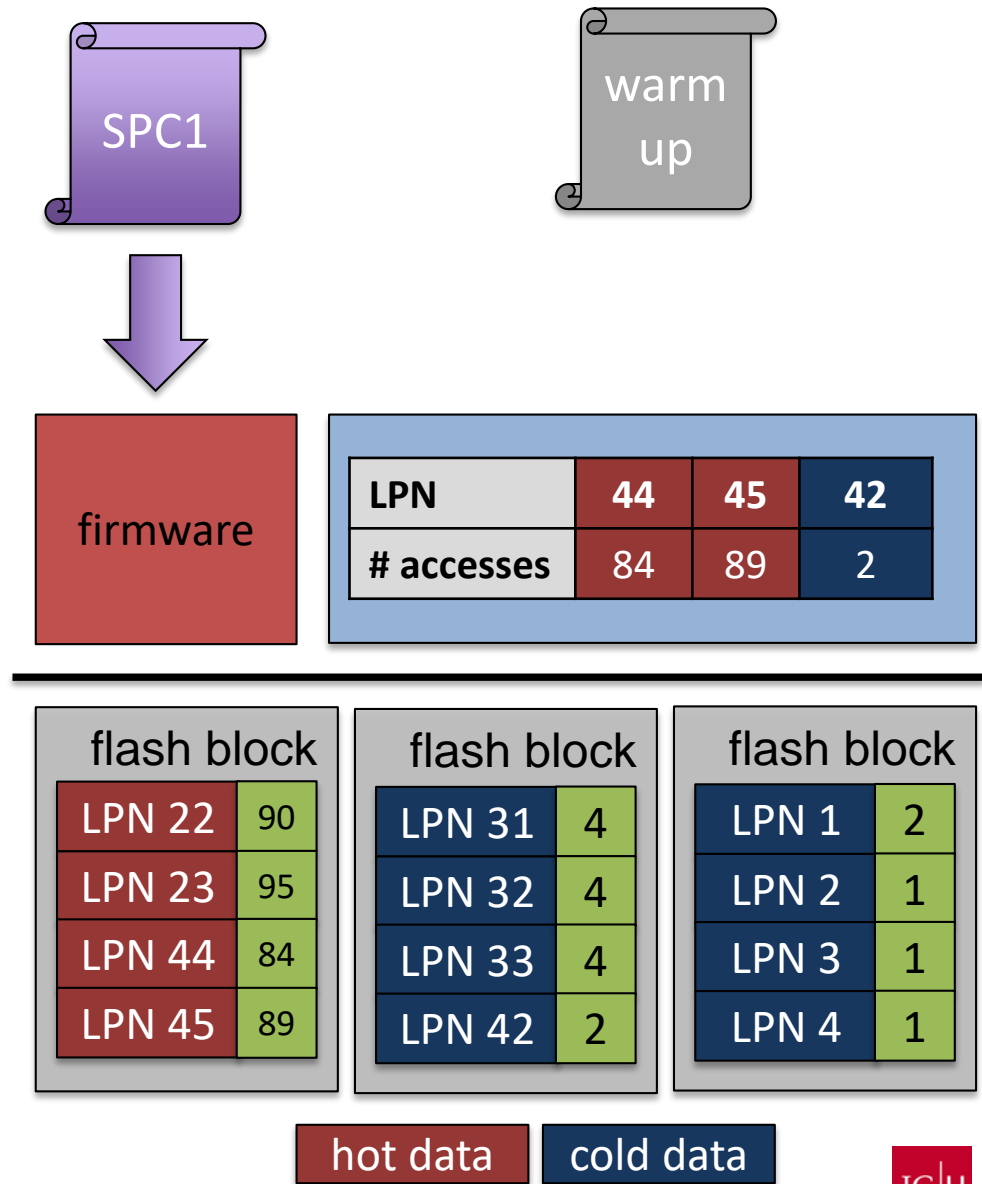
flash block	
LPN 31	4
LPN 32	4
LPN 33	4
LPN 42	2

flash block	
LPN 1	2
LPN 2	1
LPN 3	1
LPN 4	1



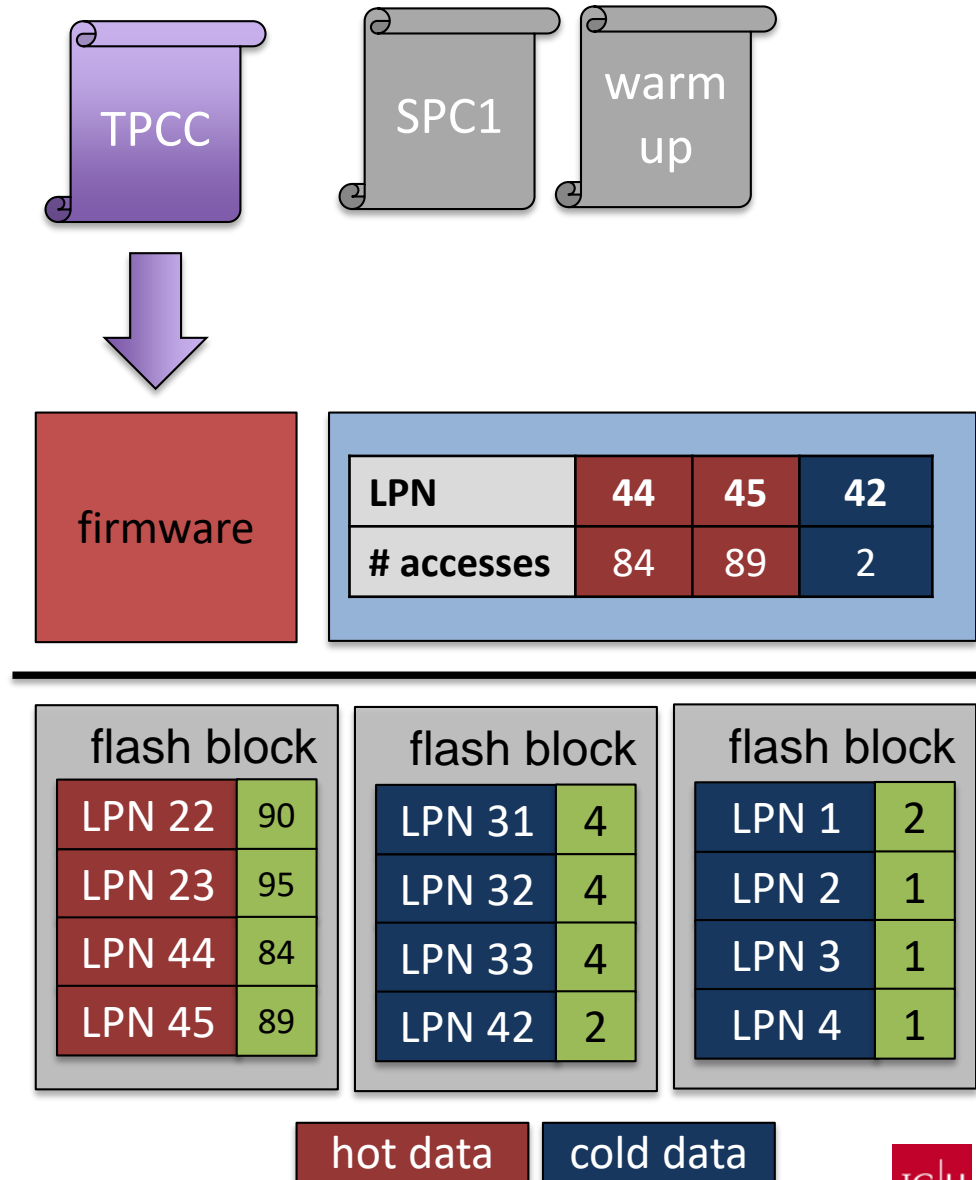
# Evaluation

- 950 GB flash memory  
326 MB DRAM
- 7% overprovisioning
- warm up phase
- run SPC1 and TPCC sequentially

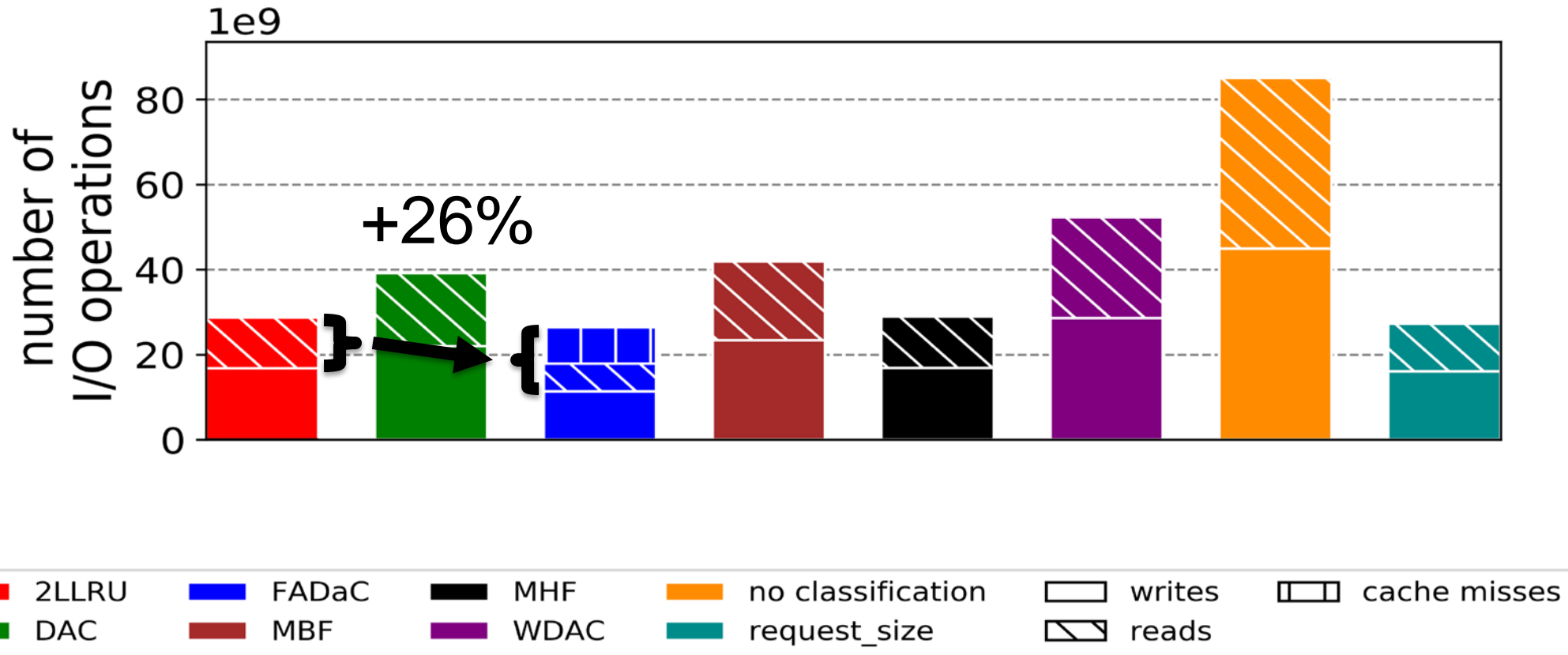


# Evaluation

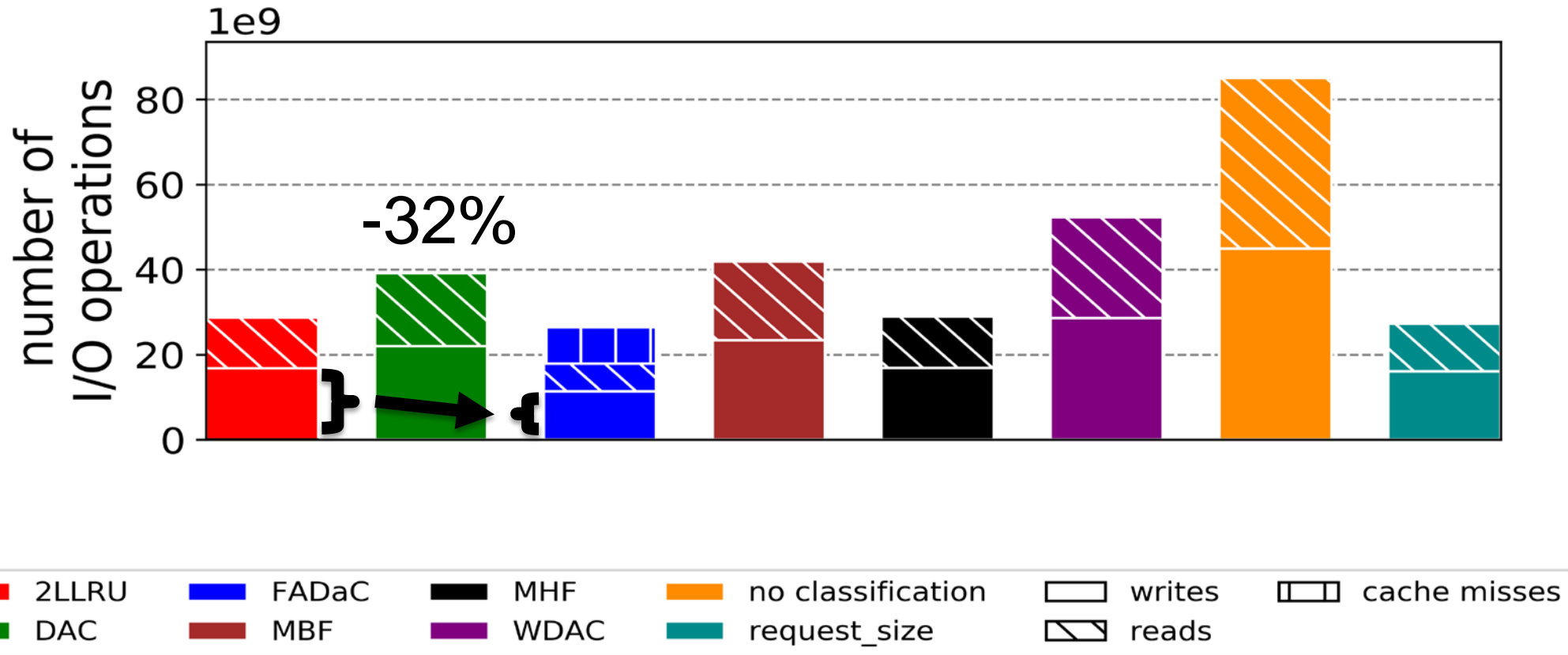
- 950 GB flash memory  
326 MB DRAM
- 7% overprovisioning
- warm up phase
- run SPC1 and TPCC sequentially



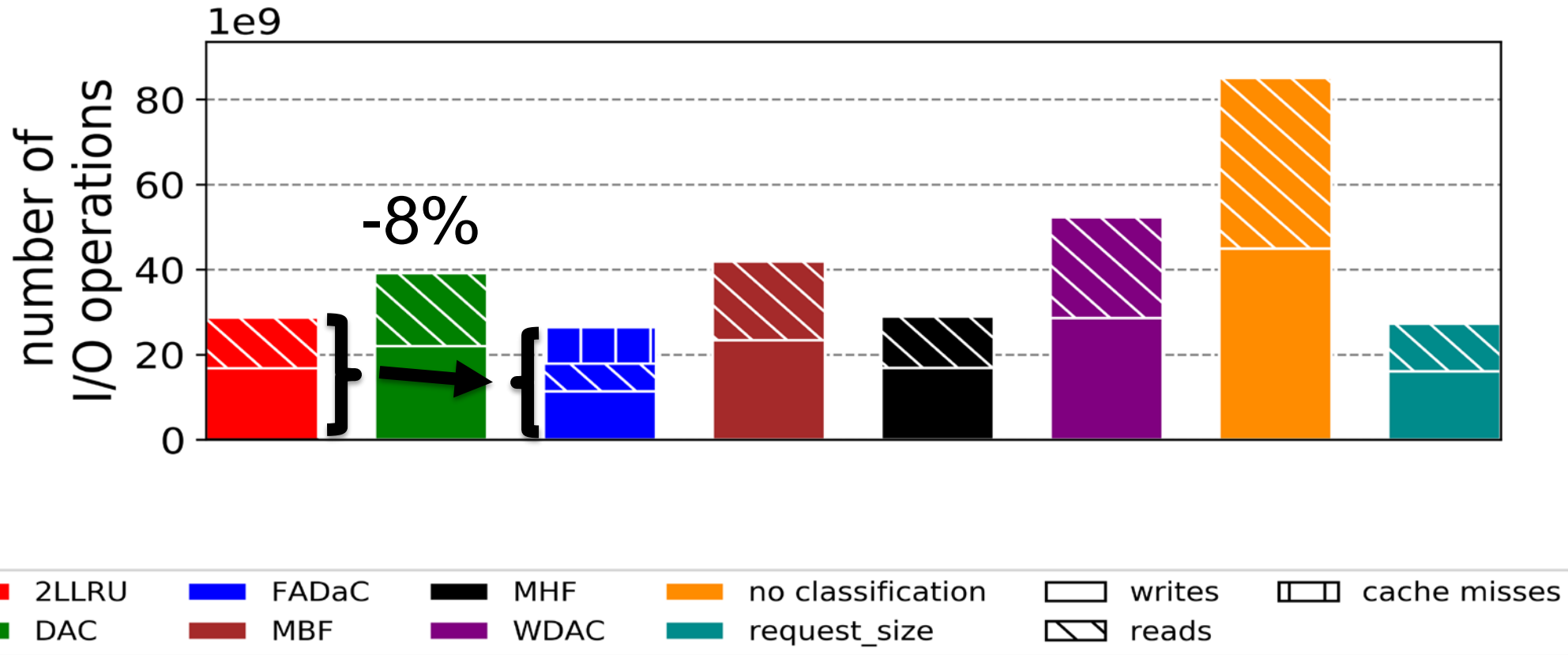
# Number of I/O operations



# Number of I/O operations



# Number of I/O operations



Please refer to the paper for more results



# Summary

- store access counters in spare region
- use DRAM as a cache only
- manage access counters when written or during GC
- use a self-adapting classification threshold

