Mitigating Read Amplification with Dynamic Block Size Control in HBase

바닥까지 파보는! HBase random read 성능 개선기

김정민
NAVER Search
채상은
성균관대학교

NAVER DEVIEW 2023

CONTENTS

NAVER
DEVIEW
2023

1. Background - 배경 소개
2. 데이터 다양성 - 효율성과 공존하기
3. HBase read amplification – 정의 및 원인 분석
4. Solution – 개선 방안
5. Evaluation – 벤치마킹 및 적용 효과
6. Conclusion - 마무리

1. Background

배경소개

AIDA

ADVANCED INTERFACE
FOR DATA & AI

CUVE C3 CQuery Logiss AiSuite Hades Deathnote

- 저장
- 유통

1.1 네이버 데이터 저장 플랫폼: CUVE

NAVER DEVIEW 2023

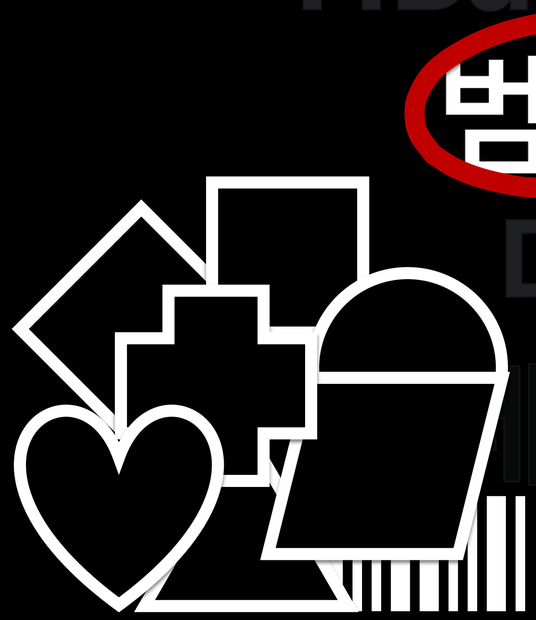
*Deview 2017 : HBase 기반 검색 데이터 저장소

HBase 기반의
범용적인
대규모
분산 데이터 저장소

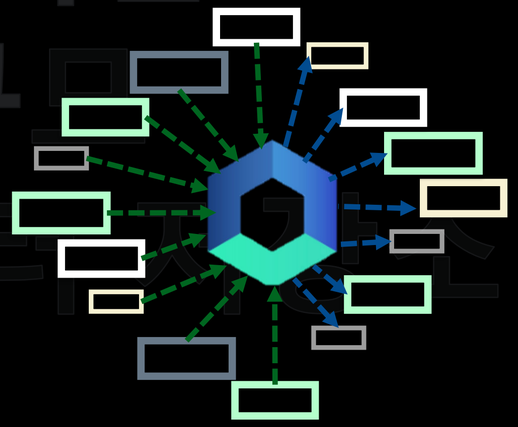
1.1 네이버 데이터 저장 플랫폼: CUVE

NAVER
DEVIEW
2023

HBase 기반의
범용적인
대규모
데이터저장소



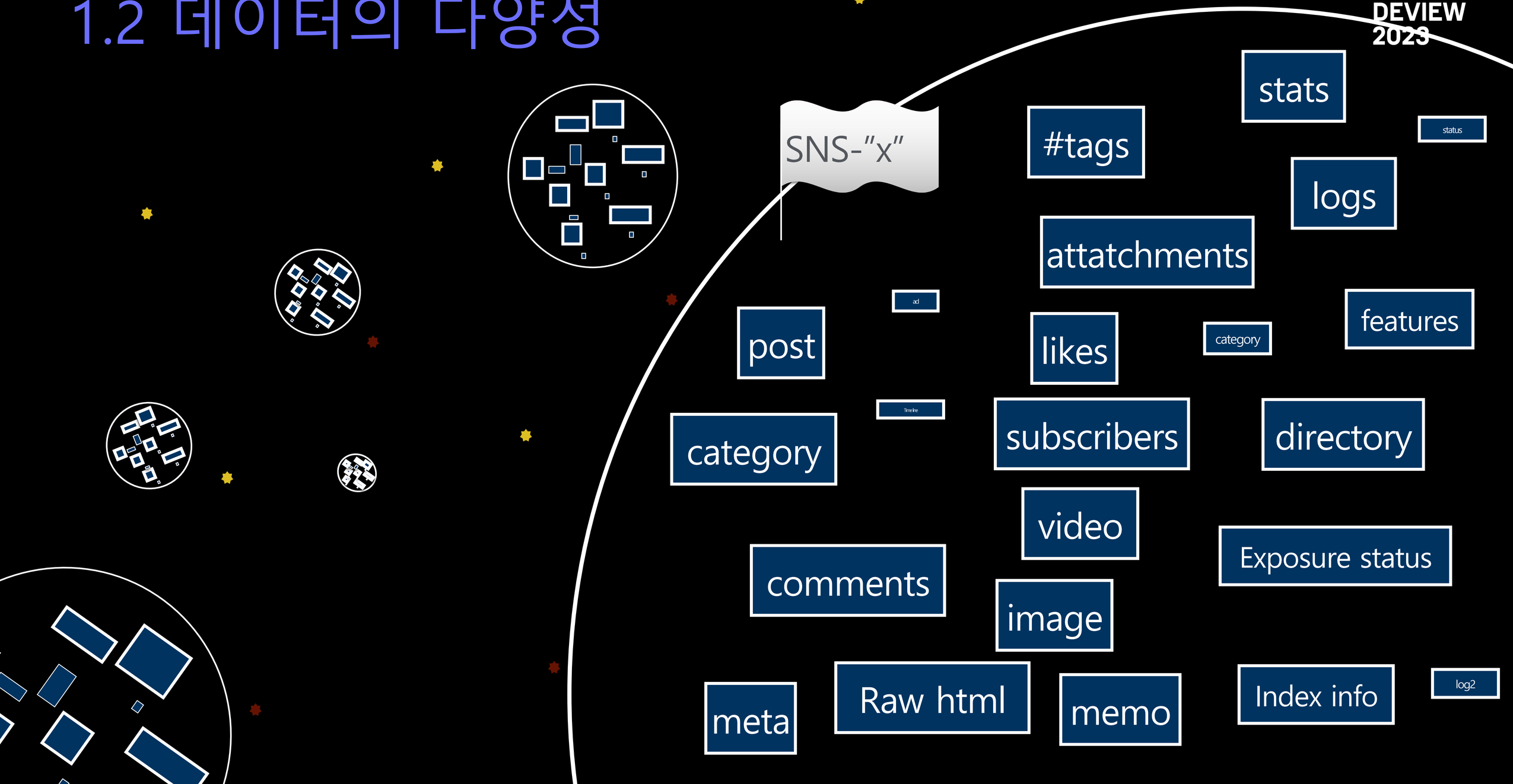
다양한 데이터



다양한 사용자

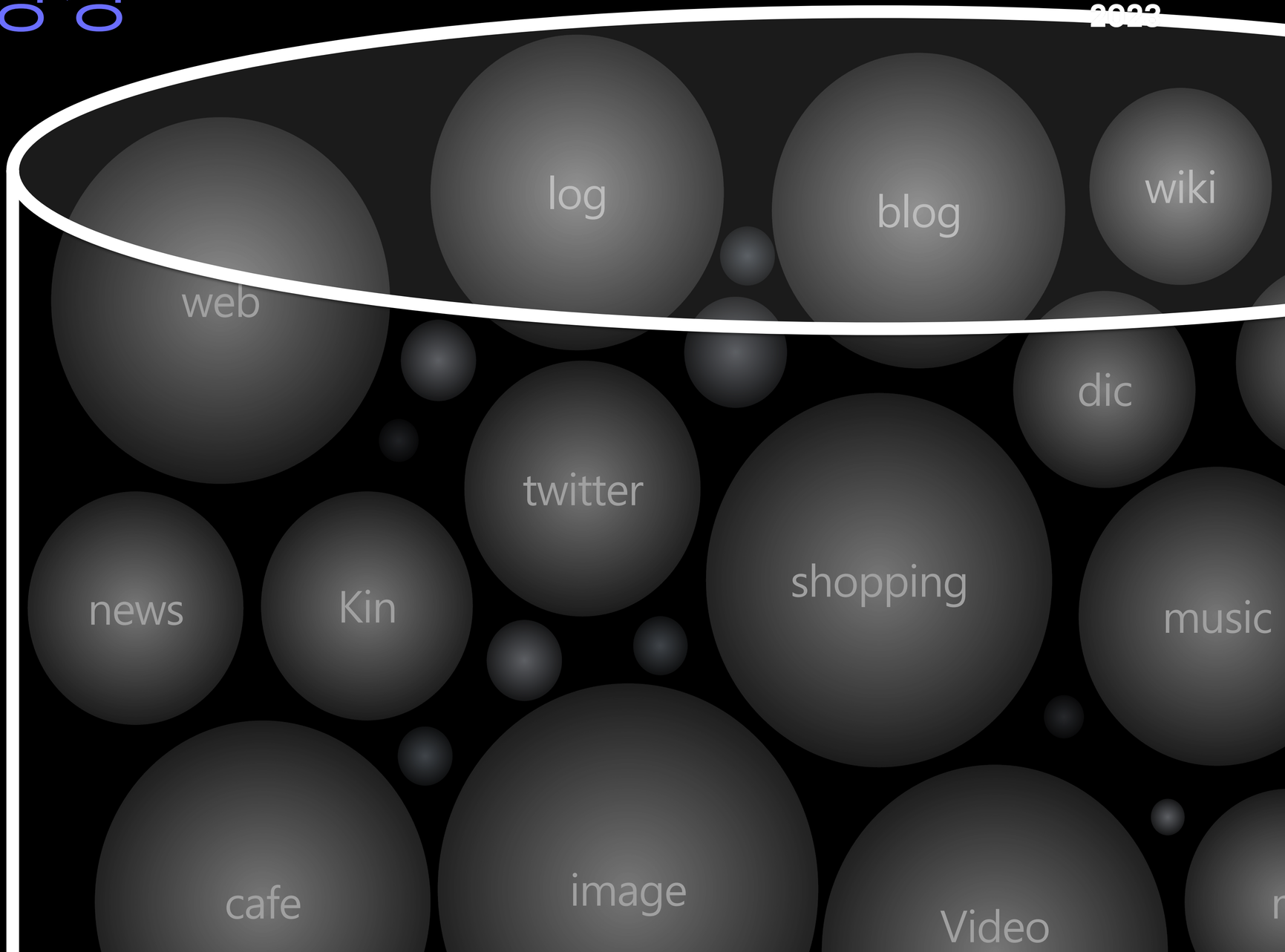
1.2 데이터의 다양성

NAVER DEVIEW 2023



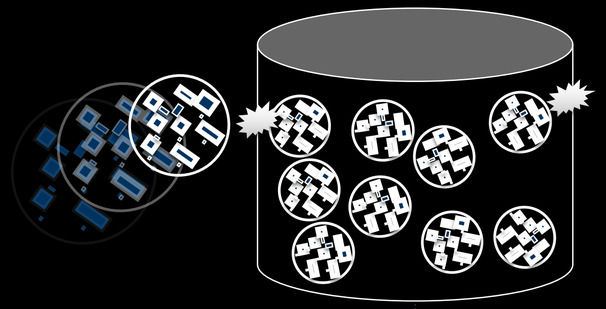
1.2 데이터의 다양성

NAVER
DEVIEW
2023



1.3 입출력 패턴의 다양성

NAVER
DEVIEW
2023

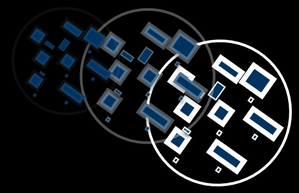


입력 패턴

- Read intensive vs Write intensive
- Timeline data vs non-time series
- Bulk update, Force update
- Update policy

  (del&create, partial, Immutable …)
- Migrations

…



출력 패턴

- Streaming
- Full-dump
- Random get
- Time-range scan
- Scan with filter
- Meta-only
- Prefix scan
- Secondary Index search

Deview 2015 : Hbase consistent 2ndary indexing

1.4 HBase와 저장소

NAVER
DEVIEW
2023

대규모

Deview 2021 : 네이버 최대의 데이터 저장소 운영기

- Scalability
- performance

범용성

- Schema-less
- Hadoop-eco friendly

Image : https://hbase.apache.org/

2. 데이터 다양성

- 효율성과 공존하기

2.1 효율적인 클러스터 운영

NAVER
DEVIEW
2023

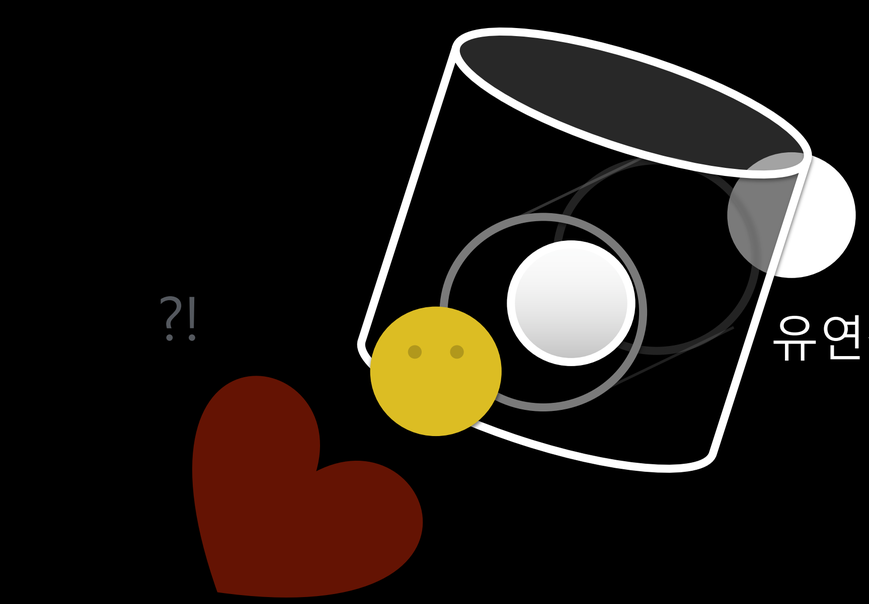
Data

사용
패턴

Image : https://unsplash.com/ko/%EC%82%AC%EC%A7%84/yny2UwDVo08

2.1 효율적인 클러스터 운영

NAVER
DEVIEW
2023



유연성의 결여

안정성 😟
효율성 ☹️

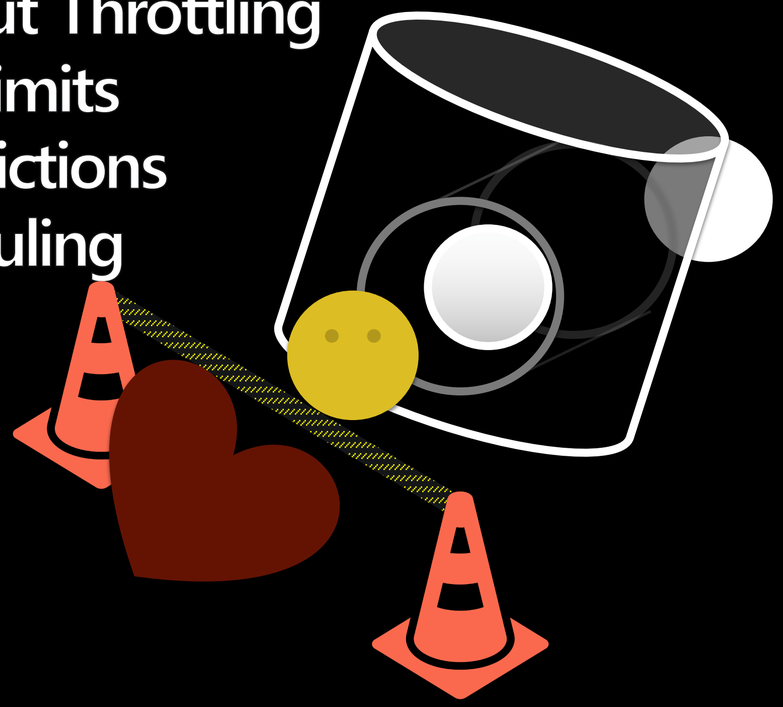
다양성에 의한 비효율

2.2 해결방안1 : 제약

NAVER
DEVIEW
2023

제약

- Throughput Throttling
- Resource limits
- Data Restrictions
- Job Scheduling

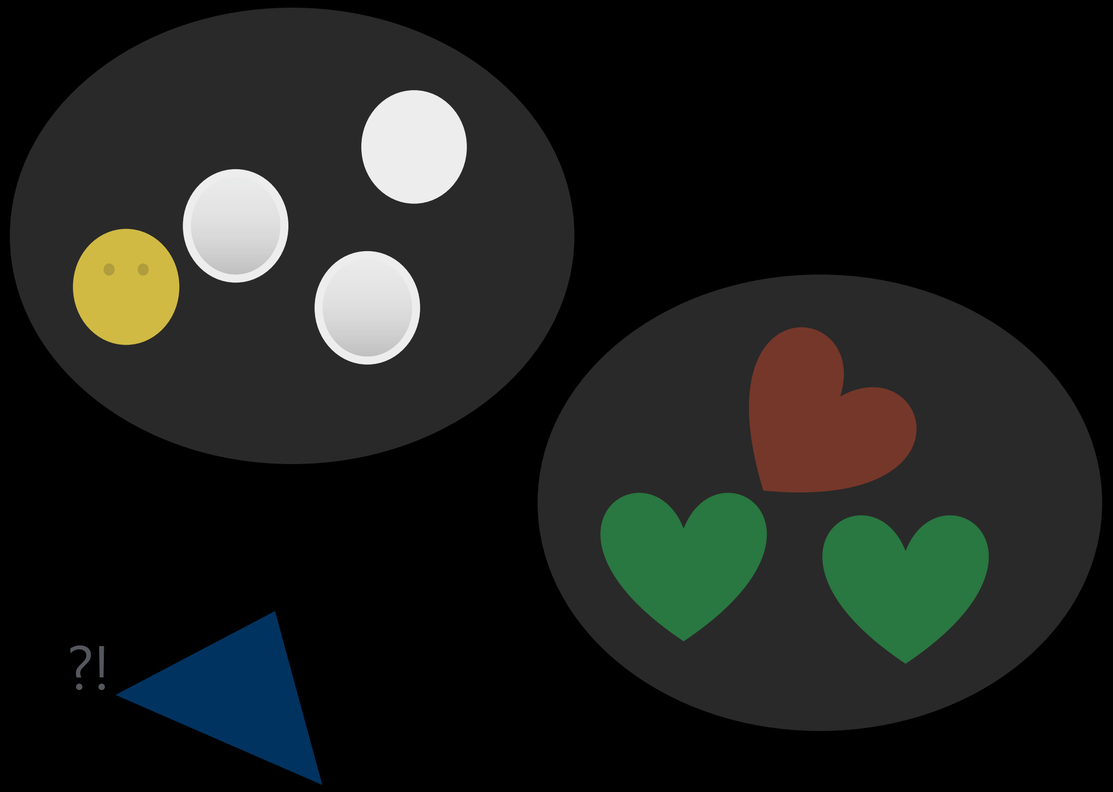


안정성 😀
효율성 😉

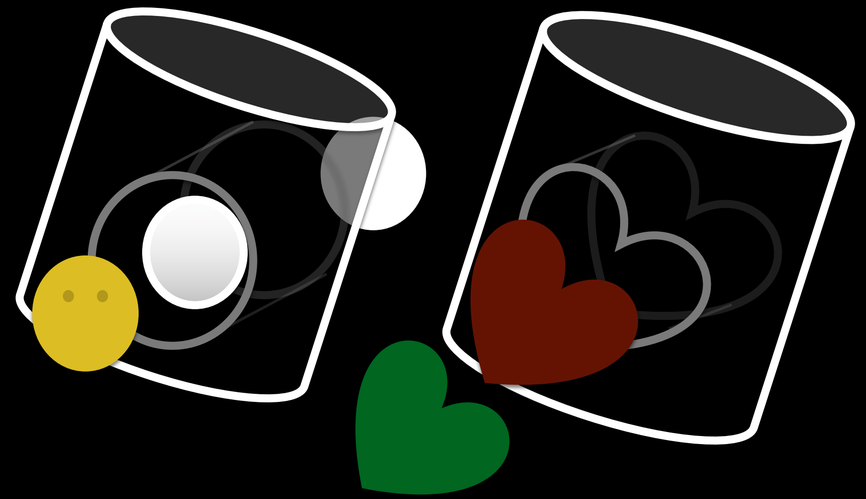
사용성 😥
범용성 ☹️

2.3 해결방안2 : Multi-clustered storage

NAVER
DEVIEW
2023



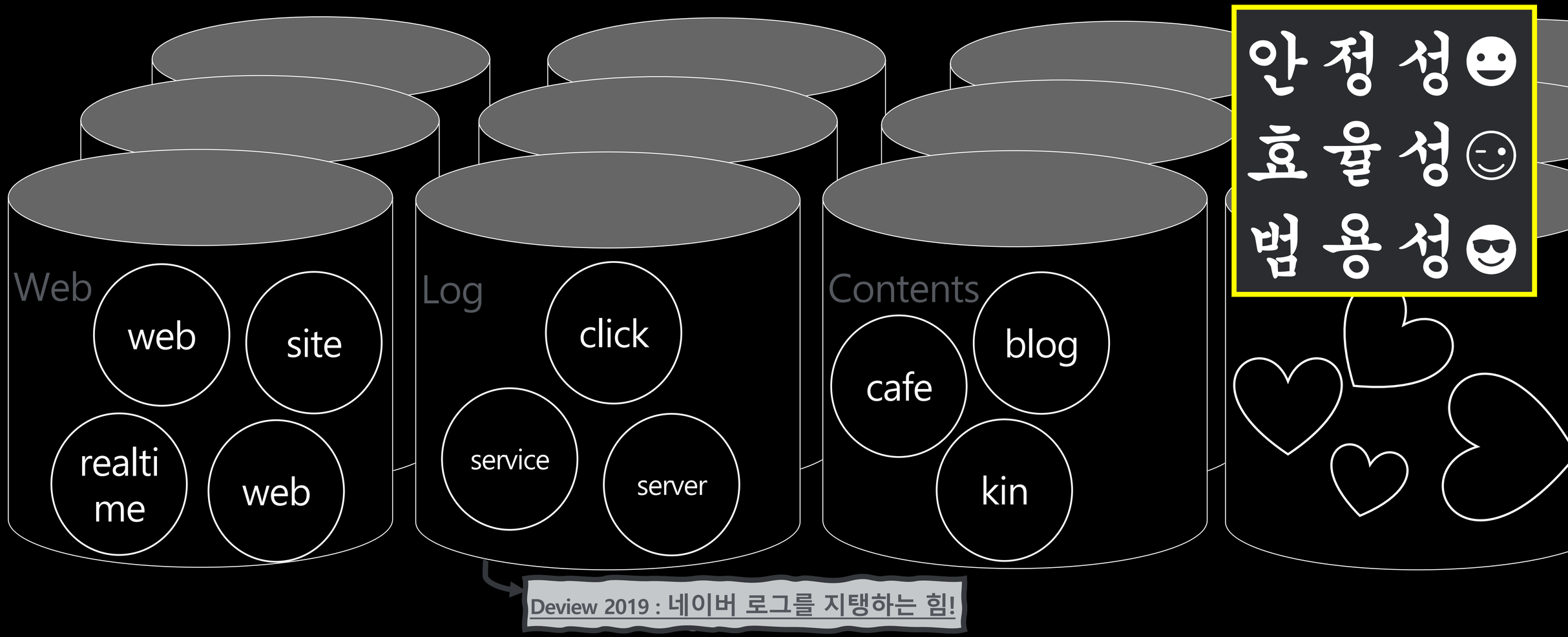
확실히 다른 데이터와 사용 패턴을
일정 수준 이상의 규모로 모아



클러스터를 구분하고 각각의 데이터에 맞게 최적화

2.3 해결방안2 : Multi-clustered storage

NAVER
DEVIEW
2023



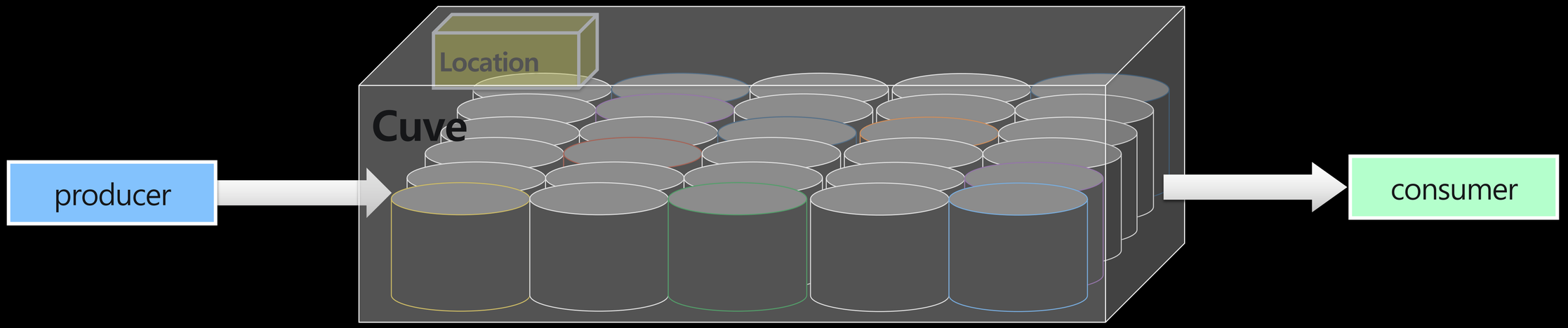
2.3 multi-clustering의 비용

NAVER
DEVIEW
2023

- one-cluster환경과 동일한 통합된 사용 경험을 제공
- 규모가 늘어날 수록 복잡도 또한 증가

사용성 ☺

개발/운영비용 😟



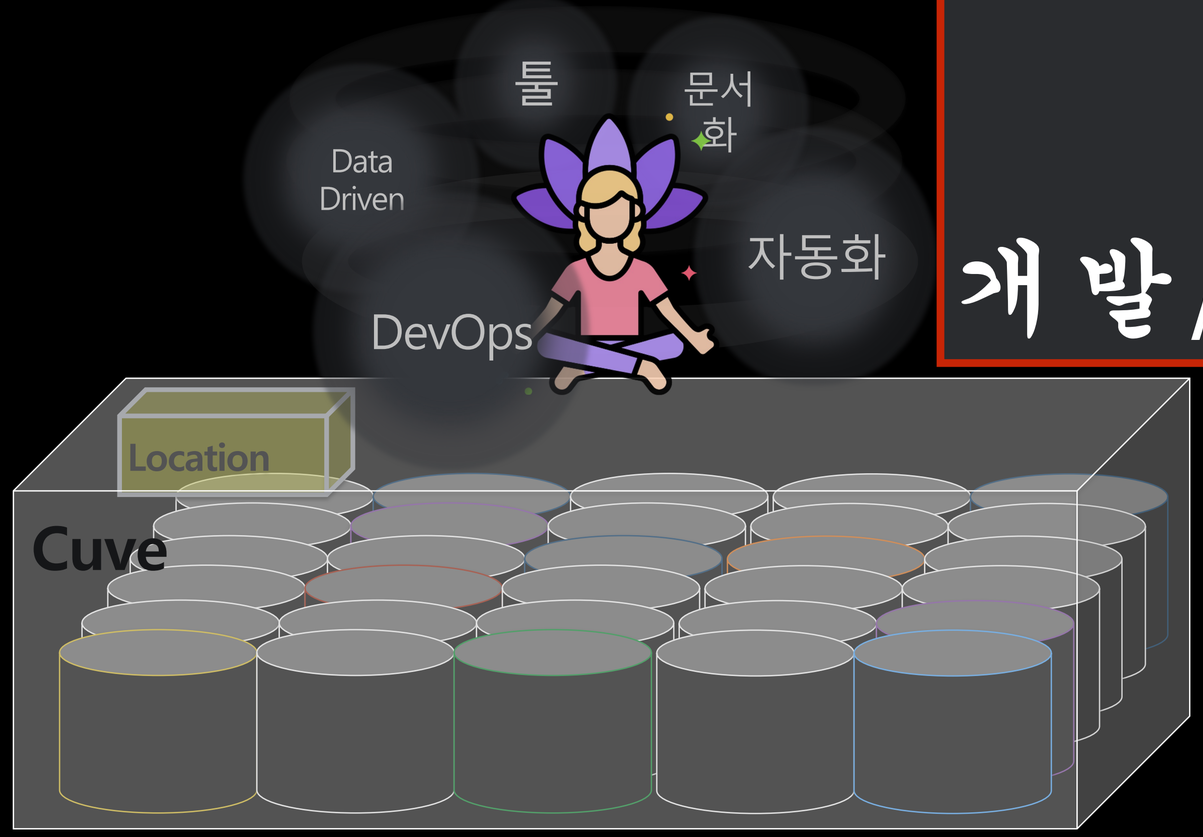
2.3 multi-clustering의 비용

NAVER
DEVIEW
2023

상이한 여러 클러스터들을 원활하게 운영해야 함

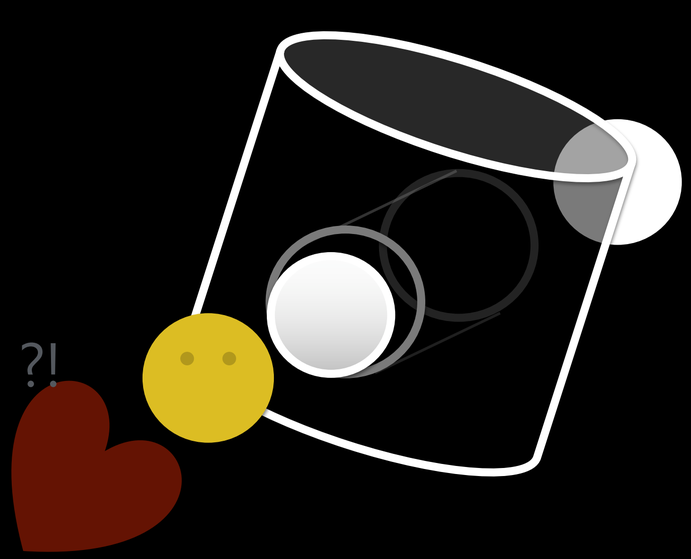
안정성 😈
효율성 😊
범용성 😎
사용성 😀

확장성 ☹️
지속성 😈
개발/운영비용 😣

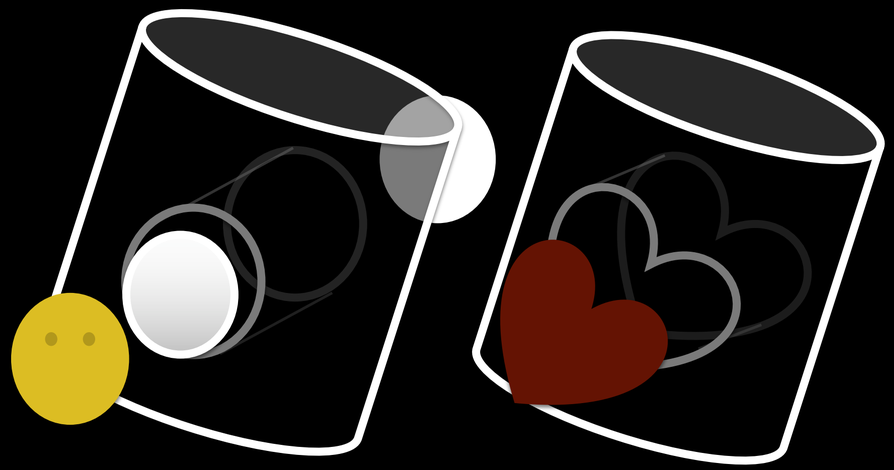


2.4 해결방안3 : HBase 데이터 유연성 개선

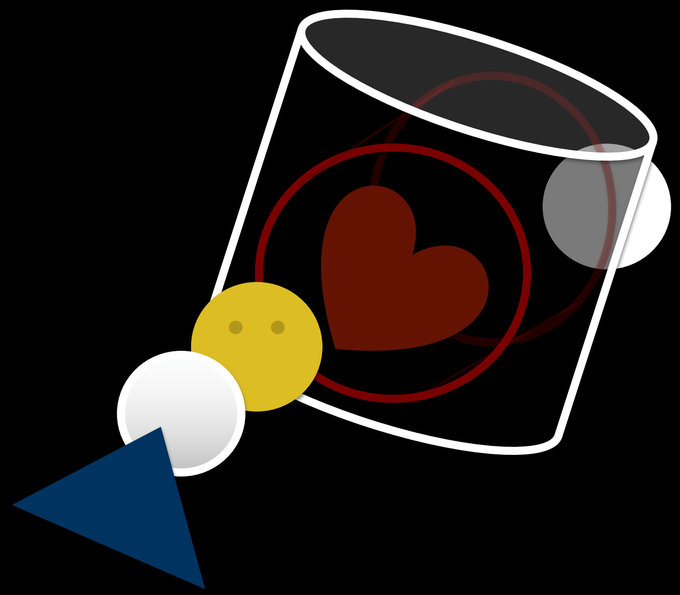
NAVER
DEVIEW
2023



제약



multi-clustering



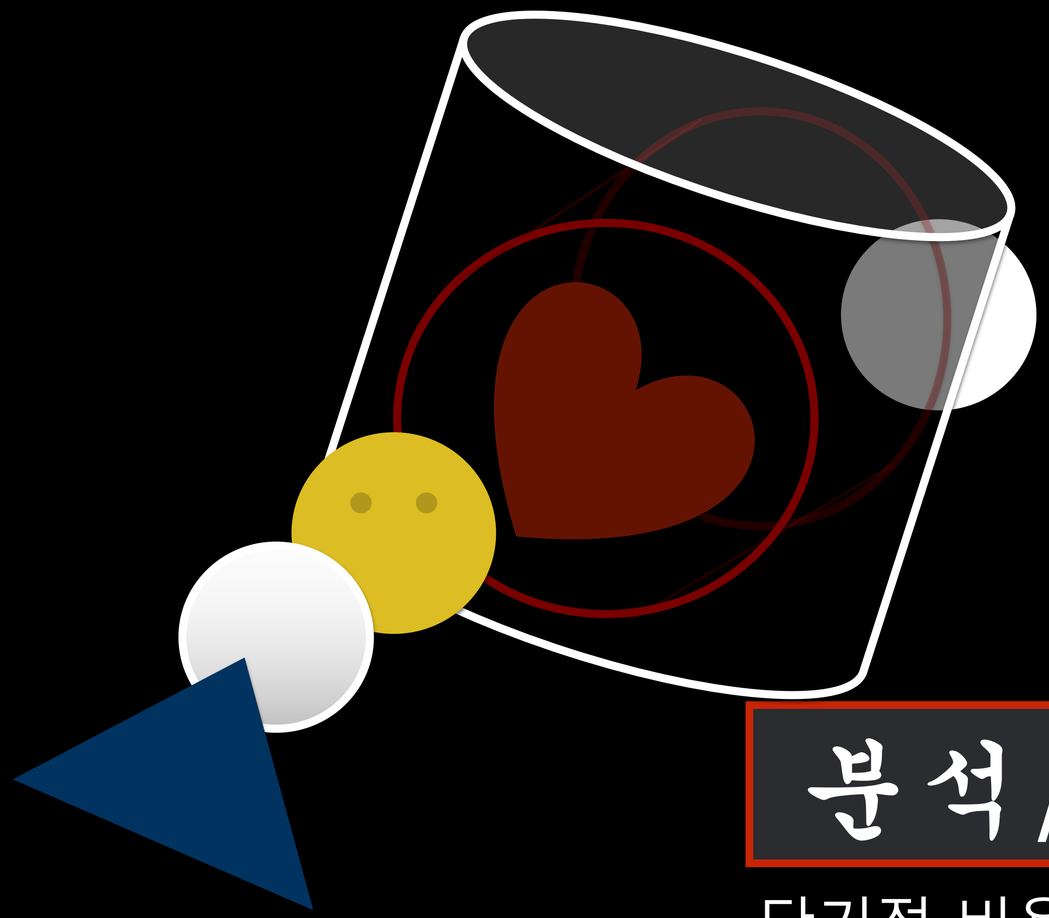
유연성 개선

2.4 해결방안3 : HBase 데이터 유연성 개선

NAVER
DEVIEW
2023

안정성
효율성
범용성
사용성
확장성
지속성

장기적 효과



분석/개발비용

단기적 비용

2.5 HBase random read amplification

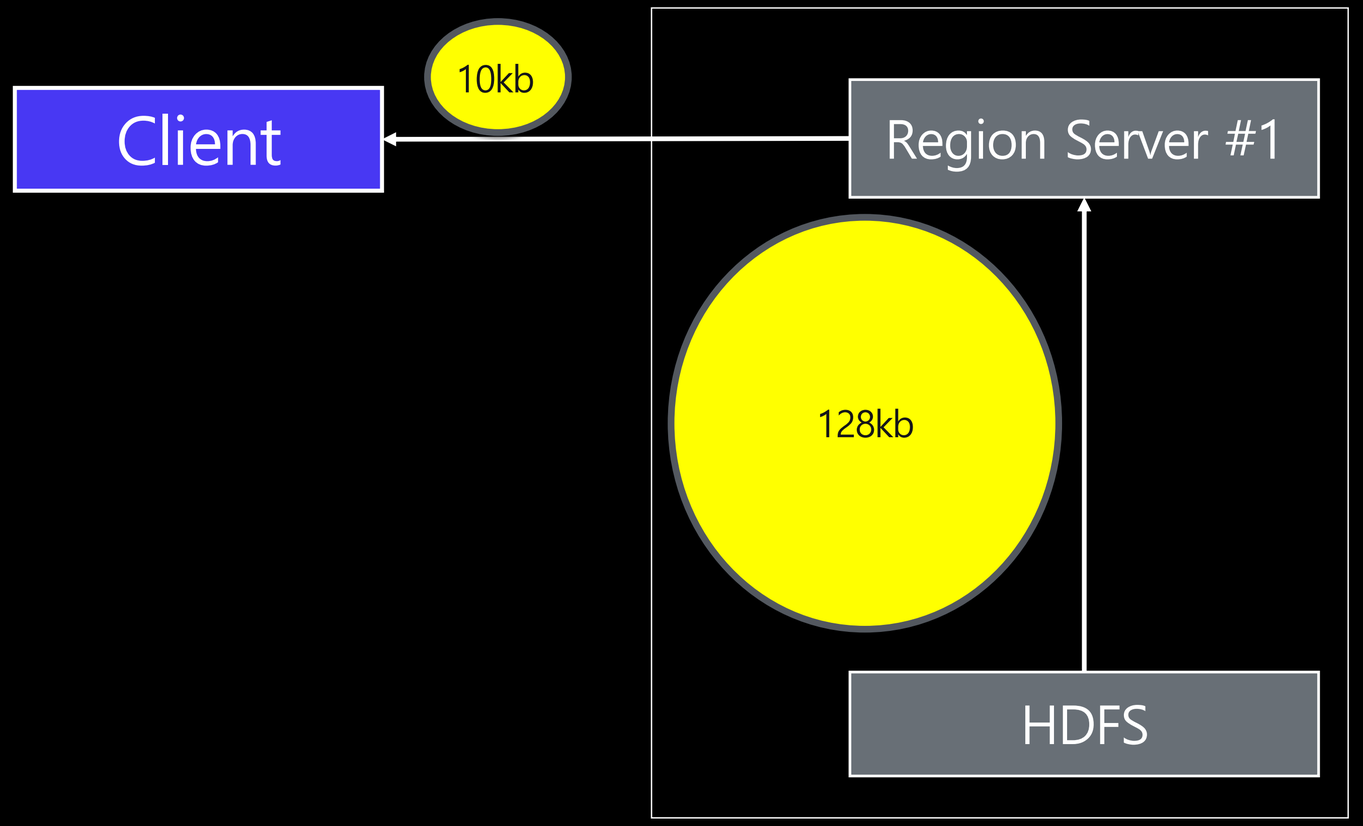
NAVER
DEVIEW
2023

HBase의 random read 과정에서

Client가 실제 쿼리한 data size 보다

증폭된 hdfs read 요청,

그 결과로 과도한 disk IO 사용

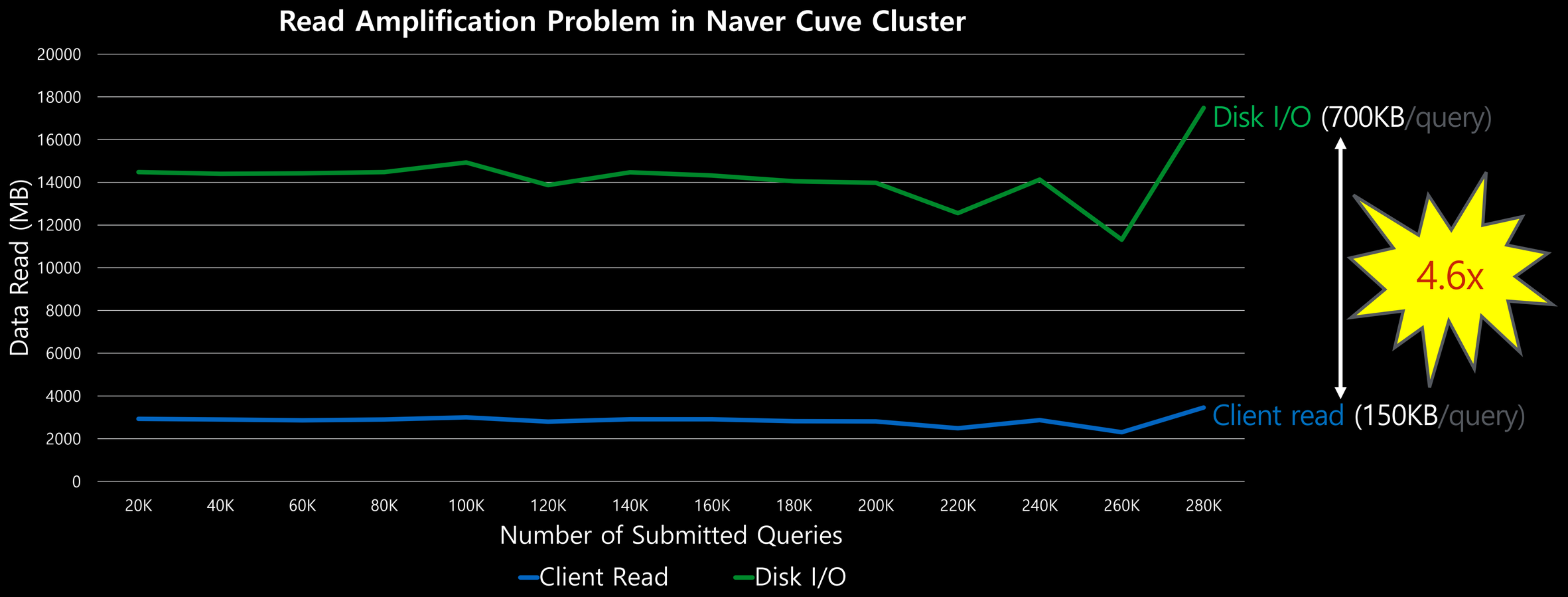


3. HBase read amplification

HBase의 read amplification 원인 분석

3.1 HBase read amplification:

NAVER DEVIEW 2023



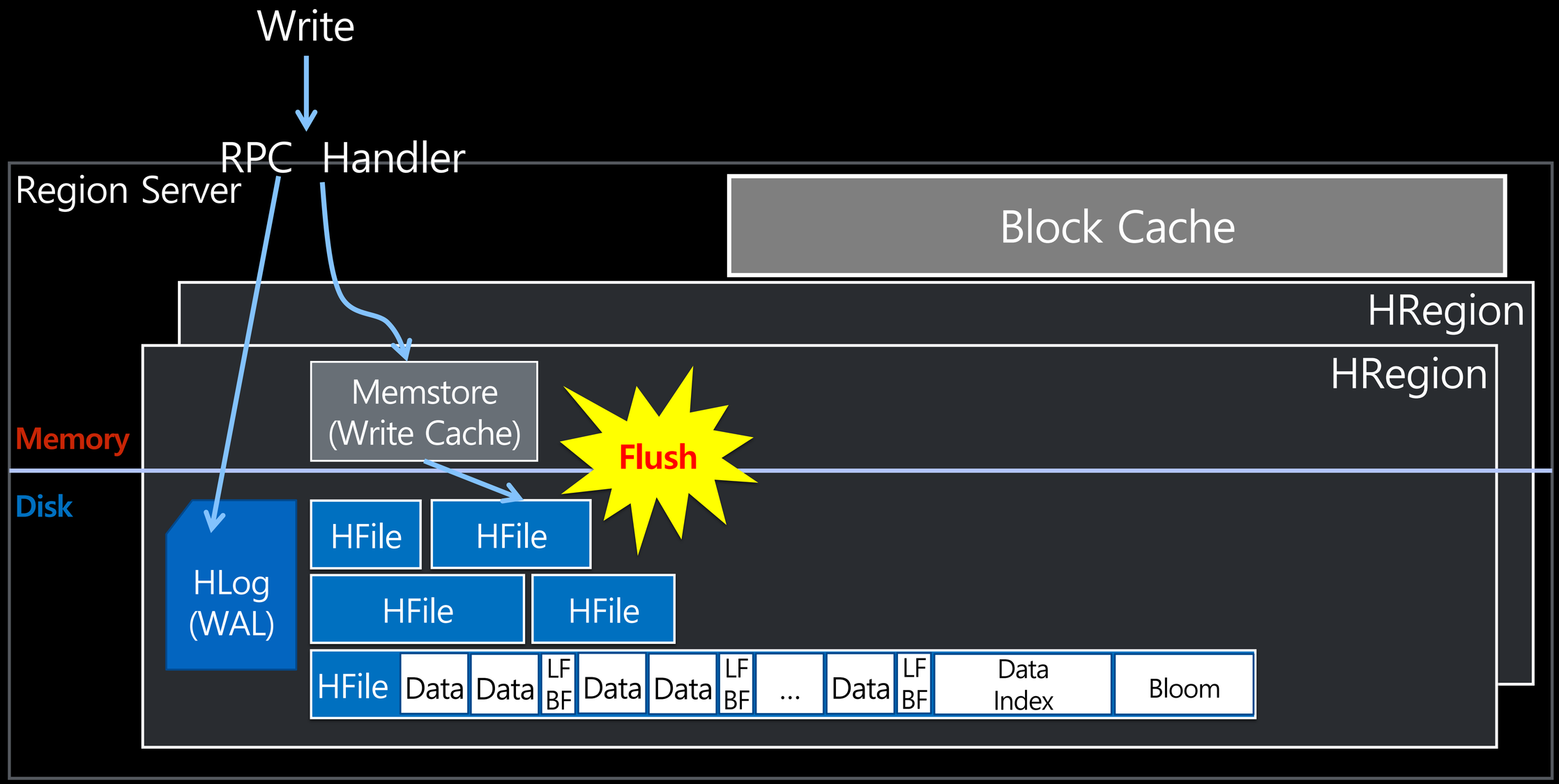
3.1 HBase read amplification:

NAVER
DEVIEW
2023

1. Multi File Access Amplification
2. Granularity Mismatch Amplification
3. Cache Pollution

3.2 Multi File Access Amplification: Write Process

NAVER DEVIEW 2023



3.2 Multi File Access Amplification: Validation

NAVER DEVIEW 2023

HBase Key Structure

Row Key	Column Family	Column Qualifier	Time Stamp

Client  RPC Handler  Scanner (**Initialization**)

3 stages of scanner initialization

- Region Scanner Initialization
- Store Scanner Initialization
- Store File Scanner Initialization

3.2 Multi File Access Amplification: Read Process

NAVER DEVIEW 2023

HBase Key Structure

Row Key	Column Family	Column Qualifier	Time Stamp

Client ⟷ RPC Handler ⟷ **Region Scanner Initialization**

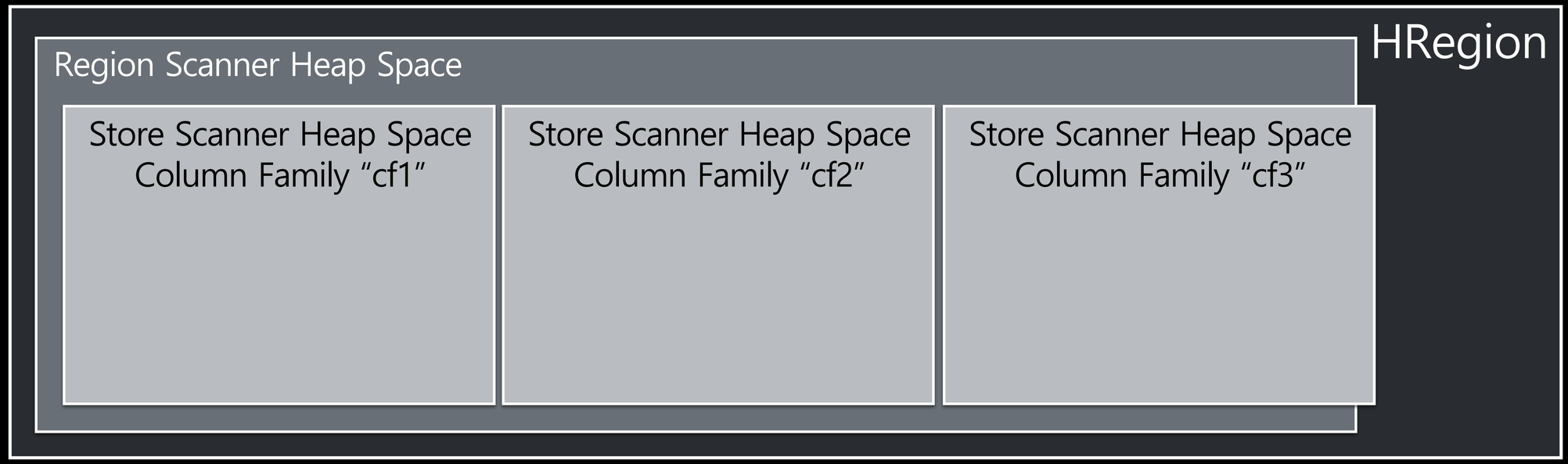
3.2 Multi File Access Amplification: Read Process

NAVER DEVIEW 2023

HBase Key Structure

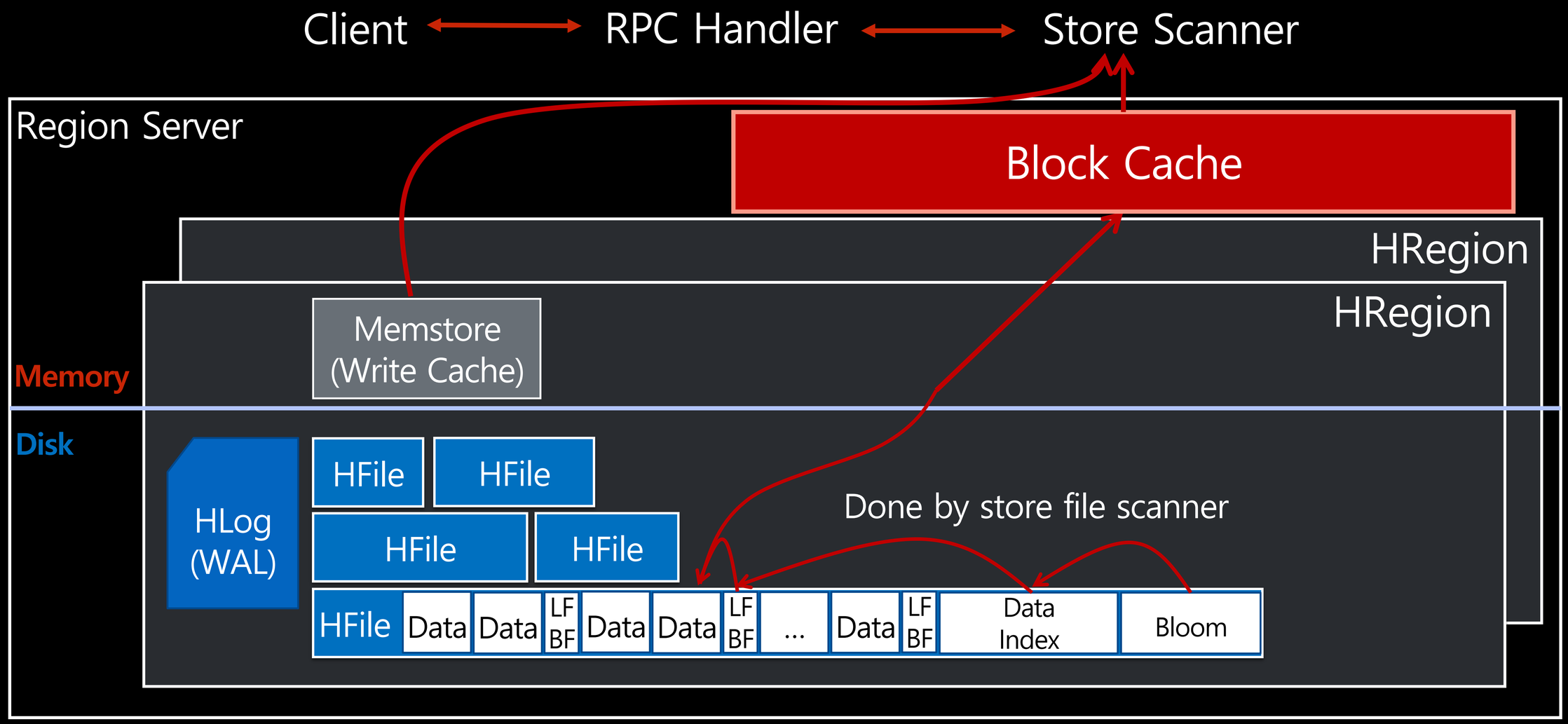
Row Key	Column Family	Column Qualifier	Time Stamp

Client ⟷ RPC Handler ⟷ **Store Scanner Initialization**



3.2 Multi File Access Amplification: Read Process

NAVER DEVIEW 2023



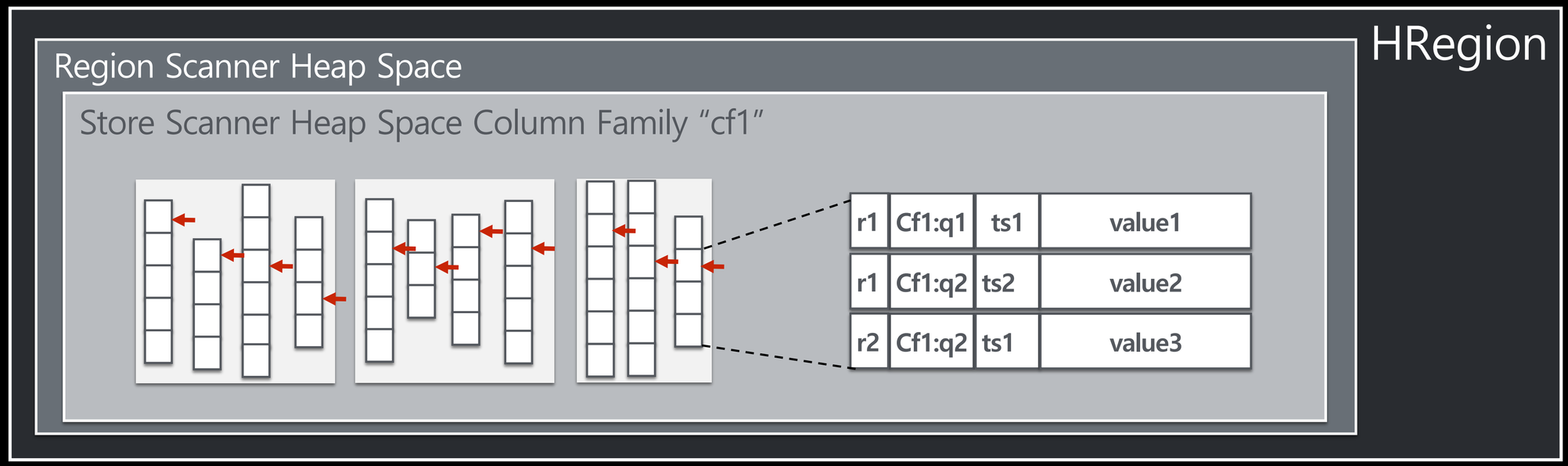
3.2 Multi File Access Amplification: Read Process

NAVER DEVIEW 2023

HBase Key Structure

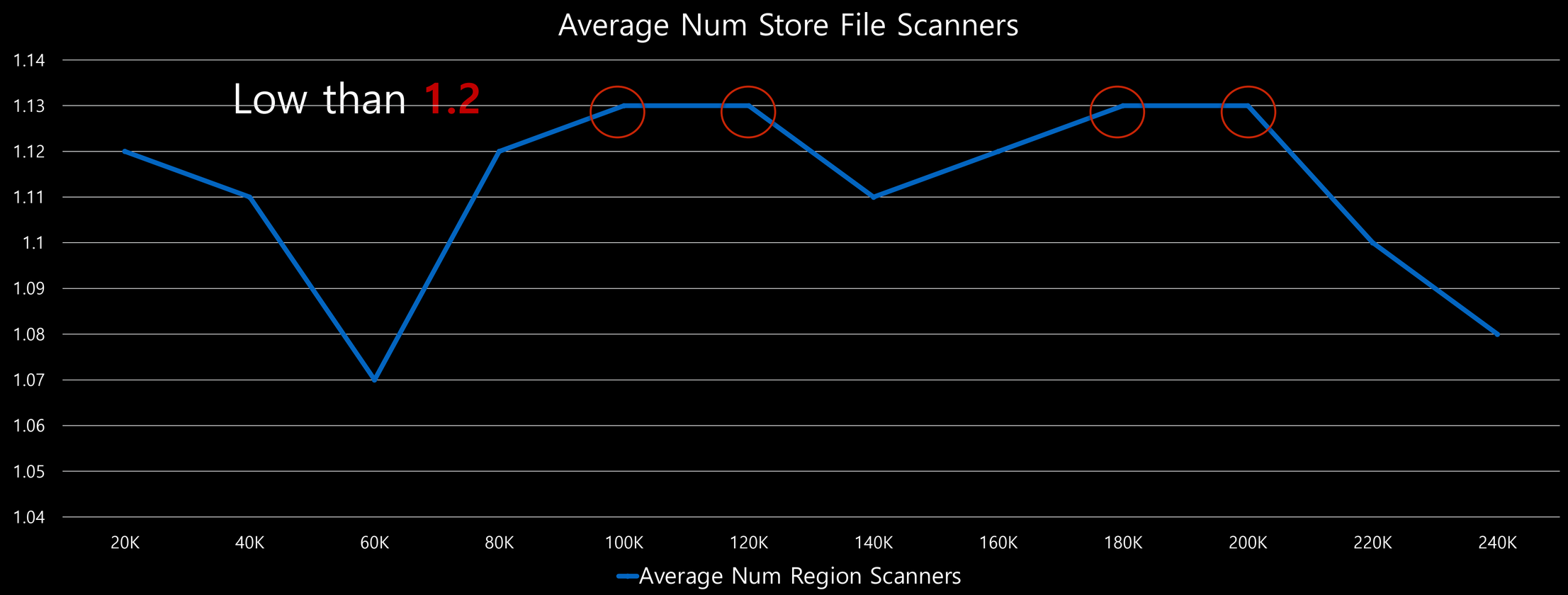
Row Key	Column Family	Column Qualifier	Time Stamp

Client ⟷ RPC Handler ⟷ **Initialization Terminate**



3.2 Multi File Access Amplification: Validation

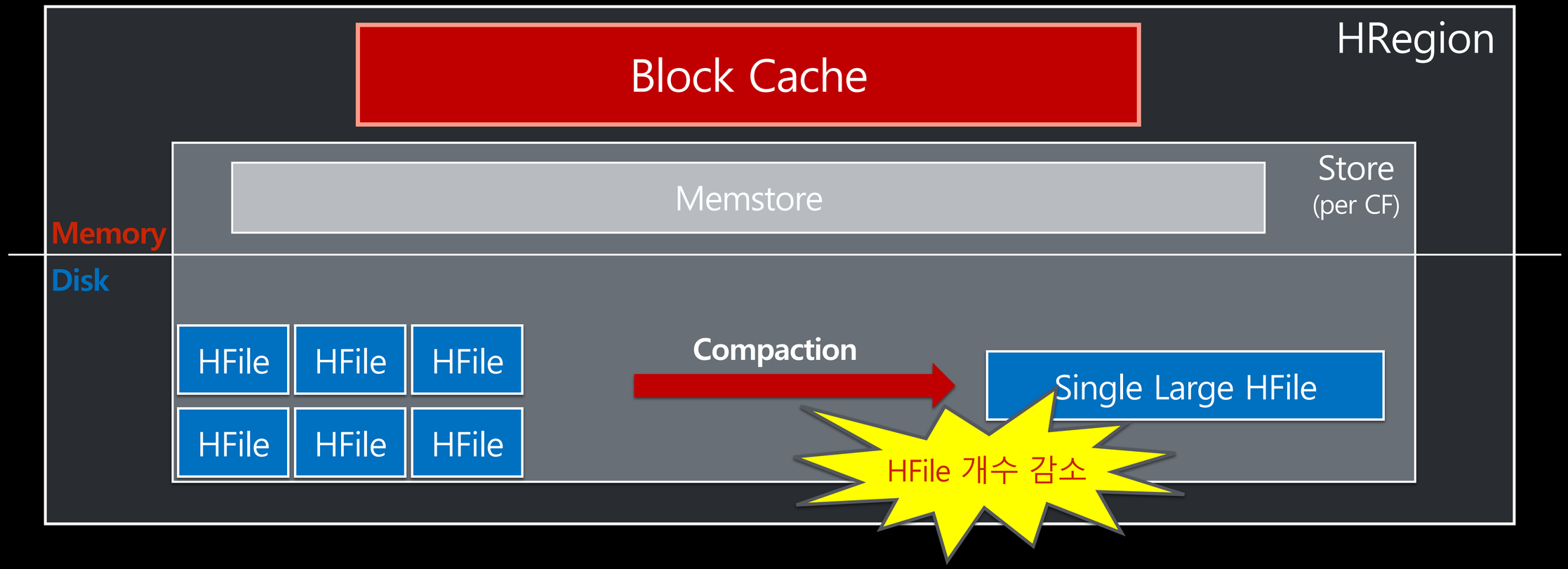
NAVER DEVIEW 2023



3.2 Multi File Access Amplification:
Compaction을 통한 HFile의 감소

NAVER DEVIEW 2023

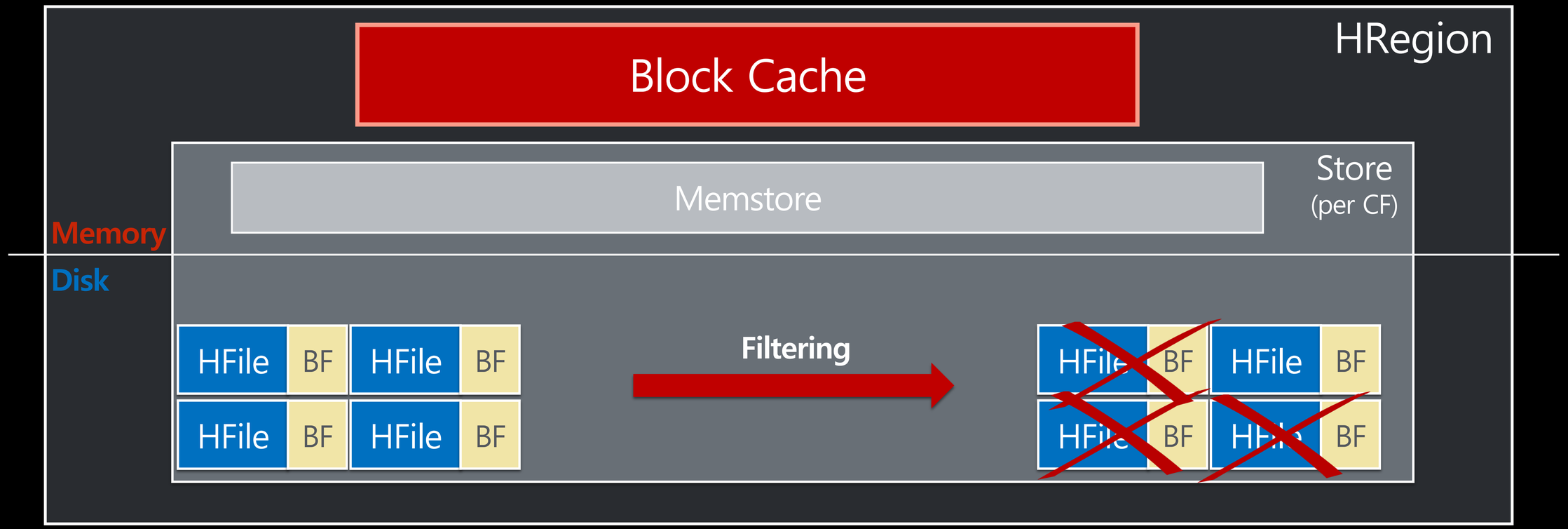
Compaction: 여러 개의 HFile의 하나의 HFile로 rewrite하는 과정



3.2 Multi File Access Amplification:
Bloom Filter로 인해 접근하는 HFile수 감소

NAVER DEVIEW 2023

Bloom Filter:



3.2 Multi File Access Amplification: if?

NAVER
DEVIEW
2023

1. Check Compaction Status: compaction throughput, compaction thread...
2. Check Flush Status: memstore size, flush threshold size...
3. Check Bloom Filtering Status: bloom filter type, bloom filter size...

3.3 HBase read amplification:

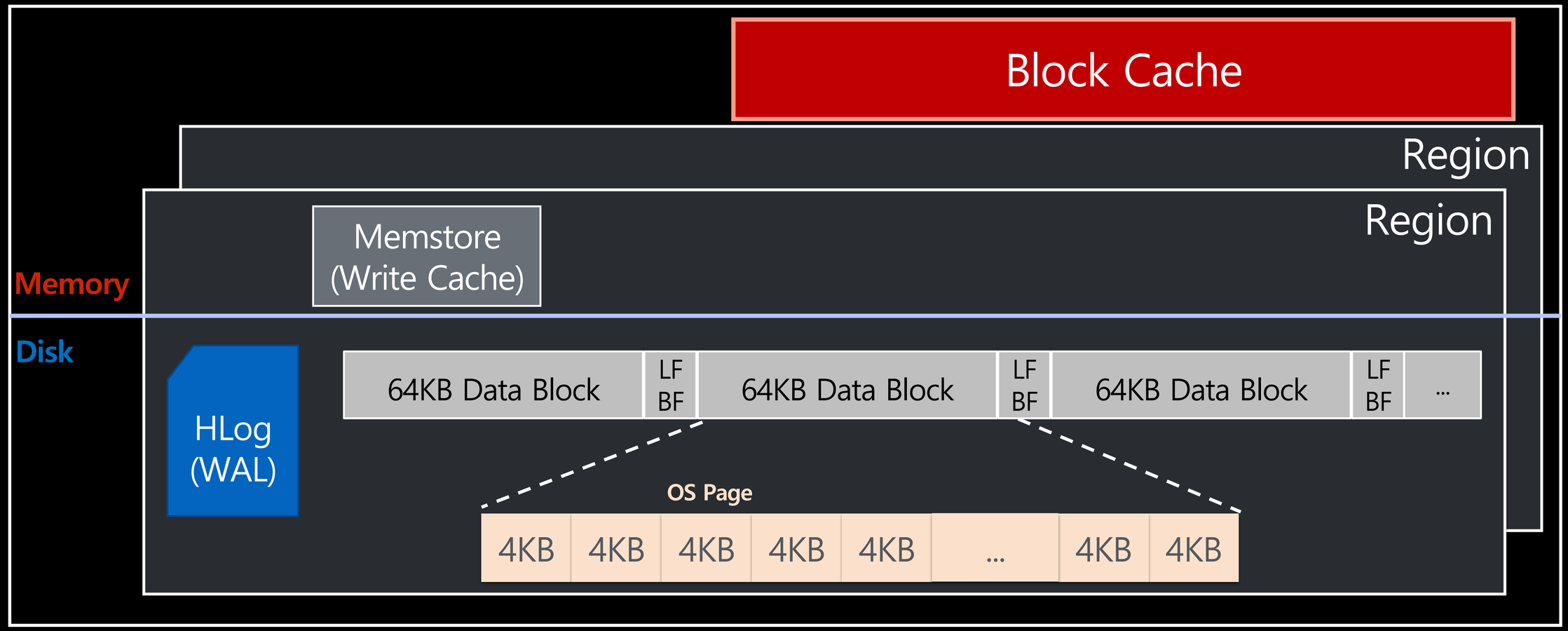
NAVER
DEVIEW
2023

1. ~~Multi File Access Amplification~~
2. Granularity Mismatch Amplification
3. Cache Pollution

3.3 Granularity Mismatch Amplification

: Application request data size differ from I/O data size

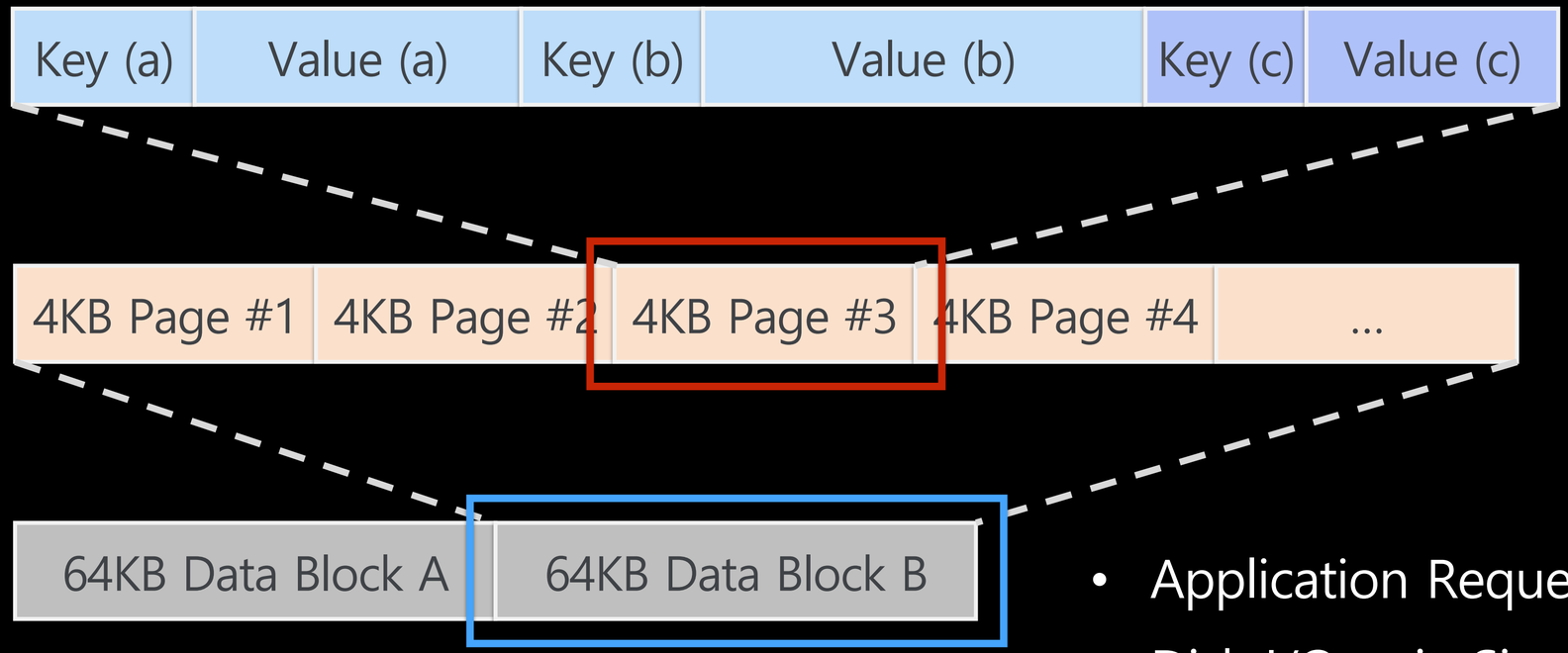
NAVER DEVIEW 2023



3.3 Granularity Mismatch Amplification

: Application request data size differ from I/O data size

NAVER DEVIEW 2023



- Application Request: Single OS Page (4KB Page #3)
- Disk I/O unit: Single Data Block (64KB Data Block B)
- Theoretically, **16x** amplification occur

3.4 HBase read amplification:

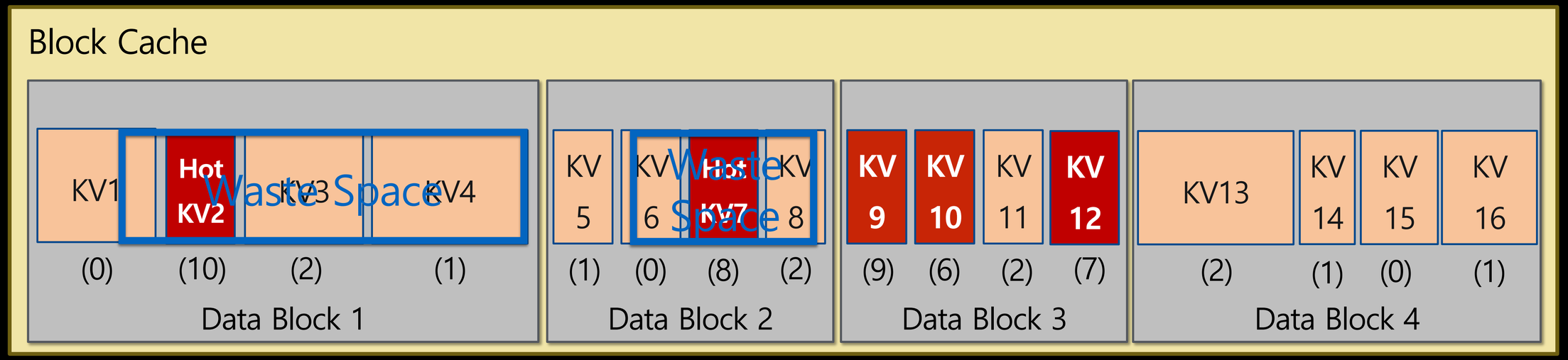
NAVER
DEVIEW
2023

1. ~~Multi File Access Amplification~~
2. Granularity Mismatch Amplification
3. Cache Pollution

3.4 Cache Pollution Problem

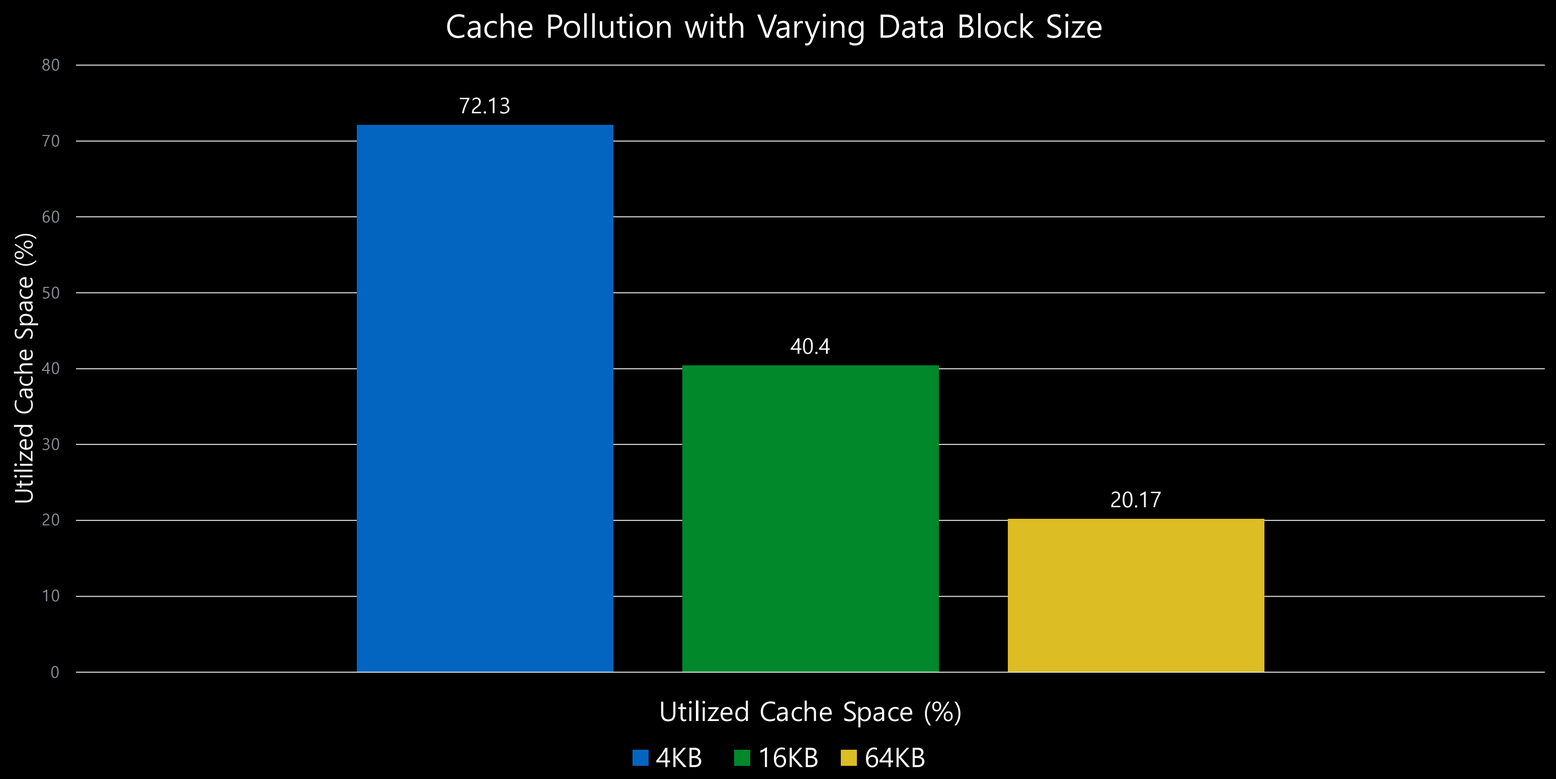
NAVER DEVIEW 2023

Data Block unit LRU Replacement



3.4 Cache Pollution Problem

NAVER DEVIEW 2023



- 4KB Data Block: 72.13%
- 16KB Data Block: 40.4%
- 64KB Data Block: 20.17%

64KB Data Block = 72.03% under utilization than 4KB Data Block

3.5 Naïve solution: Small Data Block Size

NAVER
DEVIEW
2023

1. Decrease space locality

- Low space locality leads frequent random read
- Frequent random read increase the number of disk seek operation

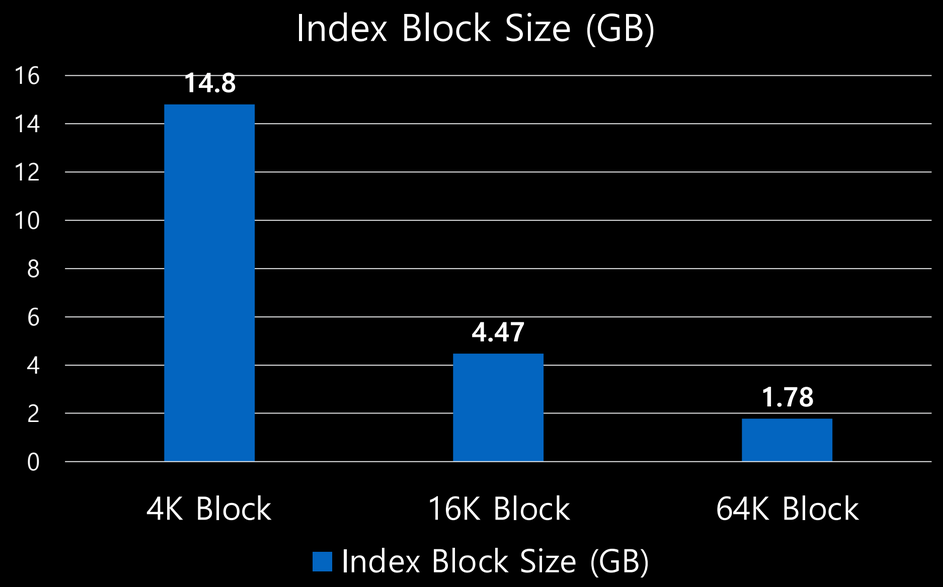
2. Increase block encoding cost

- Decrease Block encoding & compression efficiency

3.5 Naïve solution: Small Data Block Size

NAVER DEVIEW 2023

3. Index Block size grow

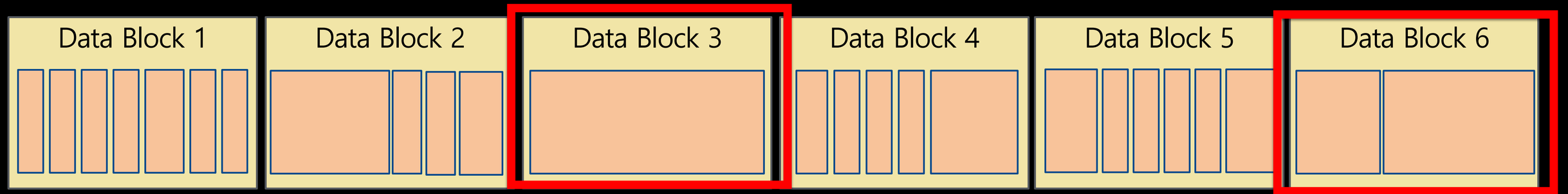


- 4KB Data Block Index size = **14.8 GB** (per RS)
- 64KB Data Block Index size = **1.78 GB** (per RS)
- 4KB DB Index size = **8.3** X 64KB DB Index size

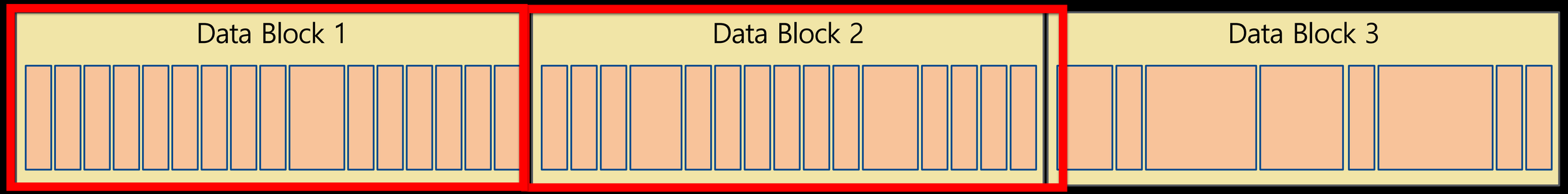
3.5 Naïve solution: 적절한 크기의 data block?

NAVER DEVIEW 2023

Case of small data block size



Case of big data block size



3.6 New Approach

NAVER DEVIEW 2023

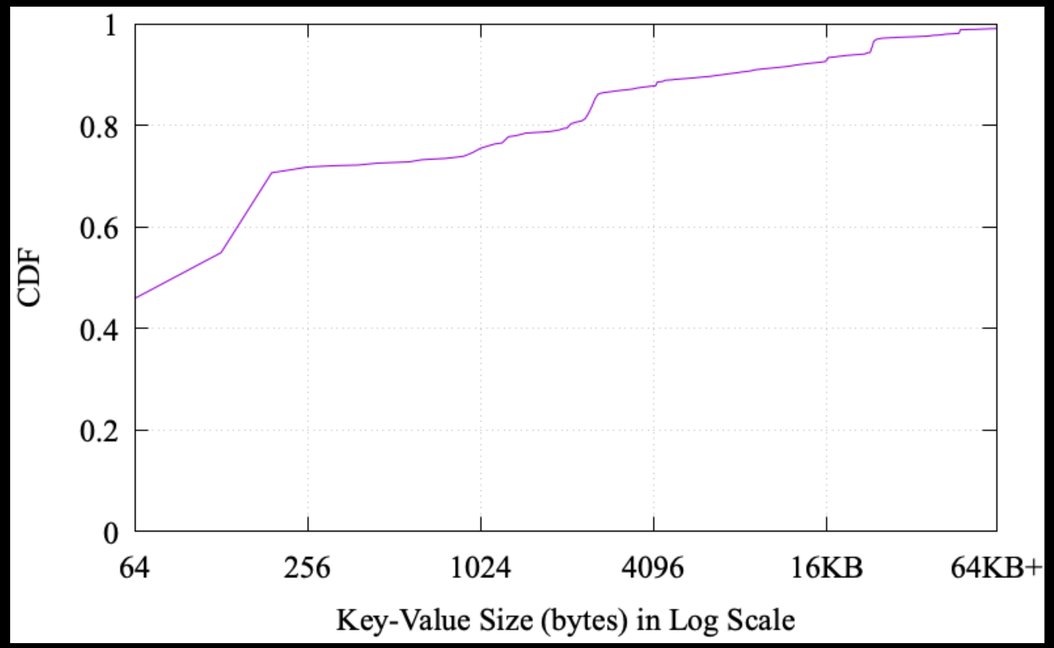


Figure: Data Block Size Distribution in Naver's HBase Cluster

Various key-value size in Naver Cuve cluster's

- 80% of key-values smaller than 4KB
- 10% of key-values larger than 16KB

Fixed-size data block cannot handle various size of KV

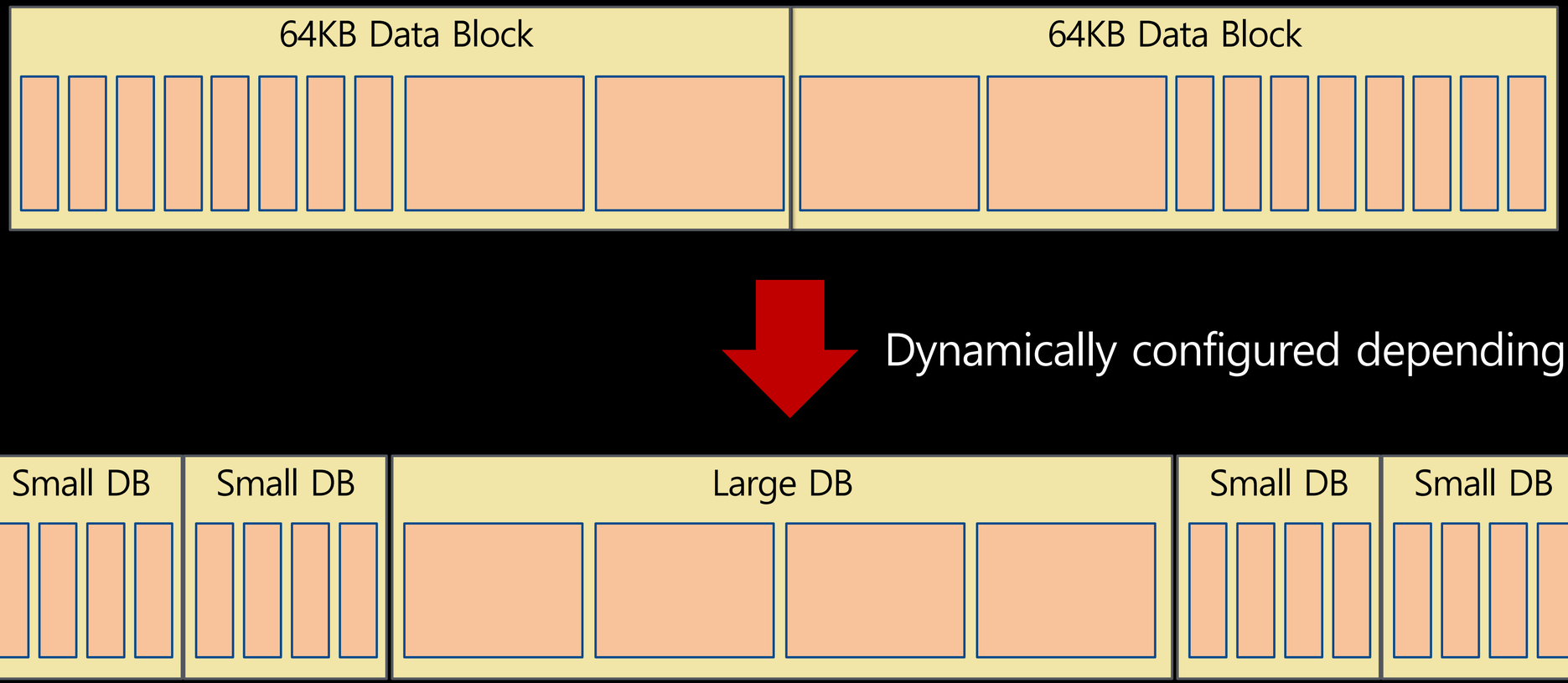
- Small KV benefit from small size data block
- Large KV benefit from large size data block

4. Solution

HBase 개선 방안

4.1 Solution 1: Dynamic Variable Block Size Control

NAVER DEVIEW 2023



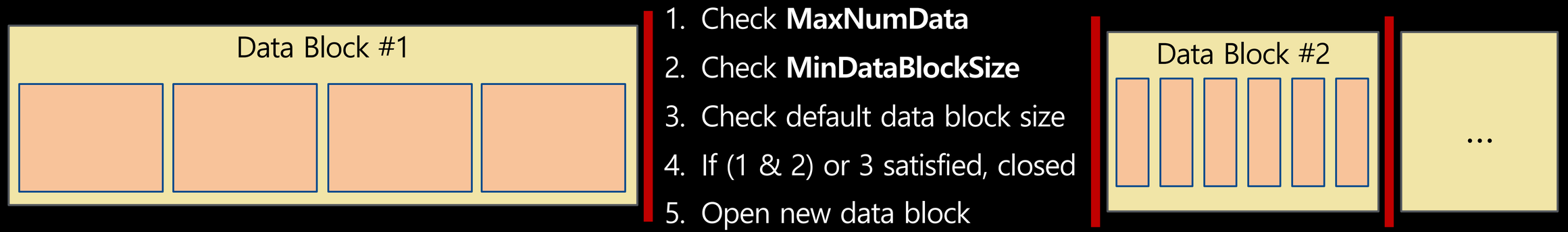
4.1 Solution 1: Dynamic Variable Block Size Control

Idea: Dynamically configured depending on KV Size

NAVER DEVIEW 2023

Add two more condition in vanilla HBase

- **MaxNumData:** Upper bound of Key-Value entry in single data block :
  - To store small Key-Value in small data block to mitigate read amplification problem
- **MinDataBlockSize:** Lower bound of data block size
  - Keep the data blocks from getting too small to get benefit from spatial locality

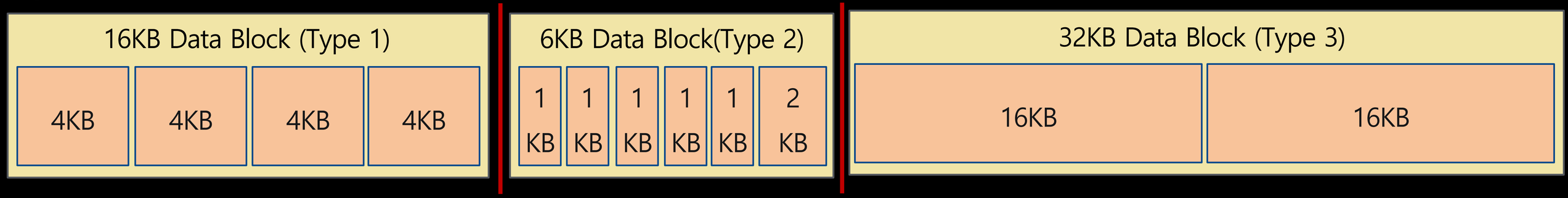


4.1 Solution 1: Dynamic Variable Block Size Control

Idea: Dynamically configured depending on KV Size

NAVER DEVIEW 2023

e.g., MaxNumData=4, MinDataBlockSize=6KB, Default Data Block Size=32KB



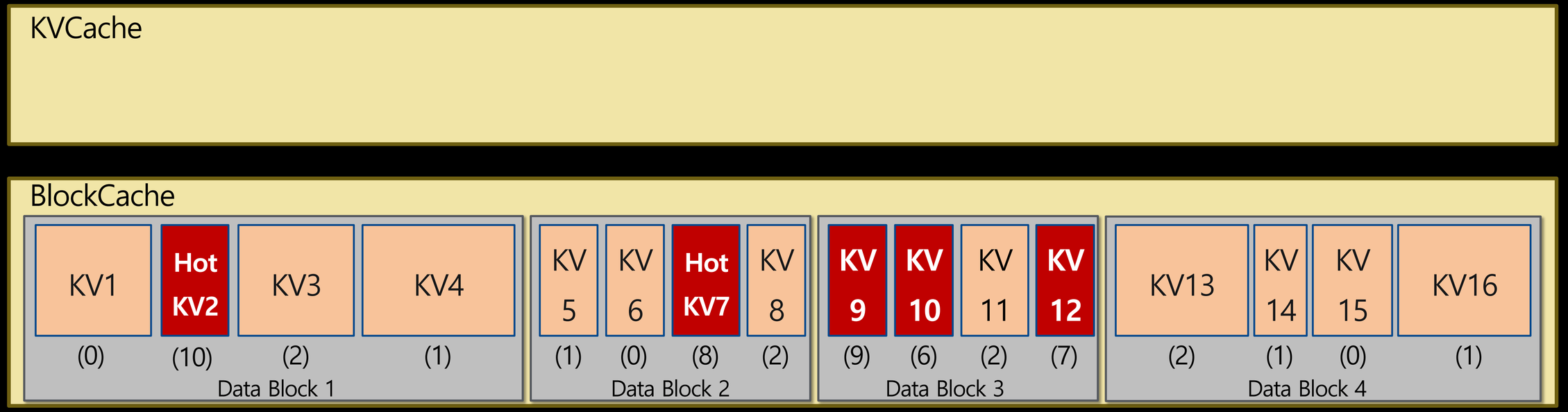
Data Block Type

1. Type 1: Closed by MaxNumData Condition
2. Type 2: Closed by MinDataBlockSize Condition
3. Type 3: Closed by Default Data Block Size

4.2 Solution 2: KV Cache

Idea: Promoting hot KV based on space utilization

NAVER DEVIEW 2023

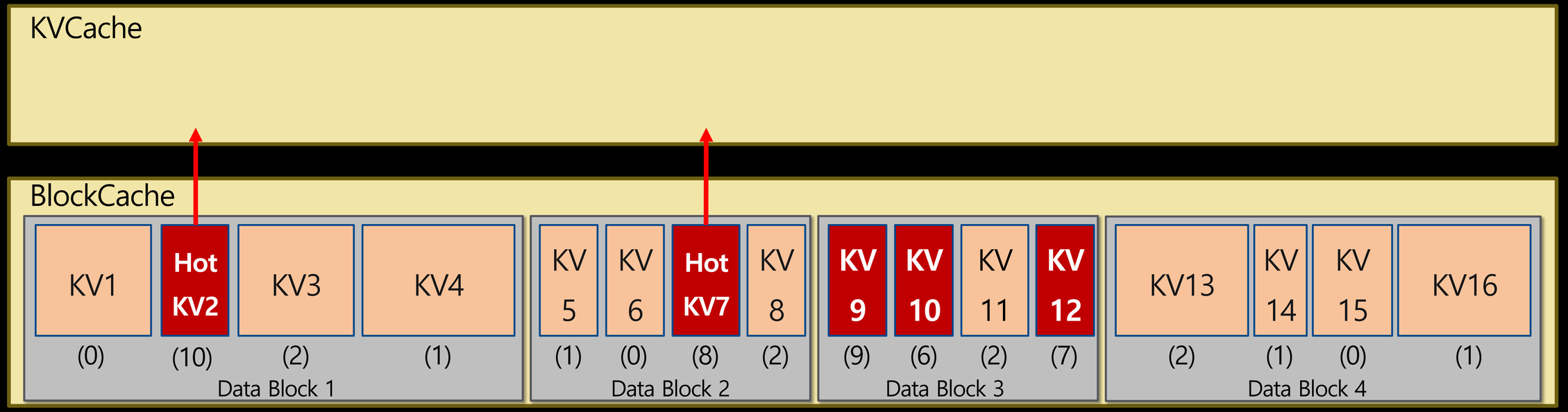


- KV2 and KV7 are promoted to KVCache due to **low cache space utilization**
- KV9, KV10 and KV12 are not promoted to KVCache due to **high cache space utilization**

4.2 Solution 2: KV Cache

Idea: Promoting hot KV based on space utilization

NAVER DEVIEW 2023

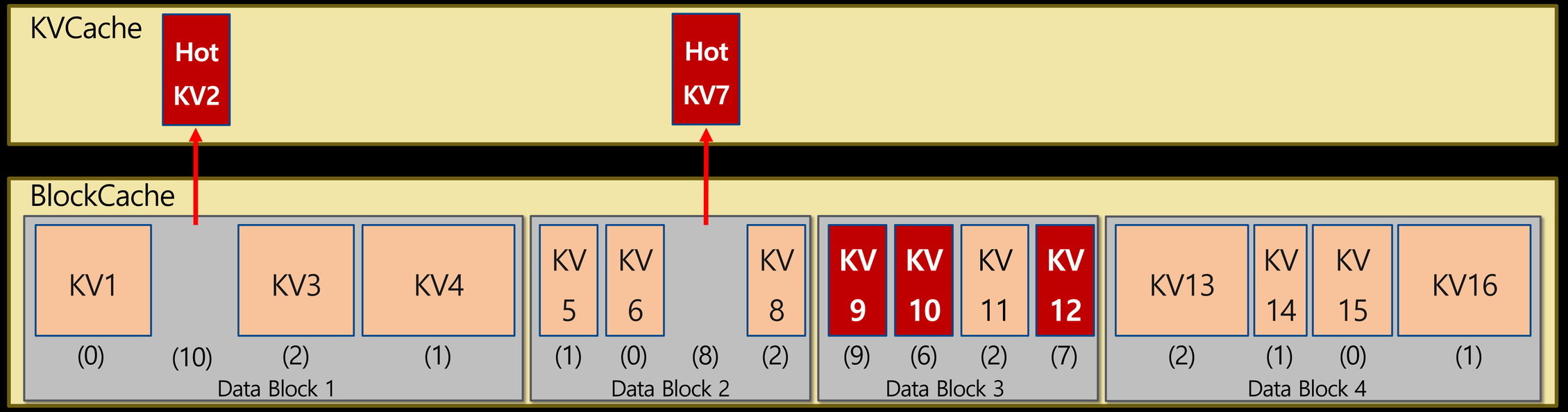


- KV2 and KV7 are promoted to KVCache due to **low cache space utilization**
- KV9, KV10 and KV12 are not promoted to KVCache due to **high cache space utilization**

4.2 Solution 2: KV Cache

Idea: Promoting hot KV based on space utilization

NAVER DEVIEW 2023

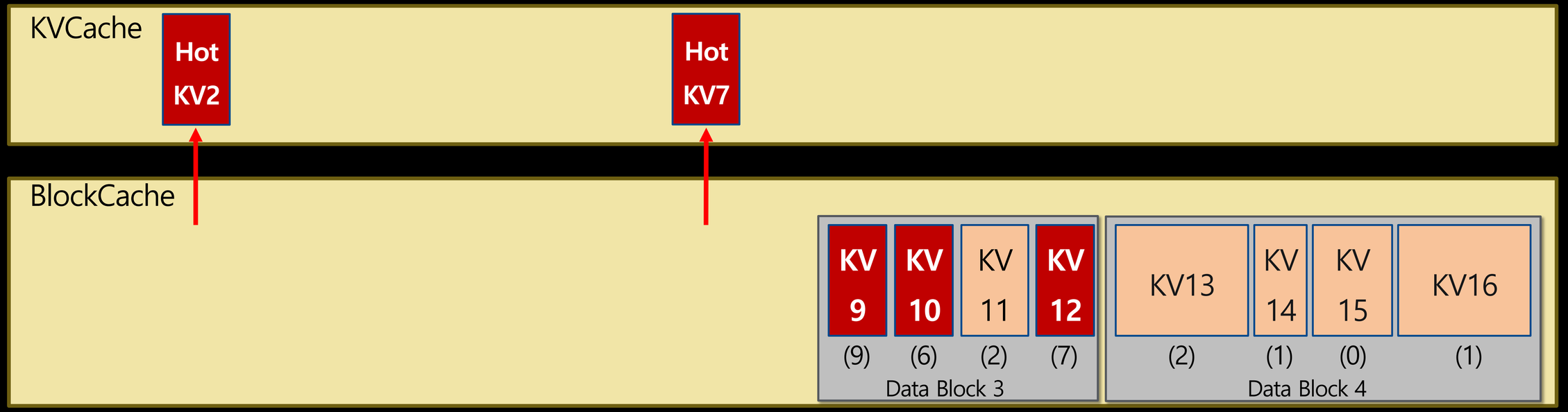


- KV2 and KV7 are promoted to KVCache due to **low cache space utilization**
- KV9, KV10 and KV12 are not promoted to KVCache due to **high cache space utilization**

4.2 Solution 2: KV Cache

Idea: Promoting hot KV based on space utilization

NAVER DEVIEW 2023



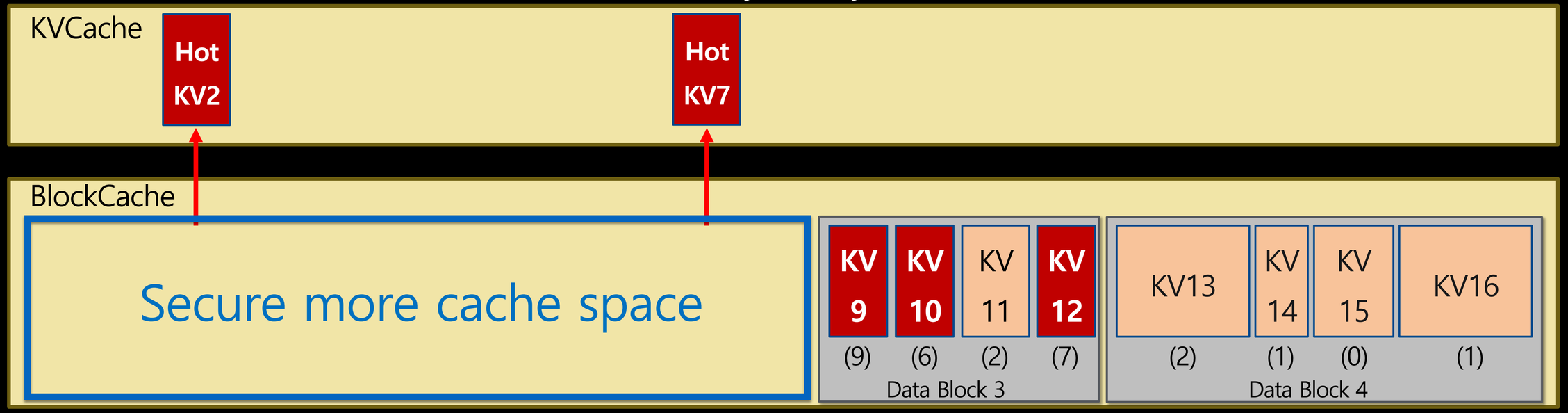
- KV2 and KV7 are promoted to KVCache due to **low cache space utilization**
- KV9, KV10 and KV12 are not promoted to KVCache due to **high cache space utilization**

4.2 Solution 2: KV Cache

Idea: Promoting hot KV based on space utilization

NAVER DEVIEW 2023

$$Promotion\ Threshold = Access\ Count[key\ entry] > Mean + Standard\ Deviation$$



- KV2 and KV7 are promoted to KVCache due to **low cache space utilization**
- KV9, KV10 and KV12 are not promoted to KVCache due to **high cache space utilization**

4.2 Solution 2: KV Cache

Cache Replacement Policy

NAVER DEVIEW 2023

Replacement Matric in KVCache

- **Last Access Time :** Due to promotion from Block Cache
- **Number of Access :** Due to relative access frequency of neighboring key-values
- **Key-Value Size :** Due to various key-value size

Replacement Formula

$$W = \frac{Normalized\ Access\ Frequency^{f}}{Data\ Size^{d} \times (Current\ Time\ - Last\ Access\ Time)}$$

5. Evaluation

벤치 마킹 및 적용 효과

5.1 Evaluation 용어 설명

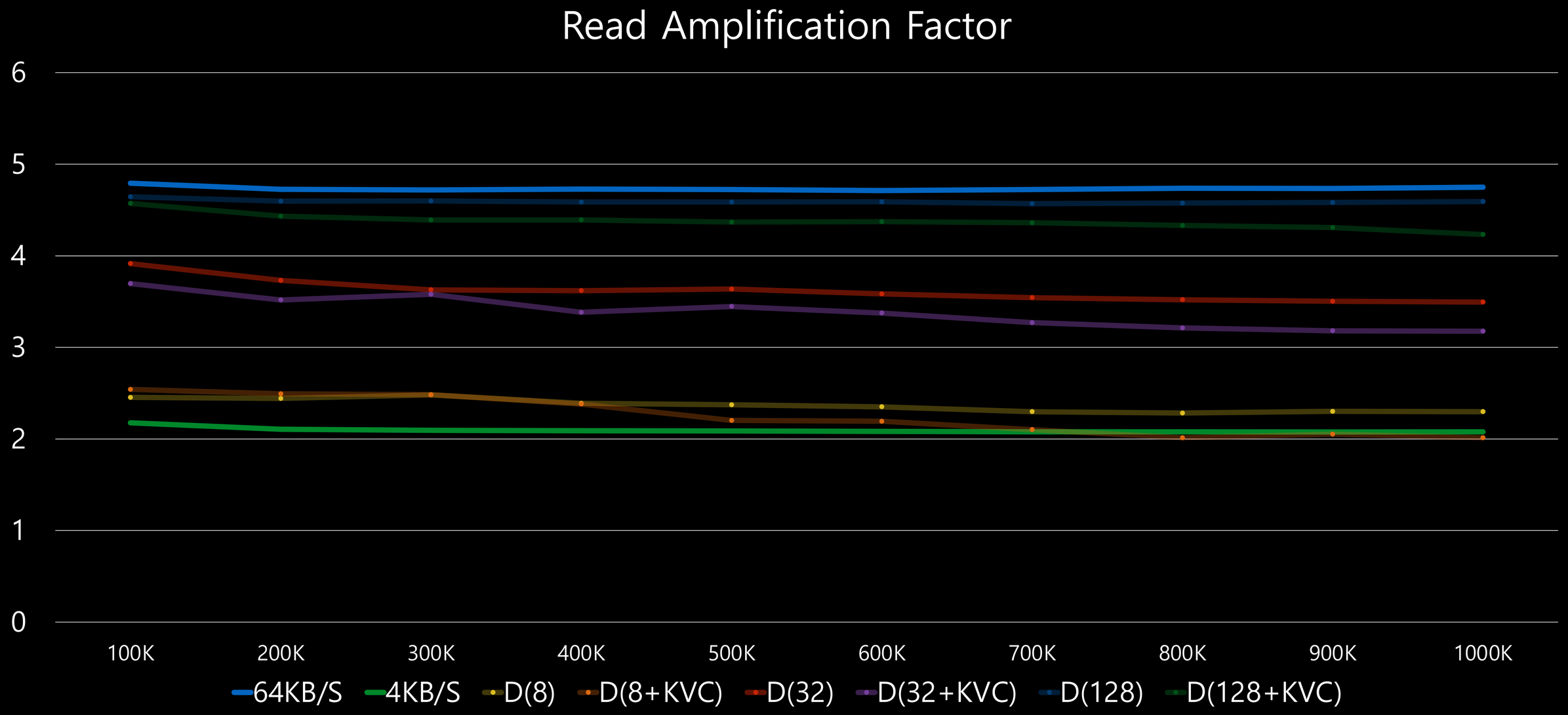
NAVER
DEVIEW
2023

- Synthetic Workload CBMG (Customer Behavior Model Graph)
- Modeling workload based on spatial-temporal customer behavior
- Apply CBMG workload in CUVE based on URL similarity

- Read Amplification Factor
- Application Request #Page / IO Request #Page
- RAF proportional to Granularity Mismatch Amplification

5.1 CUVE: CBMG Benchmark : Read Amplification Factor

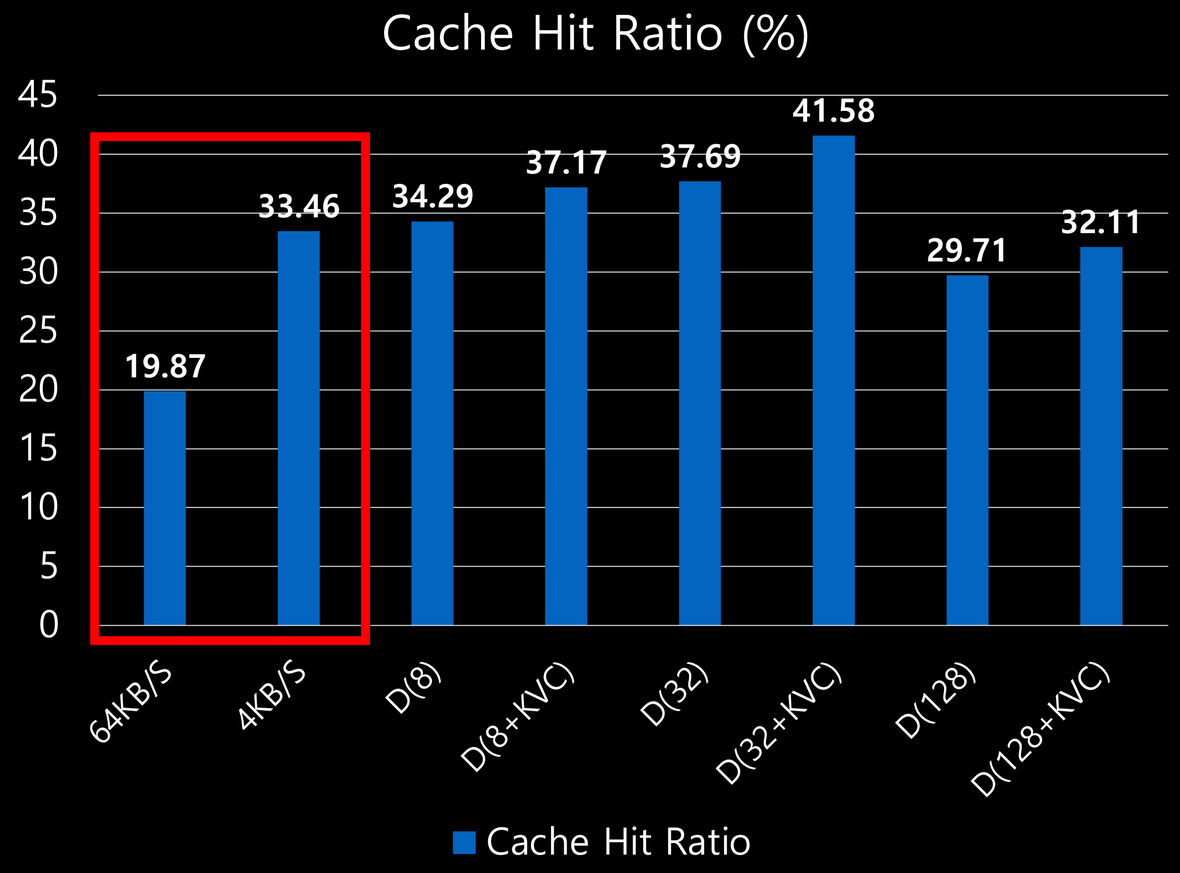
NAVER DEVIEW 2023

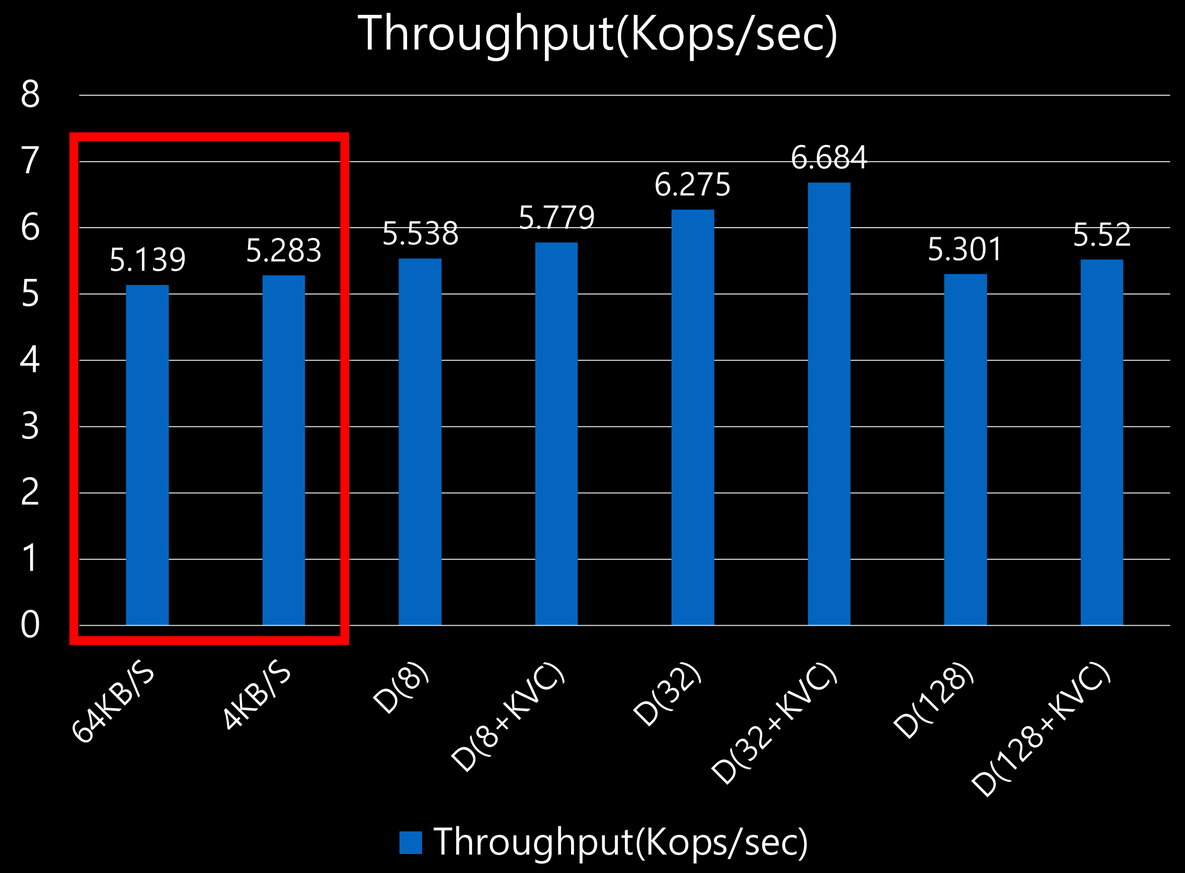


5.2 CUVE: CBMG Benchmark :

Cache Hit Ratio & Throughput

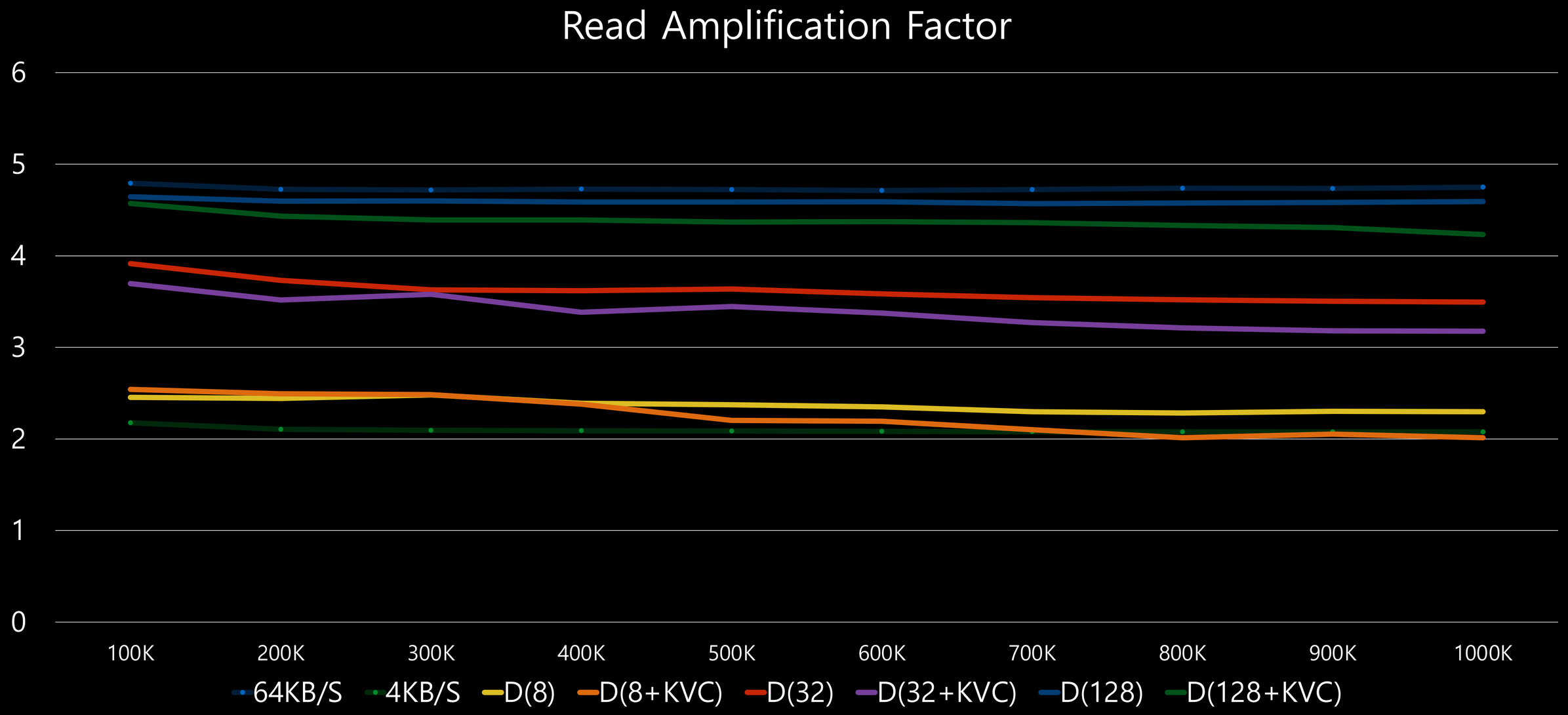
NAVER DEVIEW 2023





5.1 CUVE: CBMG Benchmark : Read Amplification Factor

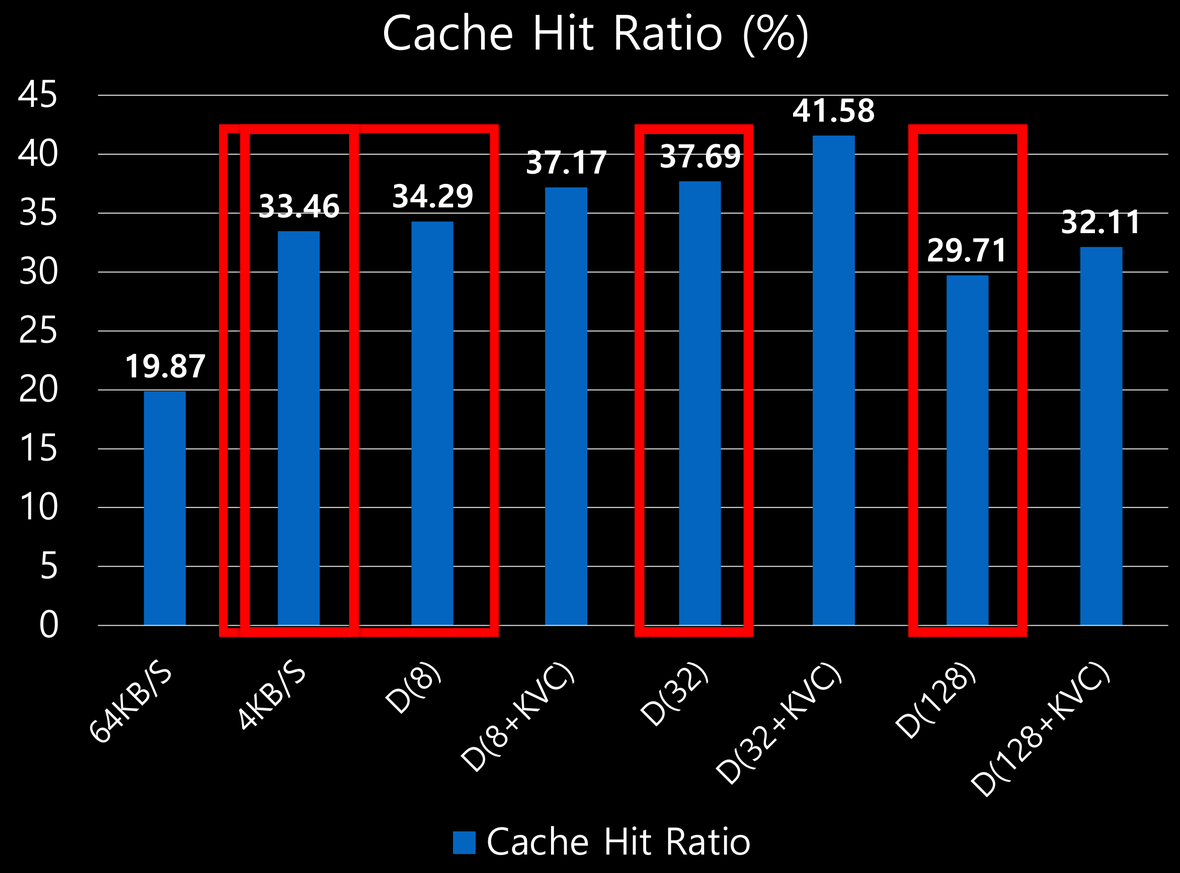
NAVER DEVIEW 2023

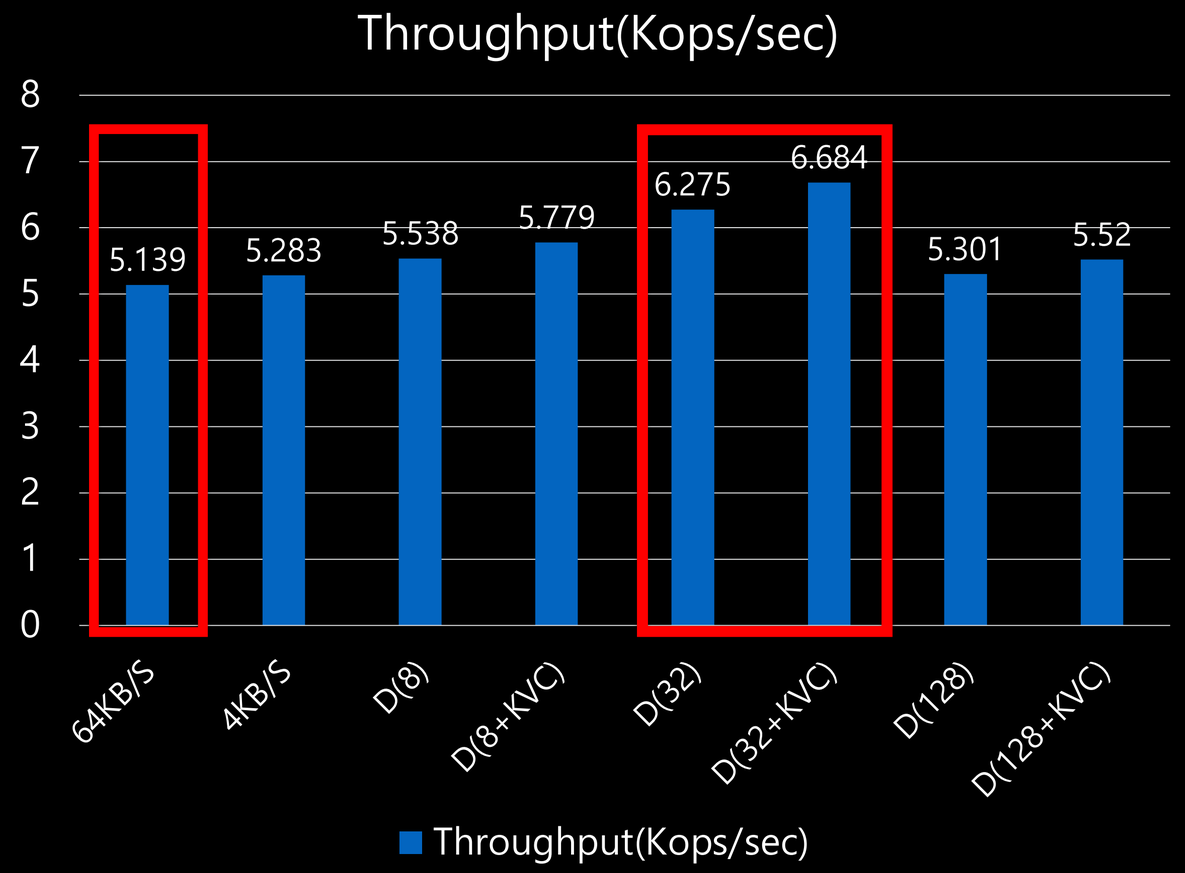


5.2 CUVE: CBMG Benchmark :

Cache Hit Ratio & Throughput

NAVER DEVIEW 2023





6. Conclusion
마무리

6 Conclusion

NAVER
DEVIEW
2023

Using 64KB Data block in Vanilla HBase:

- Granularity Mismatch Problem -> Unnecessary Disk I/O
- Cache Pollution Problem -> Degrade Cache Utilization

Using 4KB Data block in Vanilla HBase:

- Frequent small random read -> Disk seek overhead
- Increasing the size of index blocks -> Disk I/O for read index block

6 Conclusion

NAVER
DEVIEW
2023

Solution 1: **Dynamic Variable Block Size Control**

- ❑ Reduce unnecessary Disk I/O for small Key-Value
- ❑ Increase spatial locality for large Key-Value

Solution 2: **Fine-grained KV Cache**

- ❑ Increase cache efficiency for prevent frequent small random read
- ❑ Increase cache utilization in large data block

Q & A

Thank You