A Reconfigurable FTL Architecture for NAND Flash Based Applications

Park, C., Cheon, W., Kang, J., Roh, K., Cho, W., and Kim, J. 2008

2023.07.25

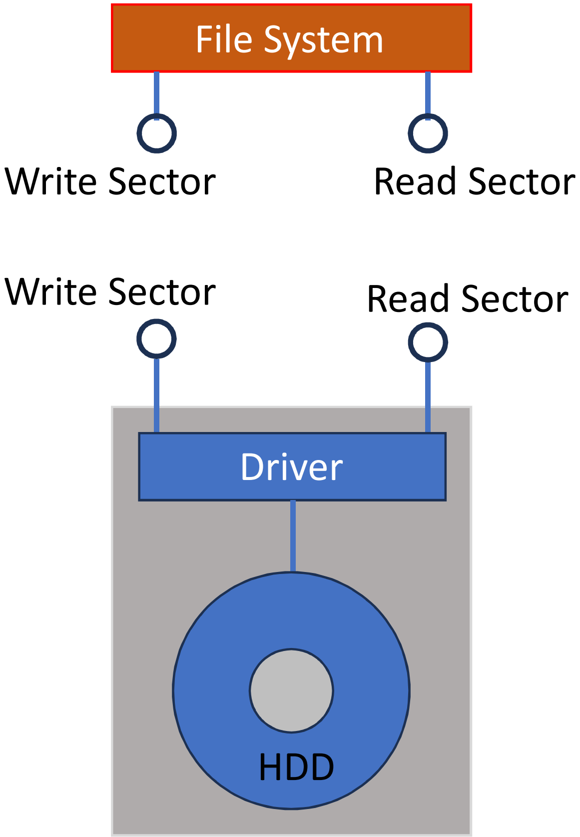
Presentation by Kim Boseung, 쥬용지에

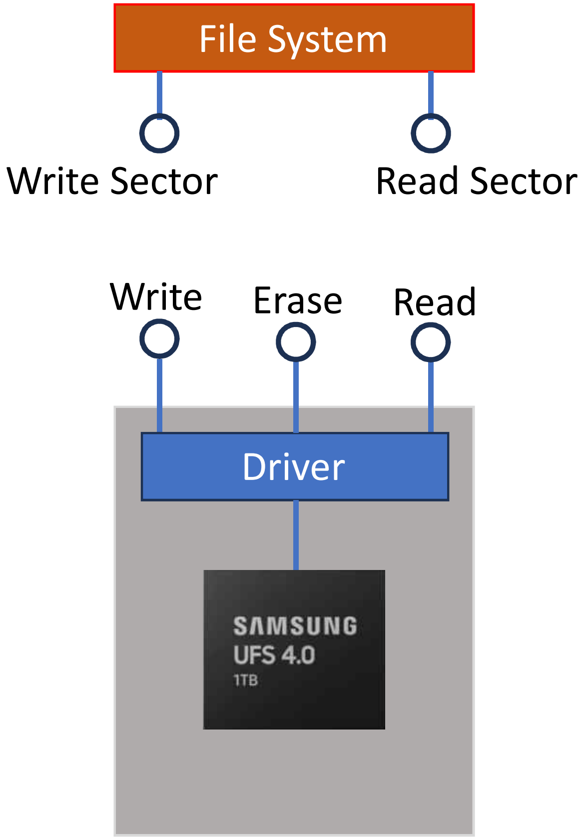
kbskbs1102@dankook.ac.kr

Contents

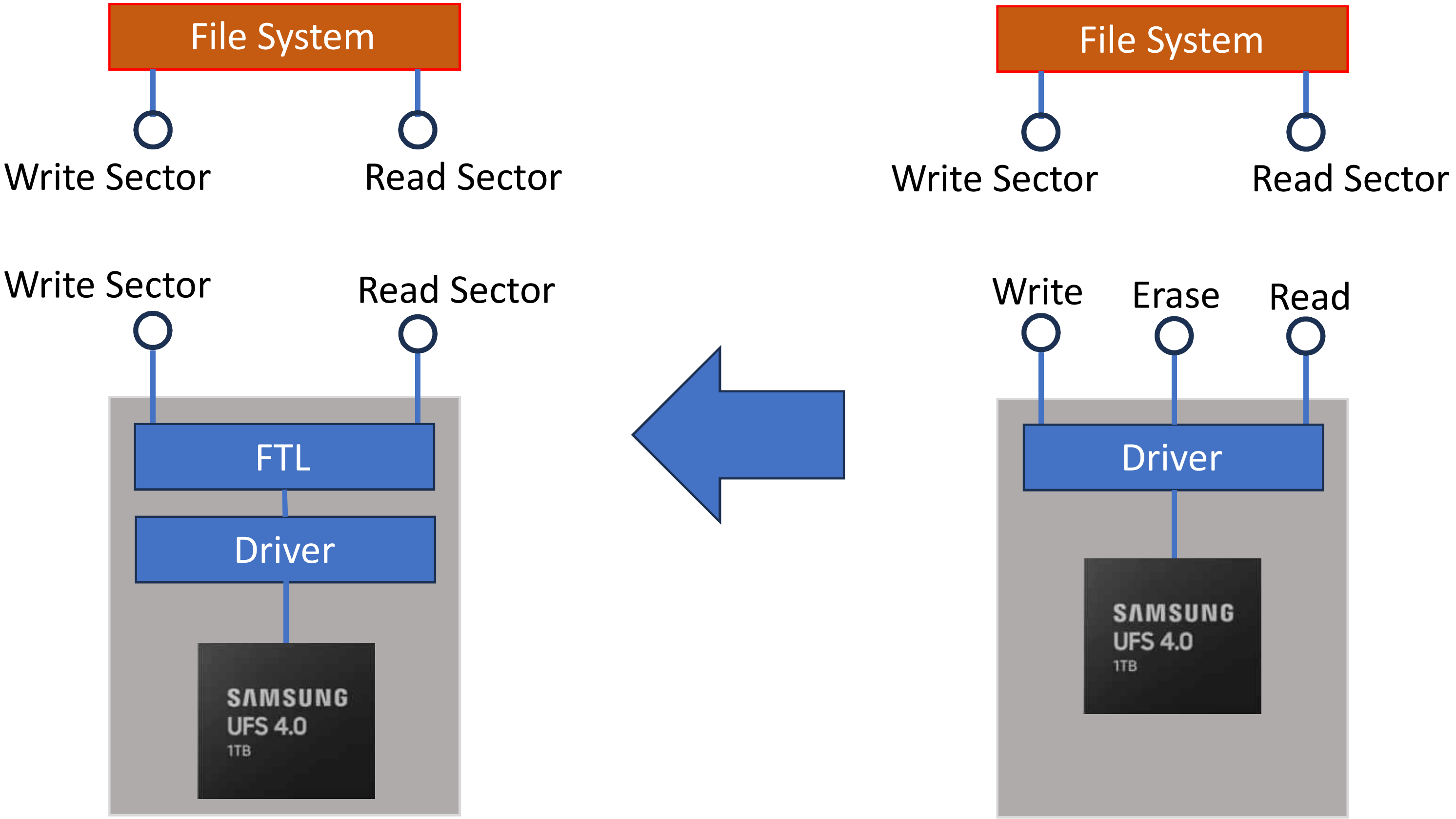
1. Introduction

1. Introduction





1. Introduction



2. NAND Flash Structure

2. NAND Flash Structure

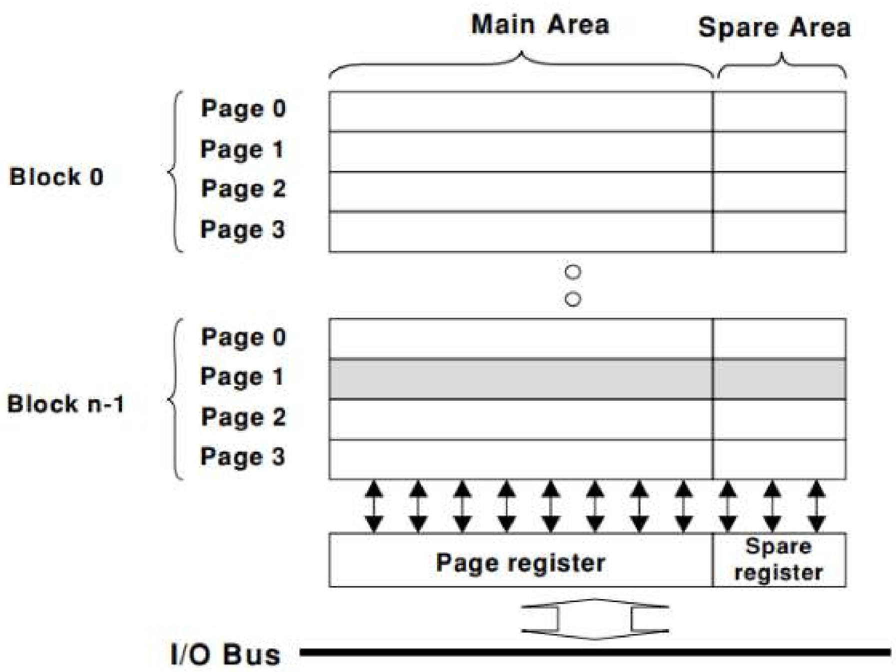


Fig. 3. NAND flash structure.

The **read/write unit** in NAND flash is a “**page**”, and **erase unit** is a “**block**”.
But it adopts the **Erase Before-Write approach**.

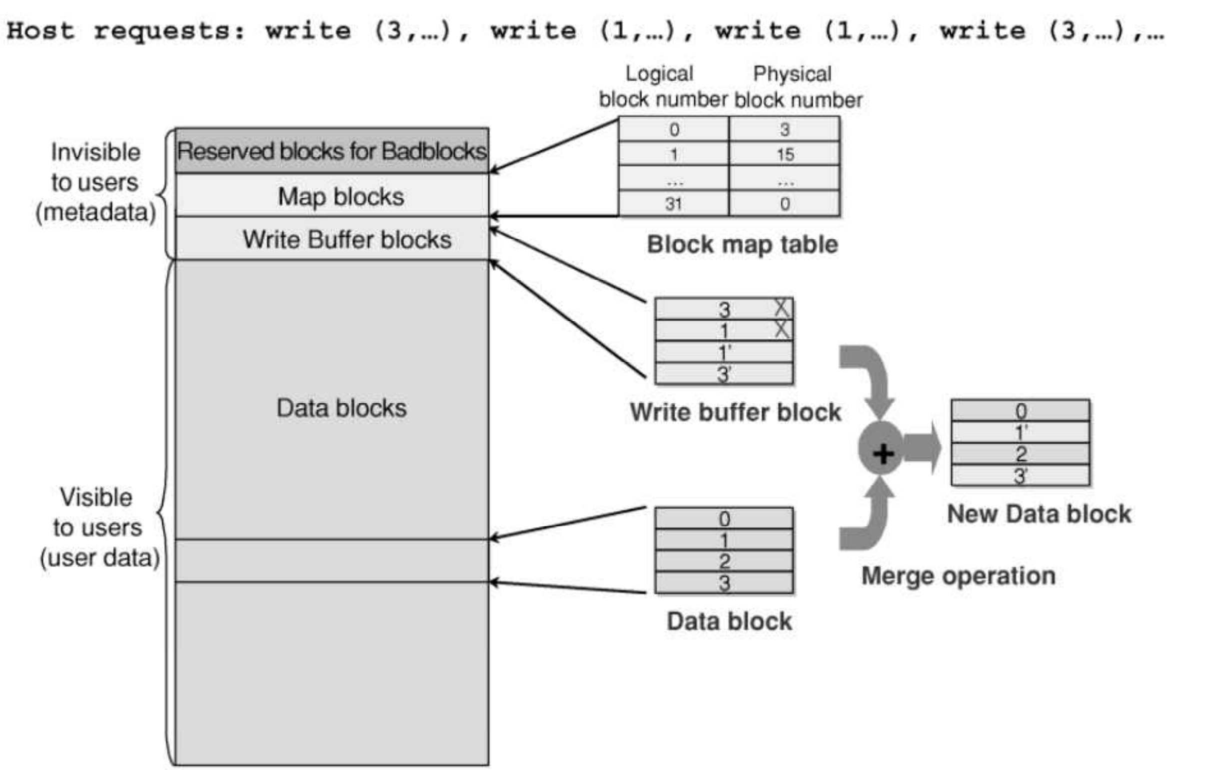
Main Area: The main area is used to store **user data**.

Spare Area: The spare area is used to store auxiliary **information** and **metadata**:
Bad-block Identification;
Error-Correction Code(ECC).

3. FTL Concepts

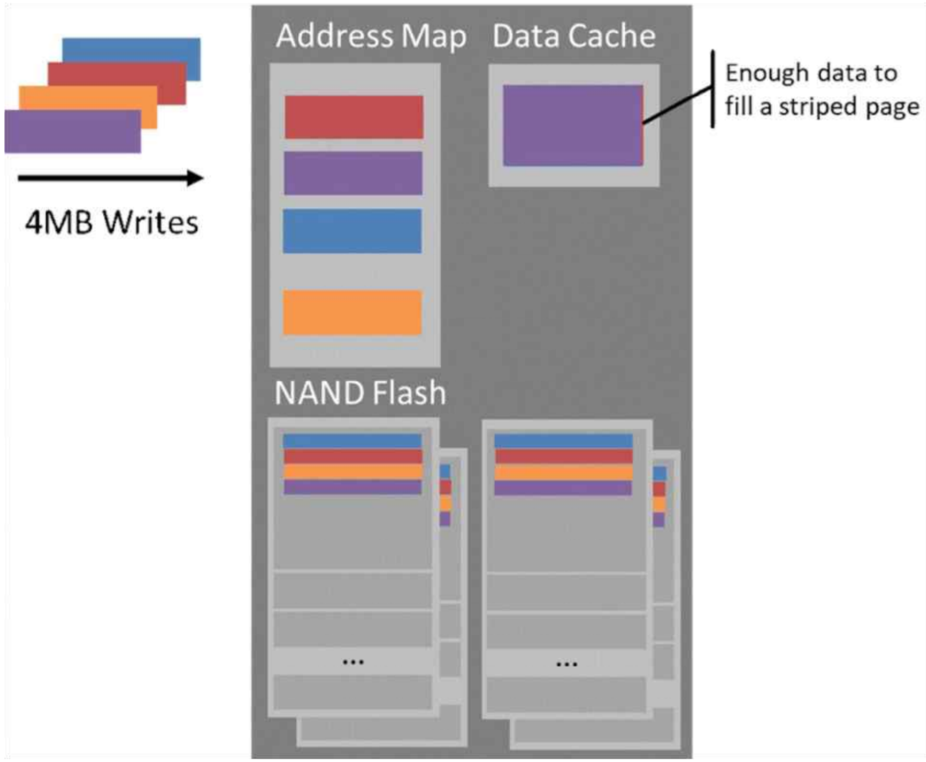
3. FTL Concepts

- **Logical view of the FTL of NAND flash memory.**



3. FTL Concepts

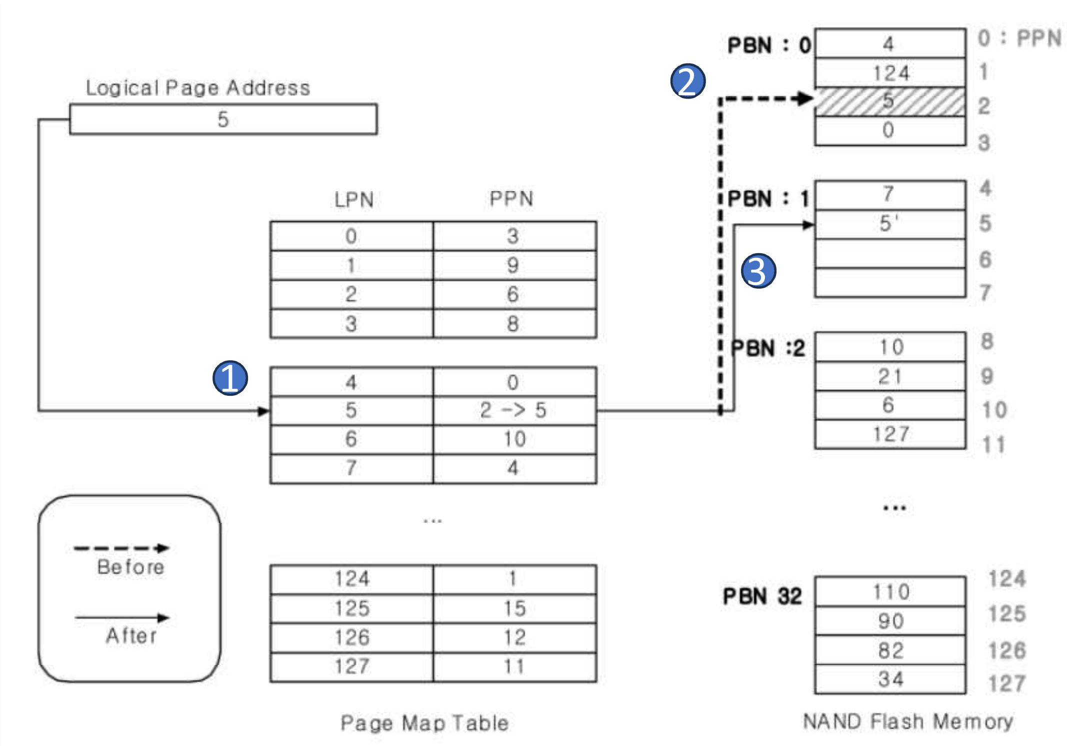
- **Logical view of the FTL of NAND flash memory.**



4. Mapping Scheme

4. Mapping Scheme

- **Page-mapping scheme**



DRAM Cost ⇧

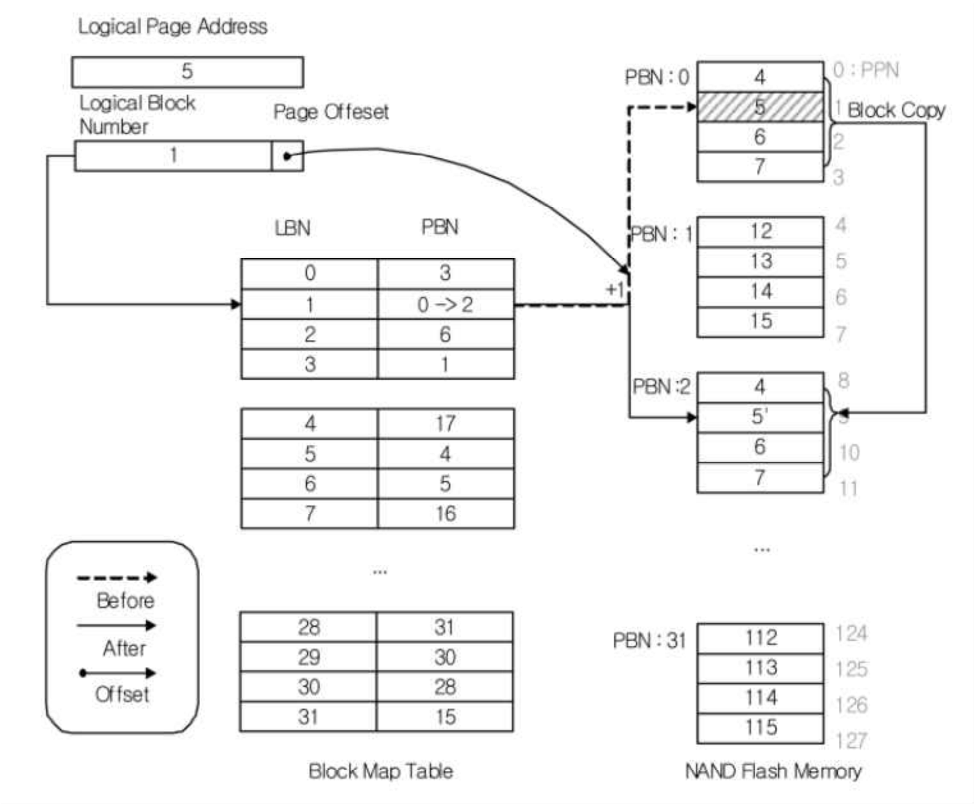
GC ⇩

Write Amplification

Random Write Performance ⇩

4. Mapping Scheme

- **Block-mapping scheme**

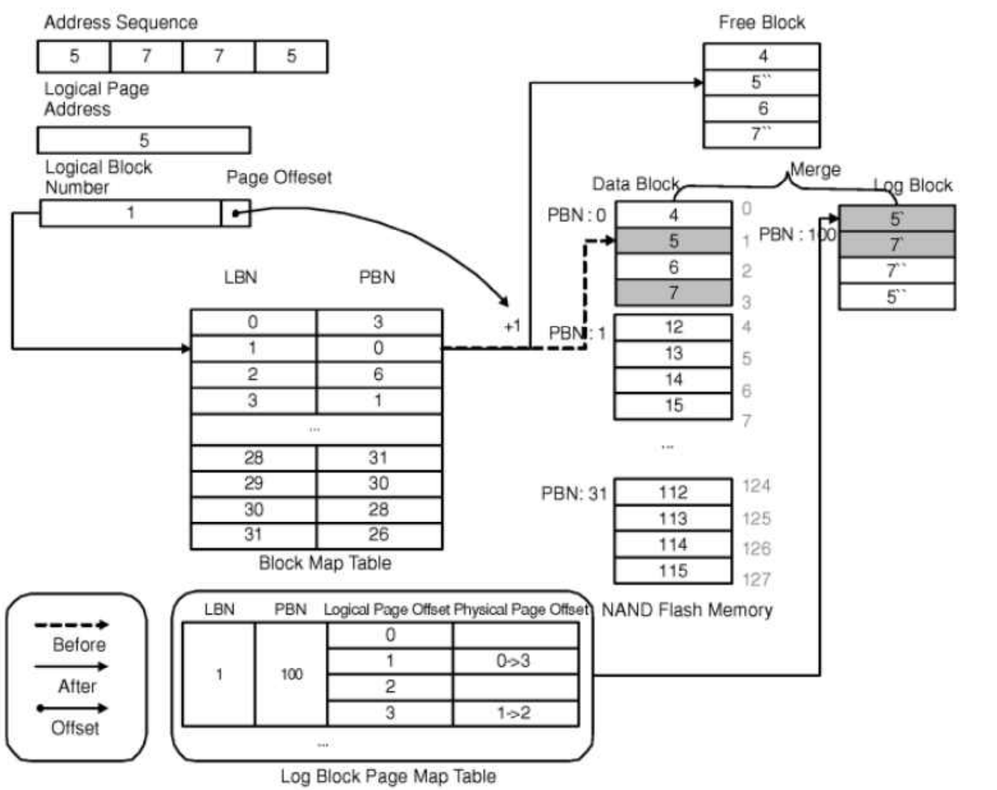


Block internal waster

Block-level copying

4. Mapping Scheme

- **Hybrid-mapping scheme**



Latency
(When Log Blocks are full or merging)

5. Flexible Group Mapping

5. Flexible Group Mapping

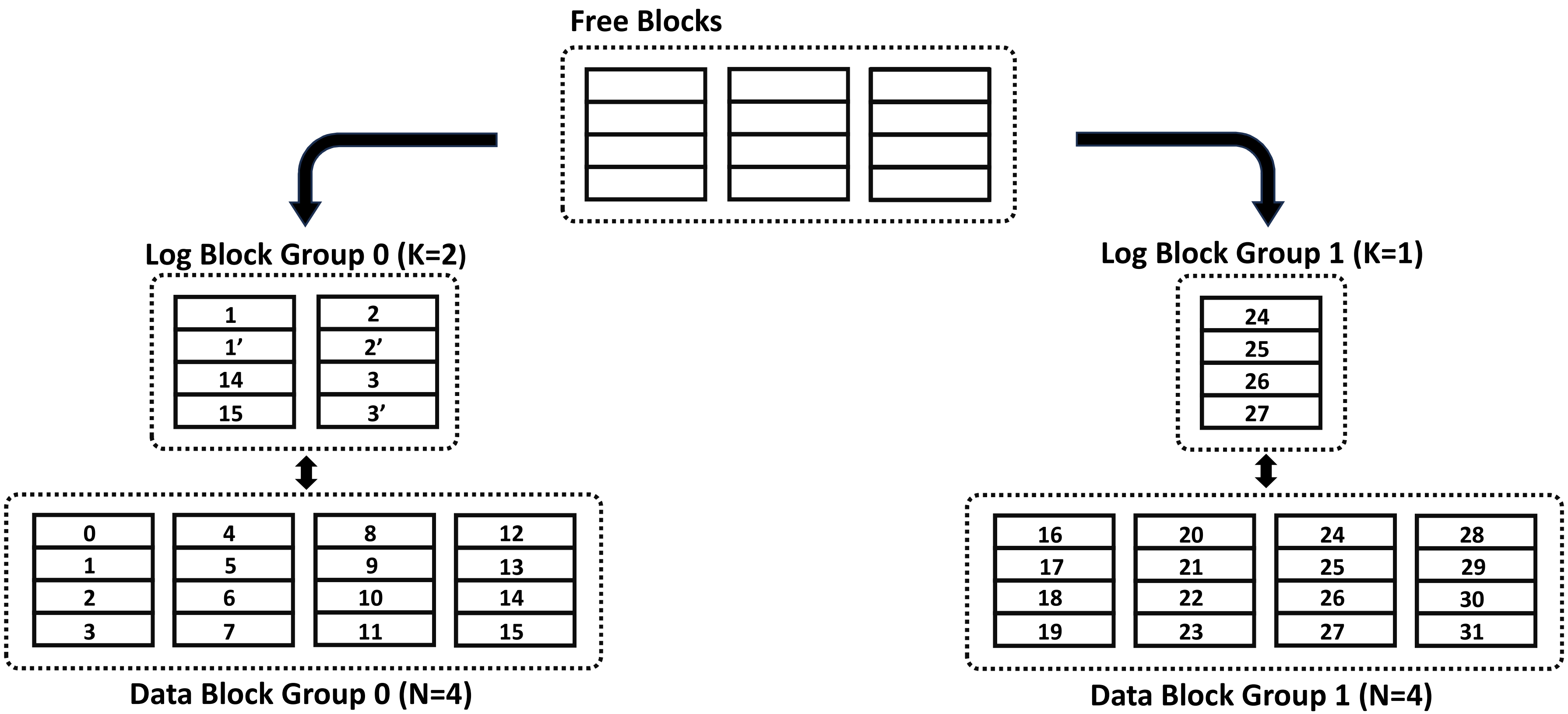
- Basic idea

Parameter N: Block level spatial locality parameter
Indicate the associativity among neighboring blocks
number of data blocks in a data block group

Parameter K: Delayed merge parameter
Explain the type of temporal locality within a block
maximum number of log blocks

- **Flexible Group Mapping Scheme**

Write request : 1, 1, 14, 15, 2, 2, 3, 3, 24, 25, 26, 27



5. Flexible Group Mapping

LBN: Logical Block Number
PBN: Physical Block Number
DGN: Data Block Group Number
LPN: Logical Page Number
PPN: Physical page Number

- **Write operation (N = 4, K = 2)**

1) WRITE: LPN = 3, Num of Pages = 2
2) WRITE: LPN = 11, Num of Pages = 4
3) WRITE: LPN = 17, Num of Pages = 4

LBN	PBN
0	100
1	101
2	102
3	103
4	104
5	105
6	206
7	303

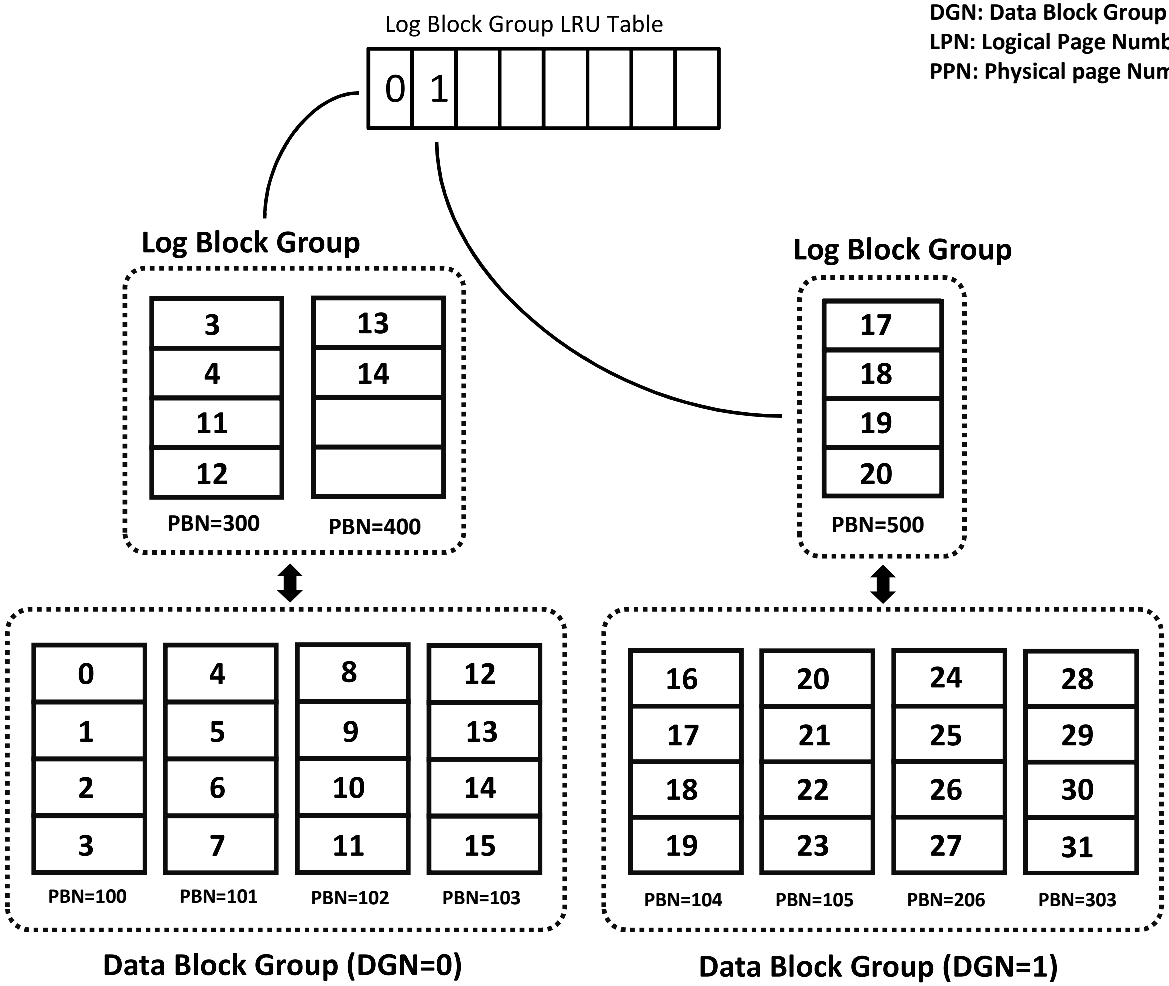
Data Block Mapping Table

DGN	PBN
0	300
	400
1	500
2	

Log Block Mapping Table

DGN	LPN	PPN
0	3	1200
	4	1201
	11	1202
	12	1203
	13	1600
	14	1601
1	17	2000
	18	2001
	19	2002
	20	2003

Log Page Mapping Table



*DGN = LPN div (N × the number of pages per block)

5. Flexible Group Mapping

LBN: Logical Block Number
PBN: Physical Block Number
DGN: Data Block Group Number
LPN: Logical Page Number
PPN: Physical page Number

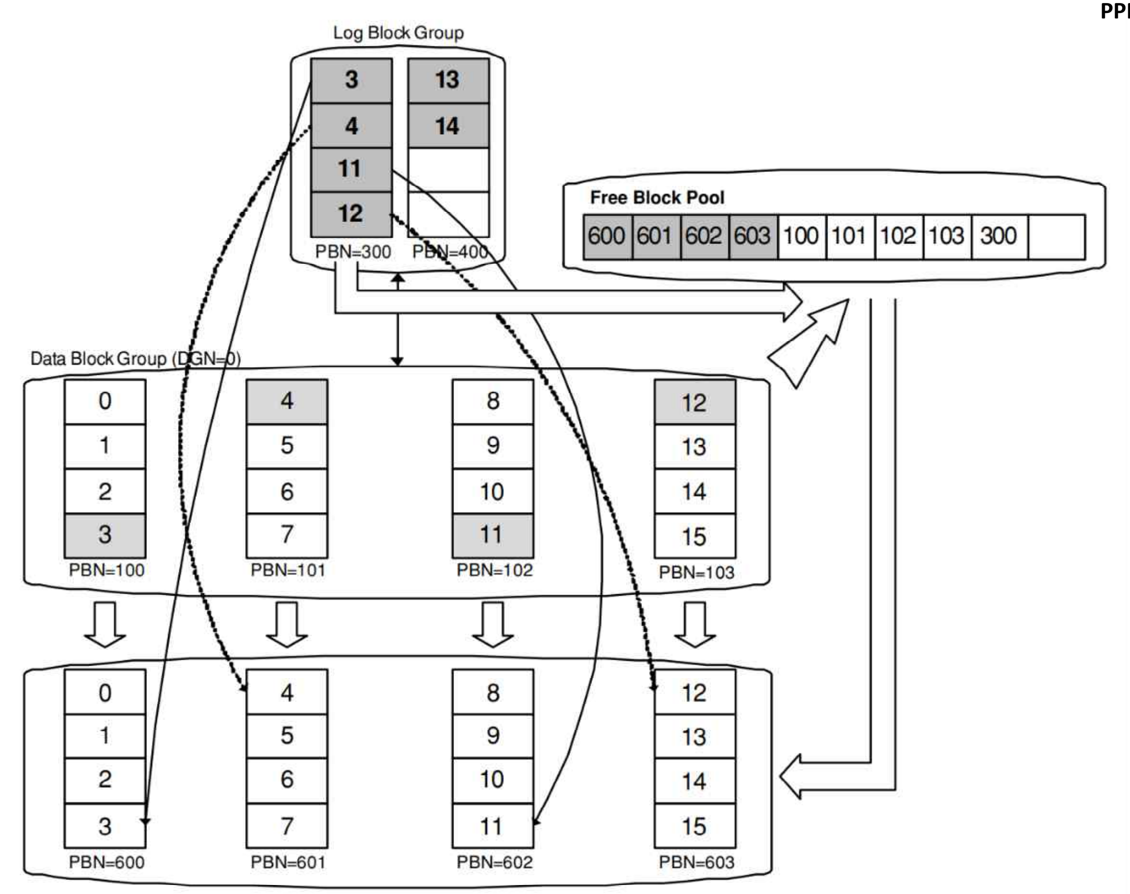
- **Simple Merge operation**

LBN	PBN
0	600
1	601
2	602
3	603
4	104
5	105
6	206
7	303

Data Block Mapping Table

DGN	PBN
0	
	400
1	500
2	

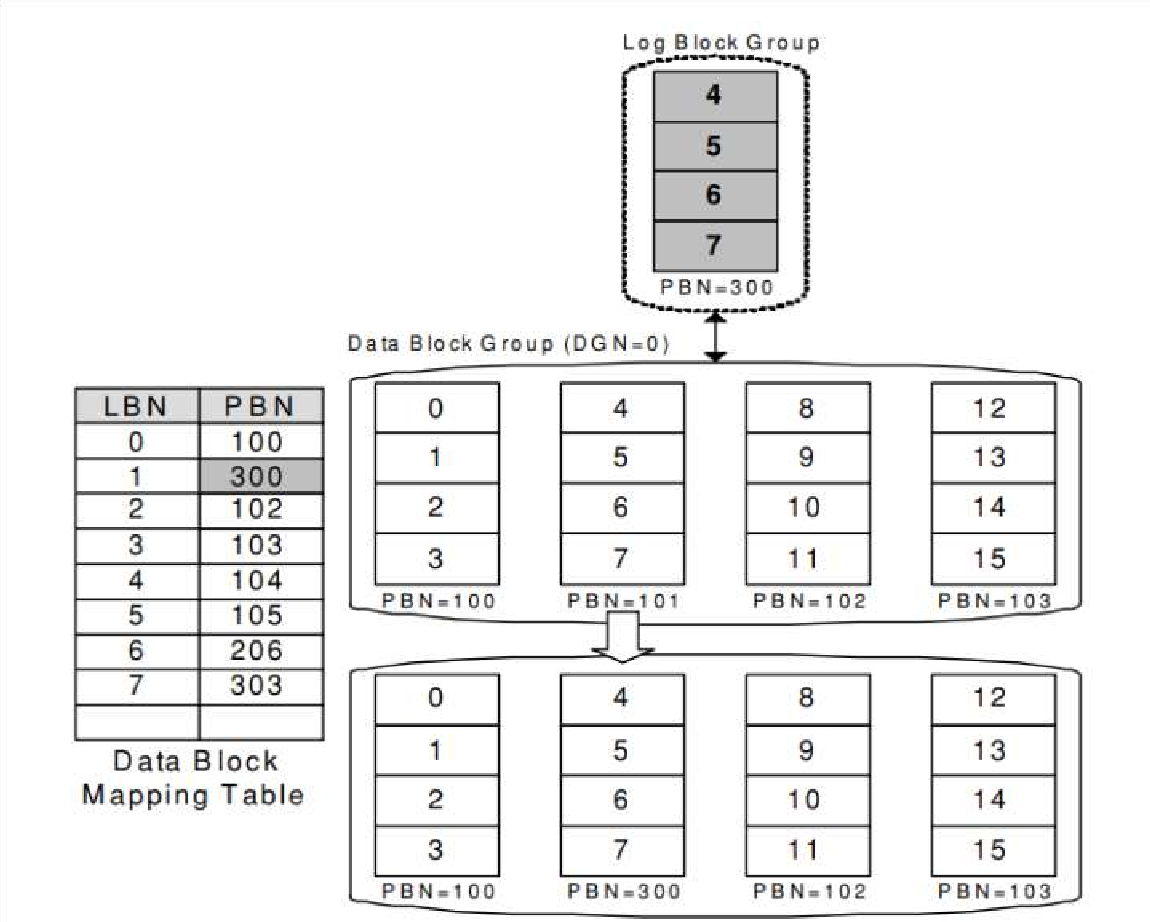
Log Block Mapping Table



5. Flexible Group Mapping

LBN: Logical Block Number
PBN: Physical Block Number
DGN: Data Block Group Number
LPN: Logical Page Number
PPN: Physical page Number

- **Swap Merge operation**



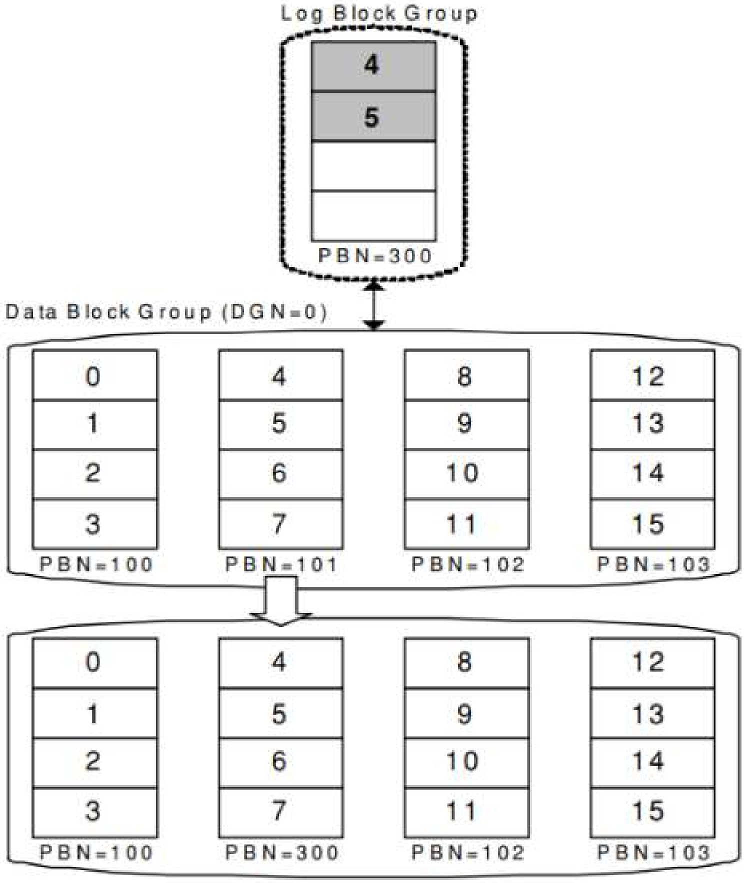
5. Flexible Group Mapping

LBN: Logical Block Number
PBN: Physical Block Number
DGN: Data Block Group Number
LPN: Logical Page Number
PPN: Physical page Number

- **Copy Merge operation**

LBN	PBN
0	100
1	300
2	102
3	103
4	104
5	105
6	206
7	303

Data Block Mapping Table



6. Analysis

- **Workload Analysis**

$$R = \langle R_0, R_1, \ldots, R_{M-1} \rangle \qquad R = kth\ write\ request$$

$$W_j = jth\ request\ window\ of\ size\ |W|$$
$$contains\ R_{j*|W|}, \ldots, R_{(j+1)*|W|-1} \quad for\ j = 0,1,\ldots,N_w - 1$$

$$|W| = \#\ of\ pages\ inside\ a\ block,\ N_w = total\ \#\ of\ W_j$$

$$RD = request\ density = C/|W|$$

6. Analysis

- **Workload Analysis**

LBN	Page	W_0	W_1	W_2	W_3	W_1
LBN 0	Page 0					
	Page 1	R_1				
	Page 2	R_2				
	Page 3	R_3				R_{16}, R_{17}
LBN 1	Page 4		R_4, R_6			
	Page 5					
	Page 6		R_5, R_7			
	Page 7					
LBN 2	Page 8			R_8		
	Page 9			R_9		
	Page 10				R_{14}	R_{18}
	Page 11					
LBN 3	Page 12				R_{15}	
	Page 13			R_{10}	R_{12}	R_{19}
	Page 14					
	Page 15			R_{11}	R_{13}	

Example Request Distribution Table

$$|W| = 4$$

$$C_{2,3} = 1 \;\Rightarrow\; RD_{0,0} = 1$$

$$C_{2,3} = 1 \;\Rightarrow\; RD_{2,3} = 0.25$$

6. Analysis

- **Workload Analysis**

N: Associativity parameter,

Determines how many LBNs are to be assigned to one log block group

N is high -> a large number of LBNs, N is low -> a small number of LBNS

$$\left\lfloor \left(\frac{1}{RD_{i,j}}\right) \right\rfloor \quad for\ i \in all\ LBNs$$

6. Analysis

- **Workload Analysis**

LBN	Page	W_3	W_2	W_1	W_3	W_2
LBN 0	Page 0	R_0				
	Page 1	R_1				
	Page 2	R_2				
	Page 3	R_3				R_{16}, R_{17}
LBN 1	Page 4		R_4, R_6			
	Page 5					
	Page 6		R_5, R_7			
	Page 7					
LBN 2	Page 8			R_8		
	Page 9			R_9		
	Page 10				R_{14}	R_{18}
	Page 11					
LBN 3	Page 12				R_{15}	
	Page 13			R_{10}	R_{12}	R_{19}
	Page 14					
	Page 15			R_{11}	R_{13}	

Request Distribution Table

$$N_j = \left\lfloor \min\left(\frac{1}{RD_{i,j}}\right) \right\rfloor$$

	W_0	W_1	W_2	W_0	W_2
LBN 0	1.00	0.00	0.00	0.00	0.50
LBN 1	0.00	1.00	0.00	0.00	0.00
LBN 2	0.00	0.00	0.50	0.25	0.25
LBN 3	0.00	0.00	0.50	0.75	0.25
N_j	1	1	2	4	4

Estimating N:

Estimating N:

6. Analysis

- **Workload Analysis**

K: Delayed merge parameter

$$K_{i,0} = 0$$

$$K_{i,j} = K_{i,j-1} + d_{i,j} \quad for\ j > 0$$

* $PAGE$ = set of pages in LBN i written during W

* $PAGE$ = set of pages in LBN i written during W

6. Analysis

- **Workload Analysis**

$K_{i,0} = 0$

		W_0	W_1	W_1	W_0	W_4
LBN 0	Page 0	R_0				
	Page 1	R_1				
	Page 2	R_2				
	Page 3	R_3				R_{16}, R_{17}
LBN 1	Page 4		R_4, R_6			
	Page 5					
	Page 6		R_5, R_7			
	Page 7					
LBN 2	Page 8			R_8		
	Page 9			R_9		
	Page 10				R_{14}	R_{18}
	Page 11					
LBN 3	Page 12				R_{15}	
	Page 13			R_{10}	R_{12}	R_{19}
	Page 14					
	Page 15			R_{11}	R_{13}	

Request Distribution Table

	W_0	W_0	W_0	W_0	W_0	K
LBN0	0	0	0	0	1	1
LBN1	0	1	1	1	1	1
LBN2	0	0	0	0	1	1
LBN3	0	0	0	1	2	2

Estimating K_i

6. Analysis

- **Performance Analysis**

A_K = active data block accessed by R_K

$SA(W_i)$, $SAG(W_i)$ = set of active blocks and active block group

$L(AG_k)$ = number of log blocks associated with AG_k

LB = maximum number of log blocks

6. Analysis

- **Performance Analysis**

$$L(AG_k) \le K$$

$$|GAG(W_j)| \approx K^{\partial} \times LB \quad for \ (0 \le \partial \le 1)$$

∂ is associated with the update frequency for the input pattern

6. Analysis

- **Performance Analysis**

$$\frac{|SA(W\ldots)|}{N^{\varepsilon}}$$

ε is associated with the associativity of the input pattern

ε is associated with the associativity of the input pattern

When associativity is strong $\rightarrow$ ε go to 1

When associativity is strong $\rightarrow$ ε go to 1

When associativity is strong $\rightarrow$ ε go to 1

When associativity is strong $\rightarrow$ ε go to 1

6. Analysis

- **Performance Analysis**

When the size of W_j is reasonably large: $|SA(W_j)| = C$

$$\sum_{AG_k \in SAG(W_j)} \qquad |SAG(W_j)| = \frac{|SA(W_j)|}{N^{\varepsilon}}, \quad |SA(W_j)| = C$$

$$\sum_{AG_k \in SAG(W_j)} \quad \Rightarrow \quad \frac{K^{\partial}}{N^{\varepsilon}}$$

6. Analysis

- **Memory Requirement Analysis**

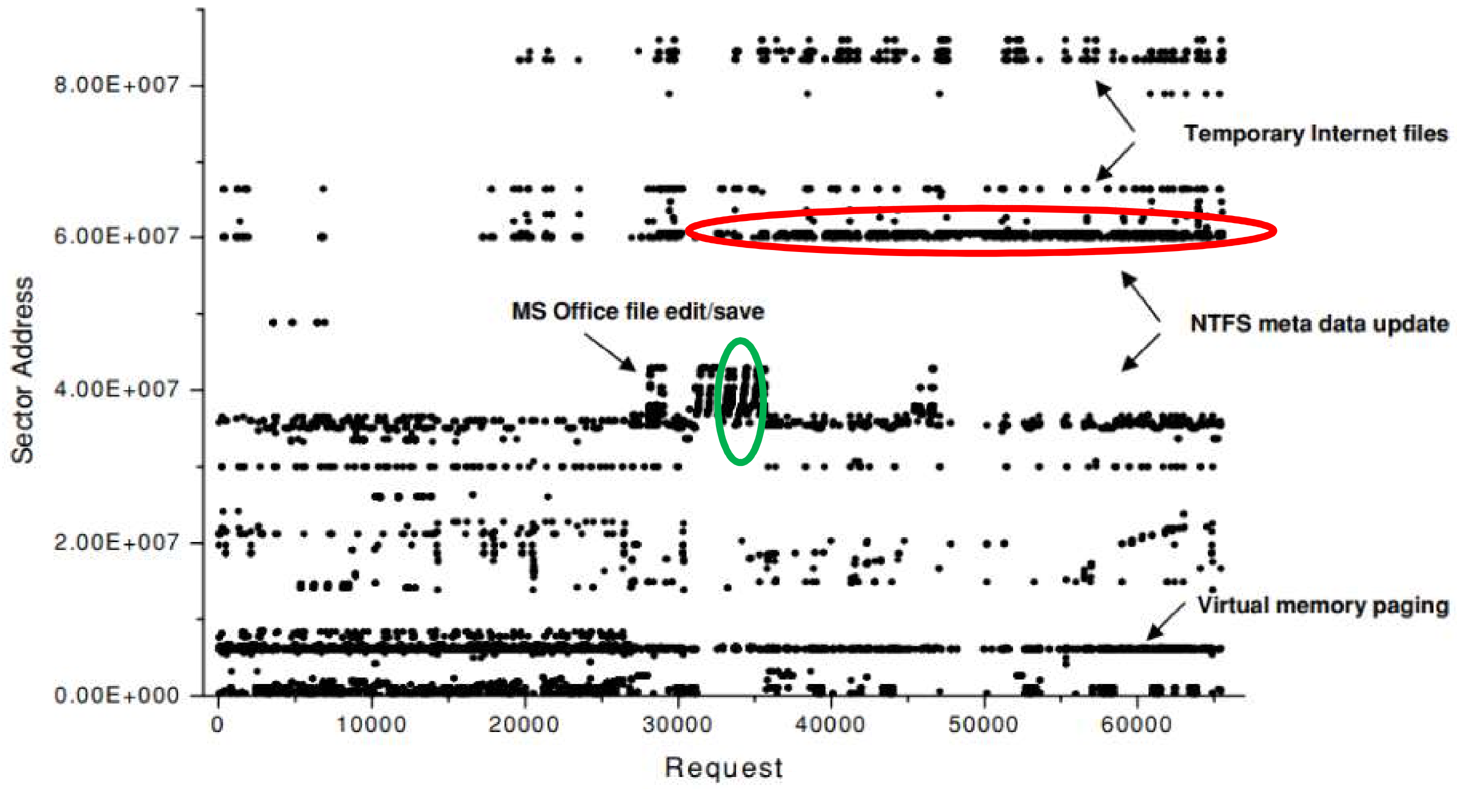
$c_2 K$ = the number of valid pages in the log

c_3 = the number of logs

7. Experimental Results

7. Experimental Results

- **Trace distribution**



Internet/MS Office use case

○ : **temporal locality**

○ : **spatial locality**

∂ is close to 1

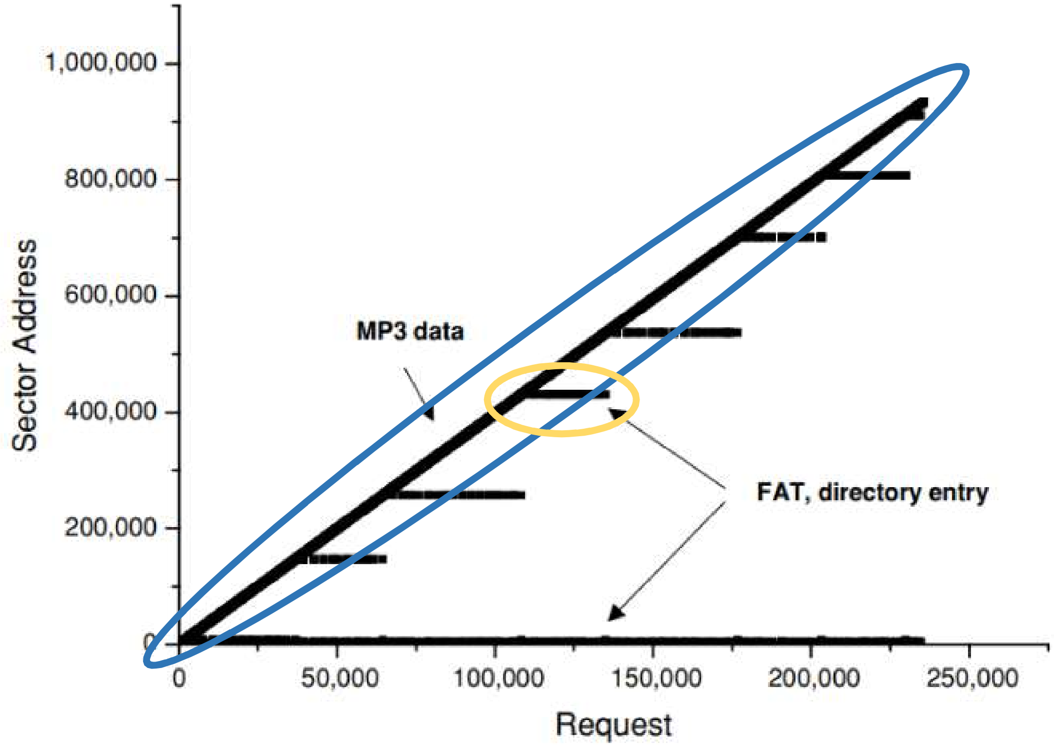
∂ is close to 1

ε is not large

$$C \times \frac{K^{\partial}}{N^{\varepsilon}} > LB$$

7. Experimental Results

- **Trace distribution**



MP3 download use case

○ : **Sequential access**

○ : **Random access**

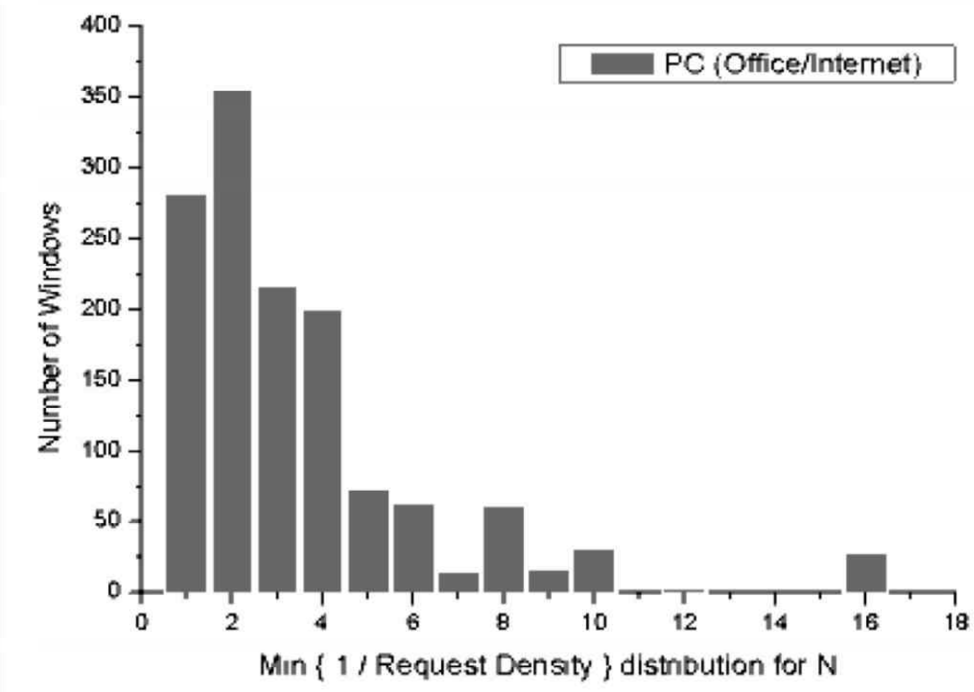
∂ is close to 0

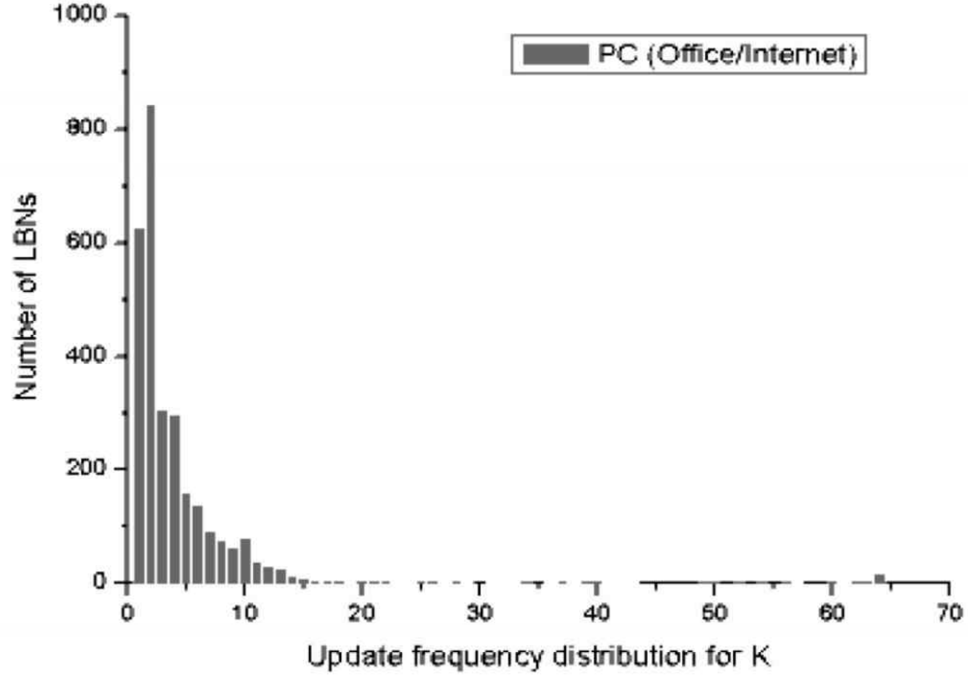
ε is close to 0

$$C \times \frac{K^{\partial}}{N^{\varepsilon}} > LB$$

7. Experimental Results

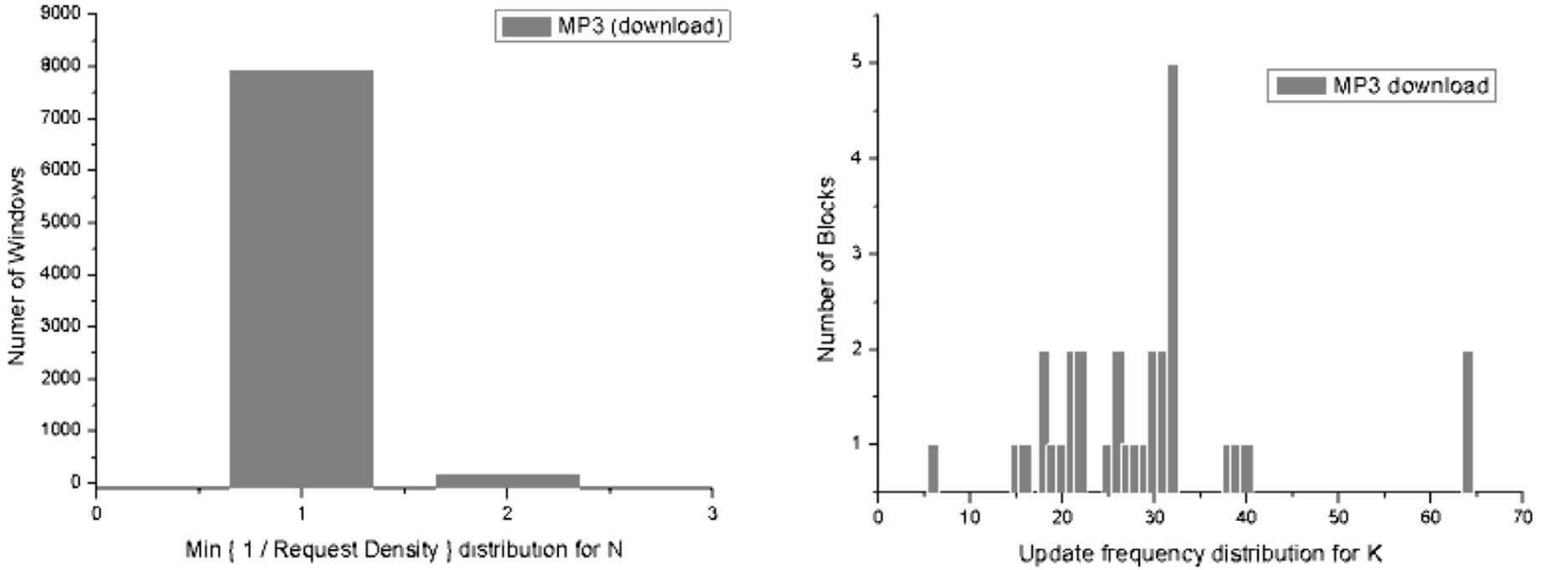
- **Distribution of N's and K's for PC applications.**





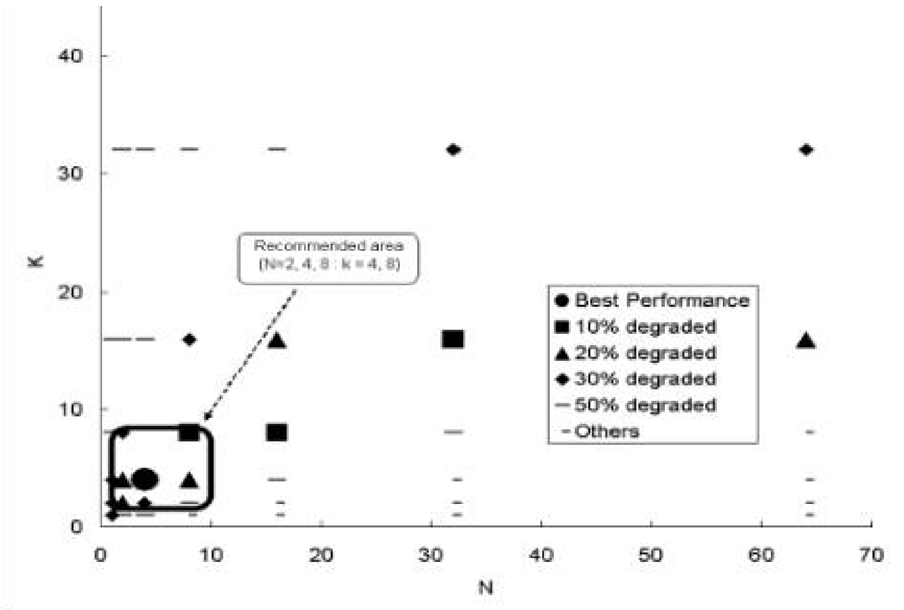
7. Experimental Results

- **Distribution of N and K values for the MP3 application**

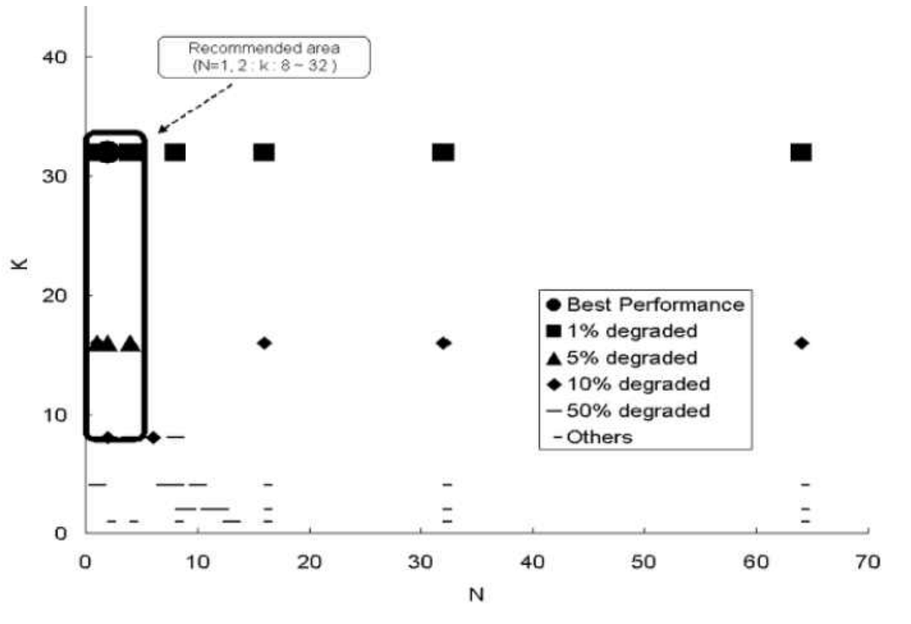


7. Experimental Results

- **Performance variation with the change of N and K**



PC application



MP3 download

8. Conclusions

8. Conclusions

- **Conclusions**

Introduced a reconfigurable FTL architecture to efficiently handle diverse NAND flash applications

The associativity between data blocks was parameterized using:

- **N**: the number of data blocks in a data block group
- **K**: the maximum number of log blocks in a log block group

The proposed analysis method
can efficiently find the optimal **N** and **K**

A Reconfigurable FTL Architecture for NAND Flash Based Applications

Park, C., Cheon, W., Kang, J., Roh, K., Cho, W., and Kim, J. 2008

Thank you!
Q & A ?

2023.07.25

Presentation by Kim Boseung, 쥬융지에

kbskbs1102@dankook.ac.kr