DB+Storage is All You Need

Jonghyeok Park

Dept of Computer Science and Engineering

March. 26, 2025

Who Am I

Jonghyeok Park 박종혁

- Experiences
  - 2013.03 ~ 2016.08: B.S. in Software from Sungkyunkwan University
  - 2016.09 ~ 2022.08: Ph.D. in Computer Science from Sungkyunkwan University
    (Advisor Prof. Sang-Won Lee)
  - 2019.11 ~ 2020.03: Visiting research student at Simon Fraser University
    (Host Prof. *Tianzheng Wang*)
  - 2023.03 ~ 2024.08: Assistant Professor at Hankuk University of Foreign Studies
  - 2024.09 ~ current: Assistant Professor at Korea University

- Research Areas
  - Database systems, Storage Systems, Flash/NVM-based DBMS

Overview

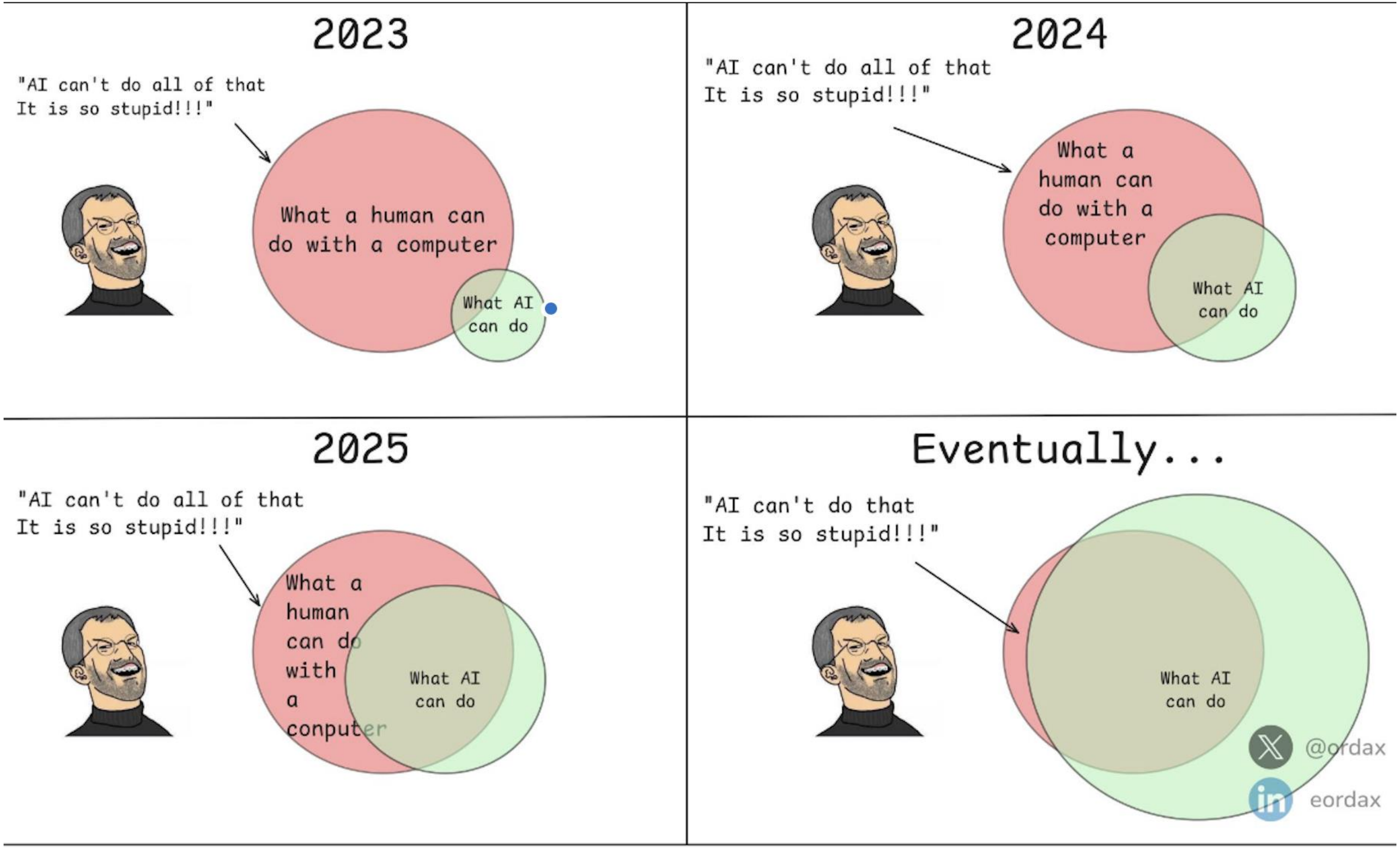
- Experiences
- Why do we study Database and Storage
- SaS: SSD as SQL engine
- DBS Lab.
- Advice

In 2013 …

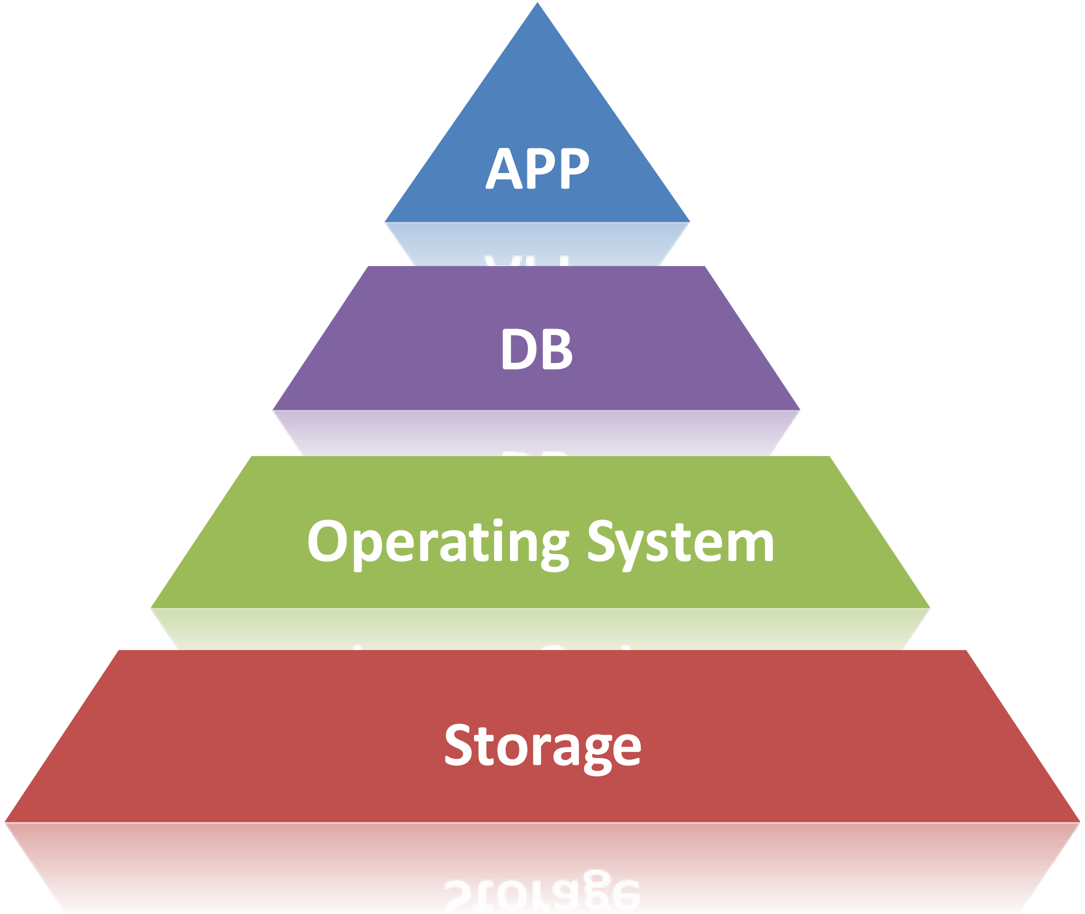
- 콩순이
- Find employment vs. **Graduate School**

* *generated from GPT*

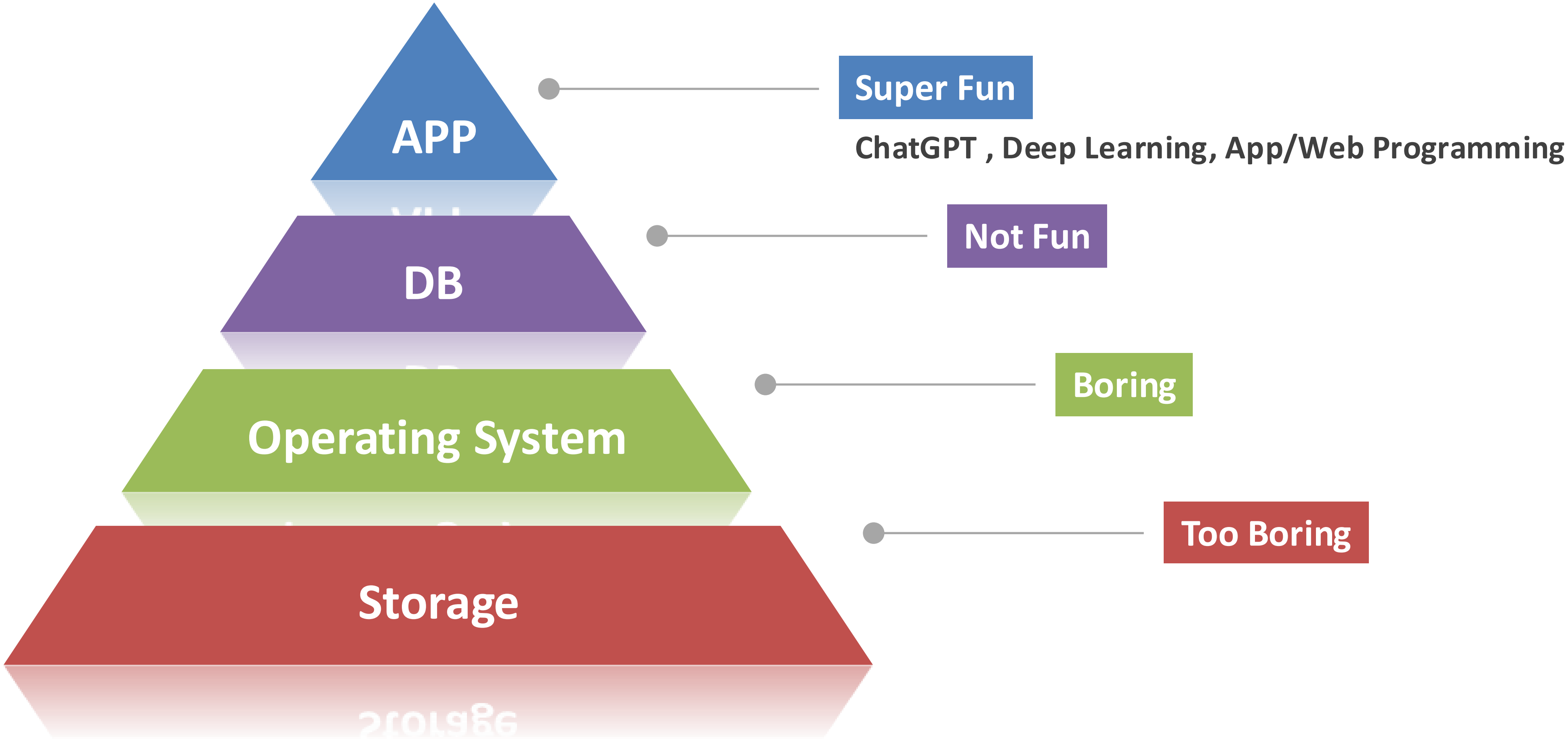
Now ...


2023
"AI can't do all of that
It is so stupid!!!"
What a human can do with a computer
What AI can do
2024
"AI can't do all of that
It is so stupid!!!"
What a human can do with a computer
What AI can do
2025
"AI can't do all of that
It is so stupid!!!"
What a human can do with a conputer
What AI can do
Eventually...
"AI can't do that
It is so stupid!!!"
What AI can do
@ordax
eordax

Why Do we Study Database?



Why Do we Study Database?



APP

DB

Operating System

Storage

Super Fun

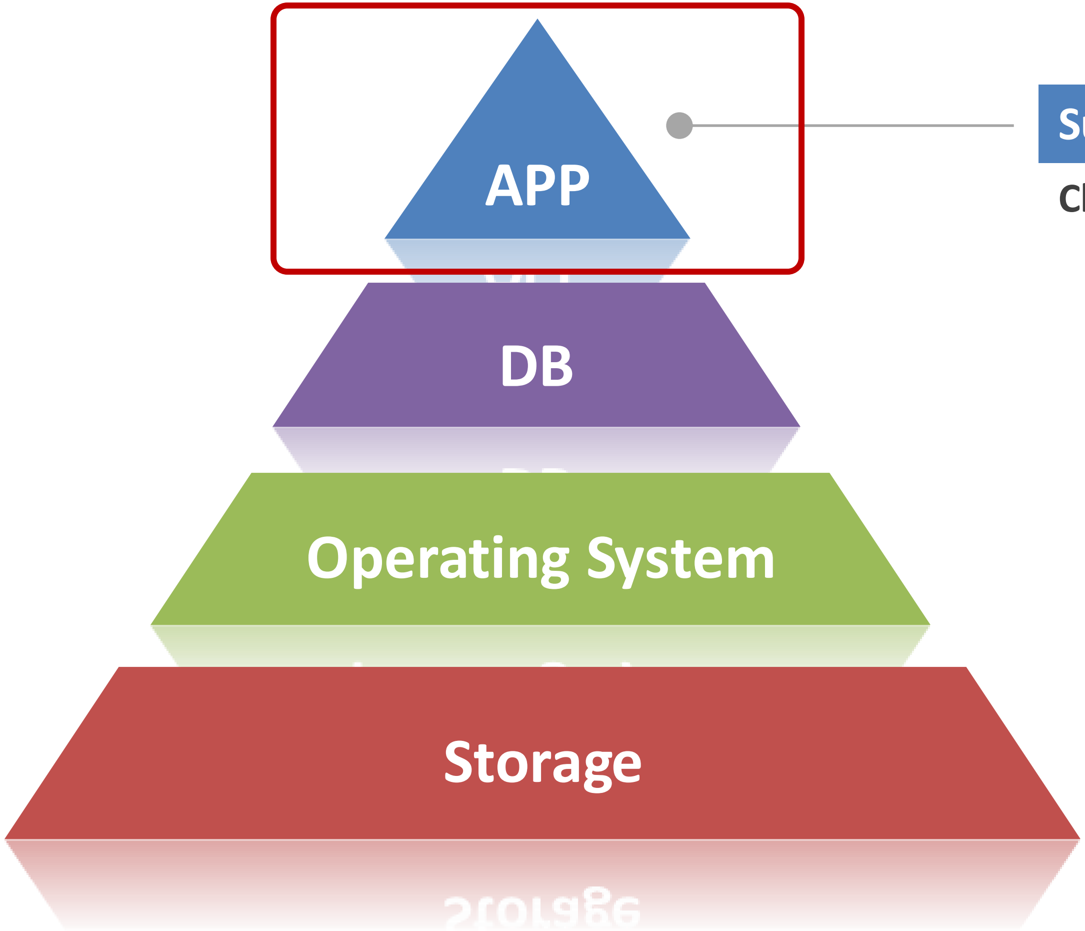
ChatGPT , Deep Learning, App/Web Programming

Not Fun

Boring

Too Boring

Why Do we Study Database?

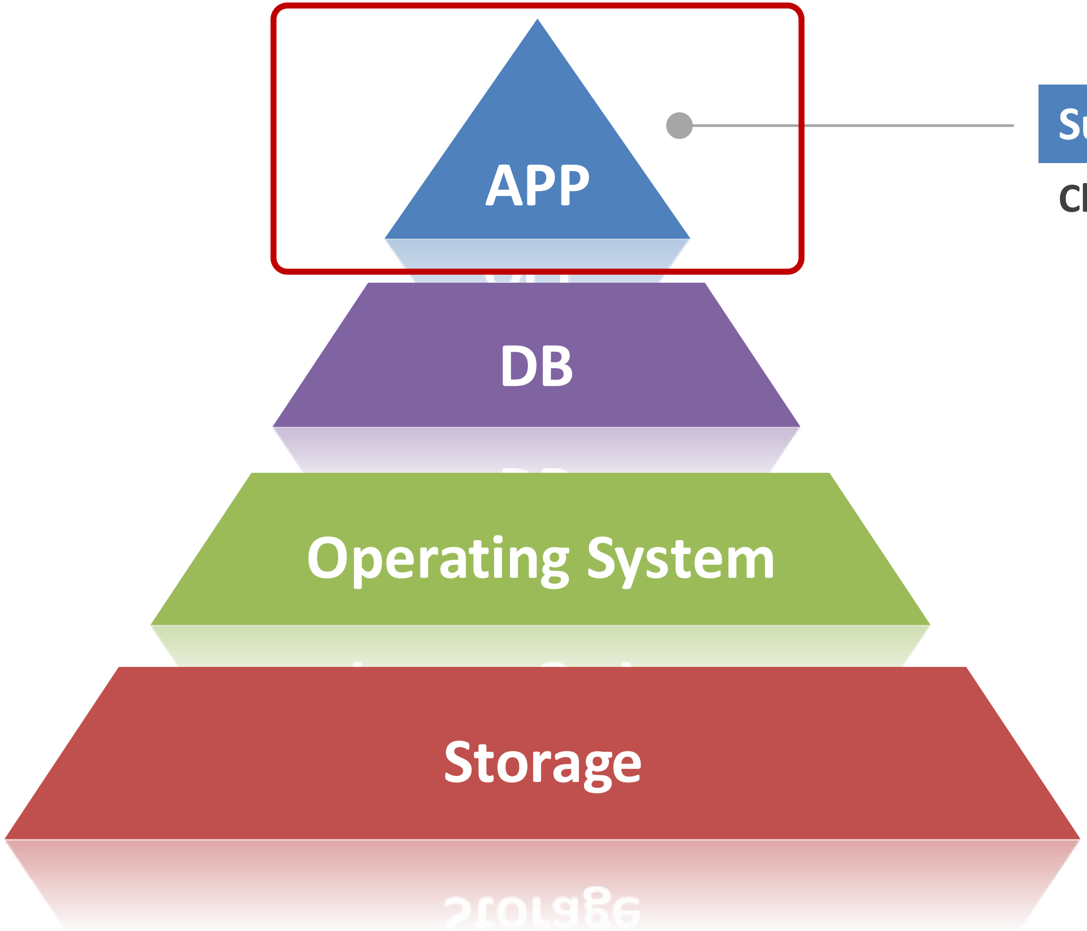


Super Fun

ChatGPT , Deep Learning, App/Web Programming

- Easy and Fun
- Fancy
- Low entry barriers
- Rapidly changing

Why Do we Study Database?



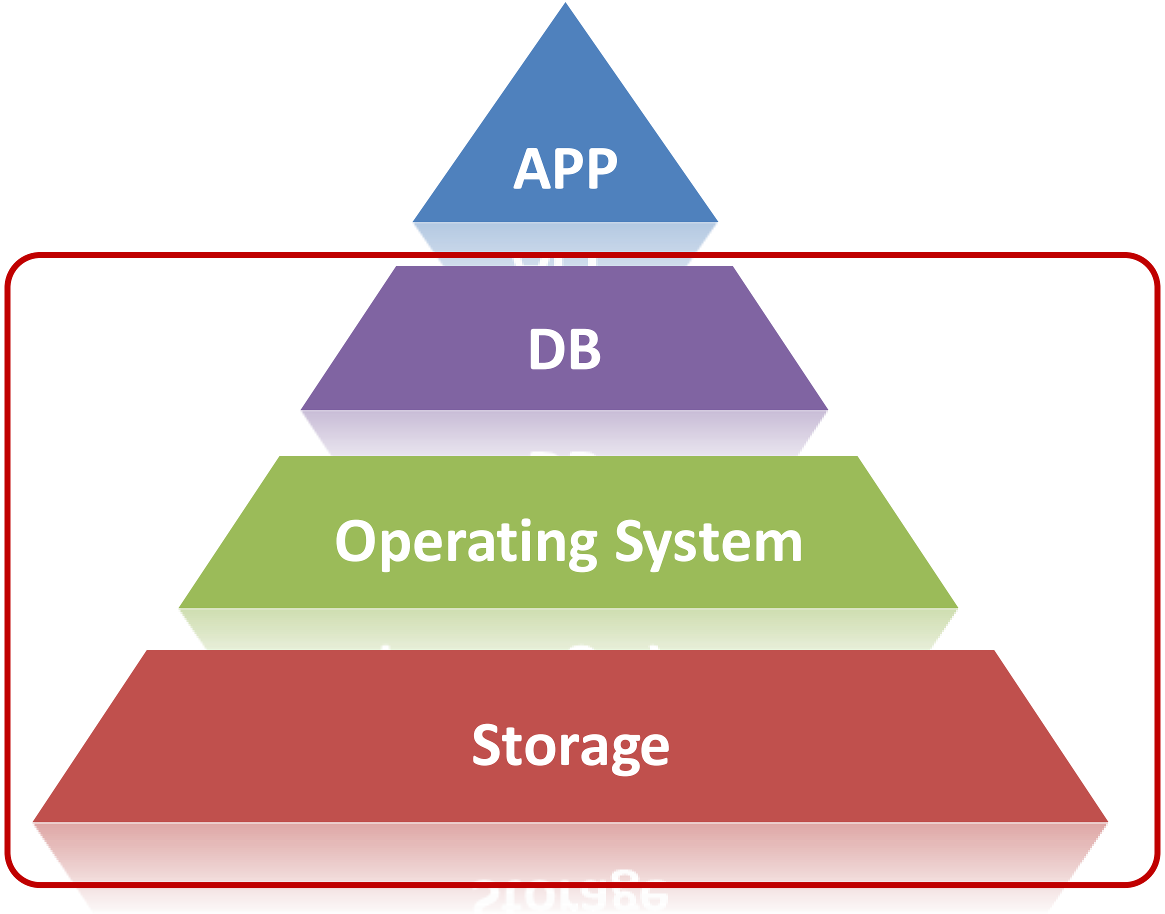
Super Fun

ChatGPT , Deep Learning, App/Web Programming

- Easy and Fun
- Fancy
- Low entry barriers
- Rapidly changing

Anyone can do it; Replaceable

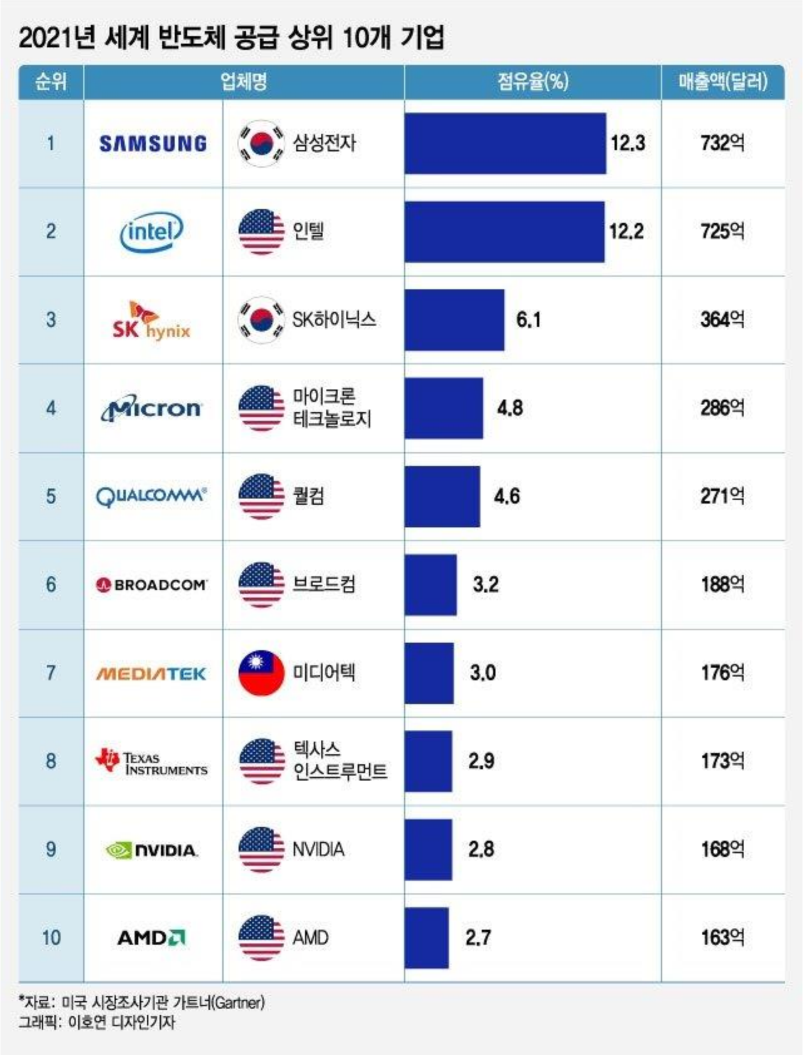
Why Do we Study Database?



- Difficult
- Hard
- High entry barrier
- Steep learning curve
- Slowly Changing

Not everyone can do it; Irreplaceable

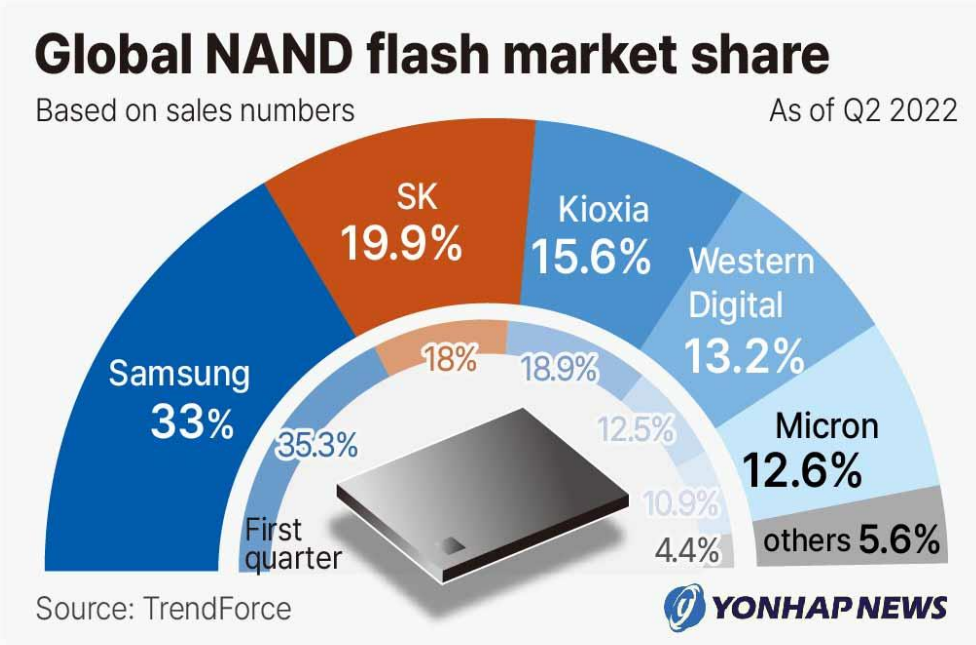
Why Should we study DB&Storage in Korea?



2021년 세계 반도체 공급 상위 10개 기업

순위	업체명		점유율(%)	매출액(달러)
1	SAMSUNG	삼성전자	12.3	732억
2	intel	인텔	12.2	725억
3	SK hynix	SK하이닉스	6.1	364억
4	Micron	마이크론 테크놀로지	4.8	286억
5	QUALCOMM	퀄컴	4.6	271억
6	BROADCOM	브로드컴	3.2	188억
7	MEDIATEK	미디어텍	3.0	176억
8	TEXAS INSTRUMENTS	텍사스 인스트루먼트	2.9	173억
9	NVIDIA	NVIDIA	2.8	168억
10	AMD	AMD	2.7	163억

*자료: 미국 시장조사기관 가트너(Gartner)
그래픽: 이호연 디자인기자



Global NAND flash market share

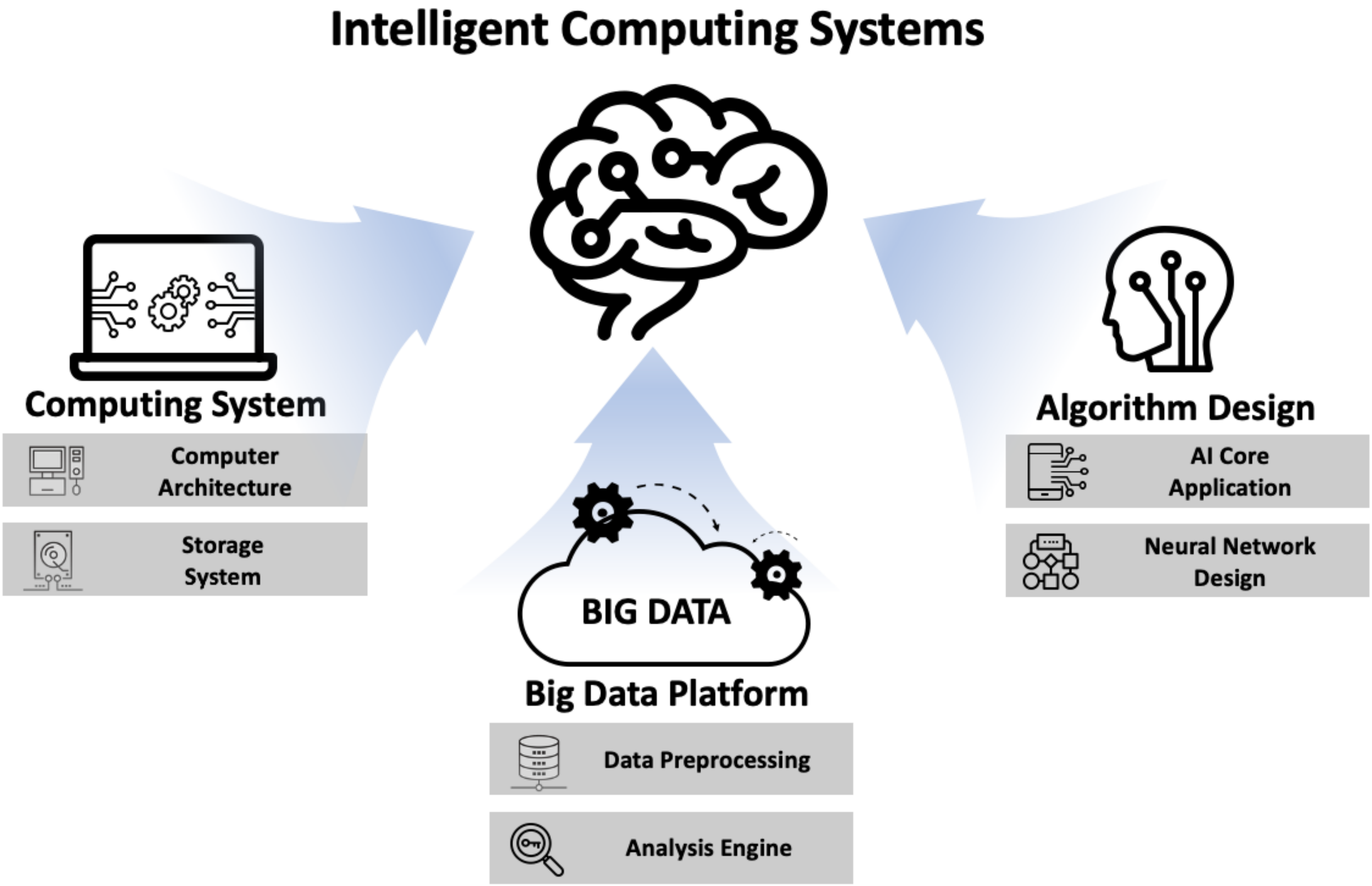
Based on sales numbers

As of Q2 2022

Source: TrendForce

YONHAP NEWS

Core Technology for AI



Intelligent Computing Systems

Computing System

- Computer Architecture
- Storage System

Algorithm Design

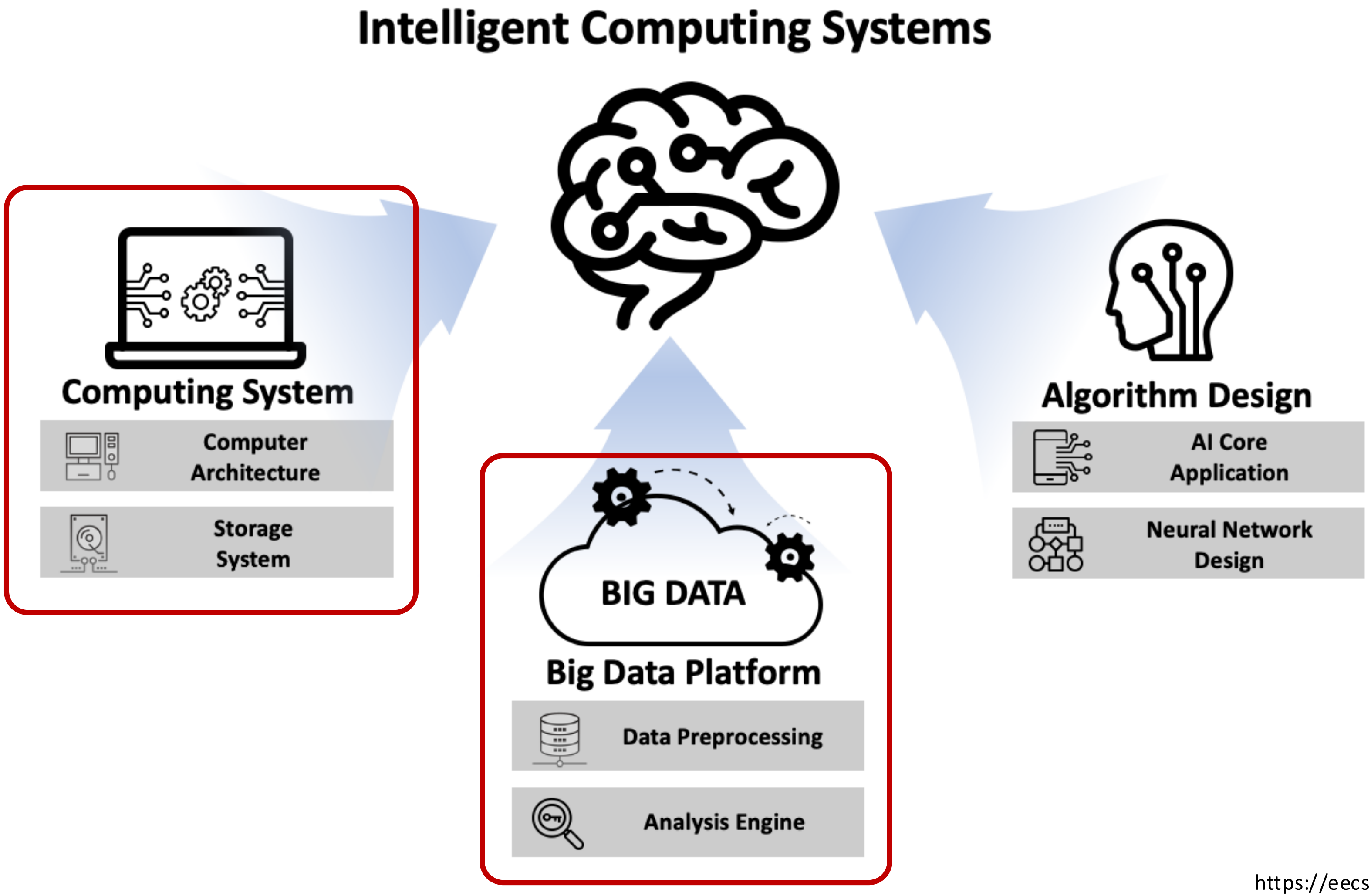
- AI Core Application
- Neural Network Design

Big Data Platform

- Data Preprocessing
- Analysis Engine

https://eecs.dgist.ac.kr/en/page/36/

Core Technology for AI



Intelligent Computing Systems

Computing System

- Computer Architecture
- Storage System

Big Data Platform

- Data Preprocessing
- Analysis Engine

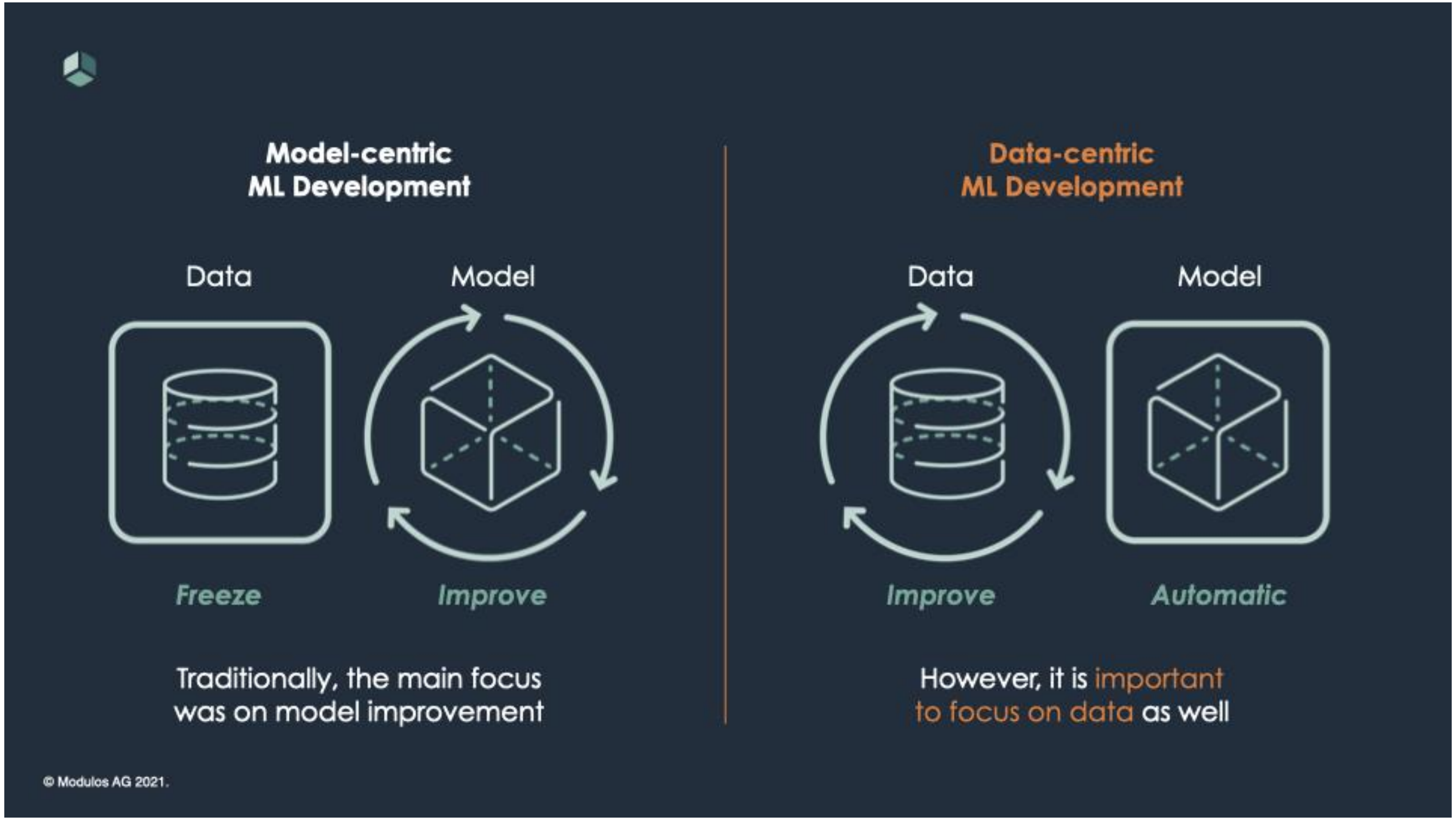
Algorithm Design

- AI Core Application
- Neural Network Design

https://eecs.dgist.ac.kr/en/page/36/

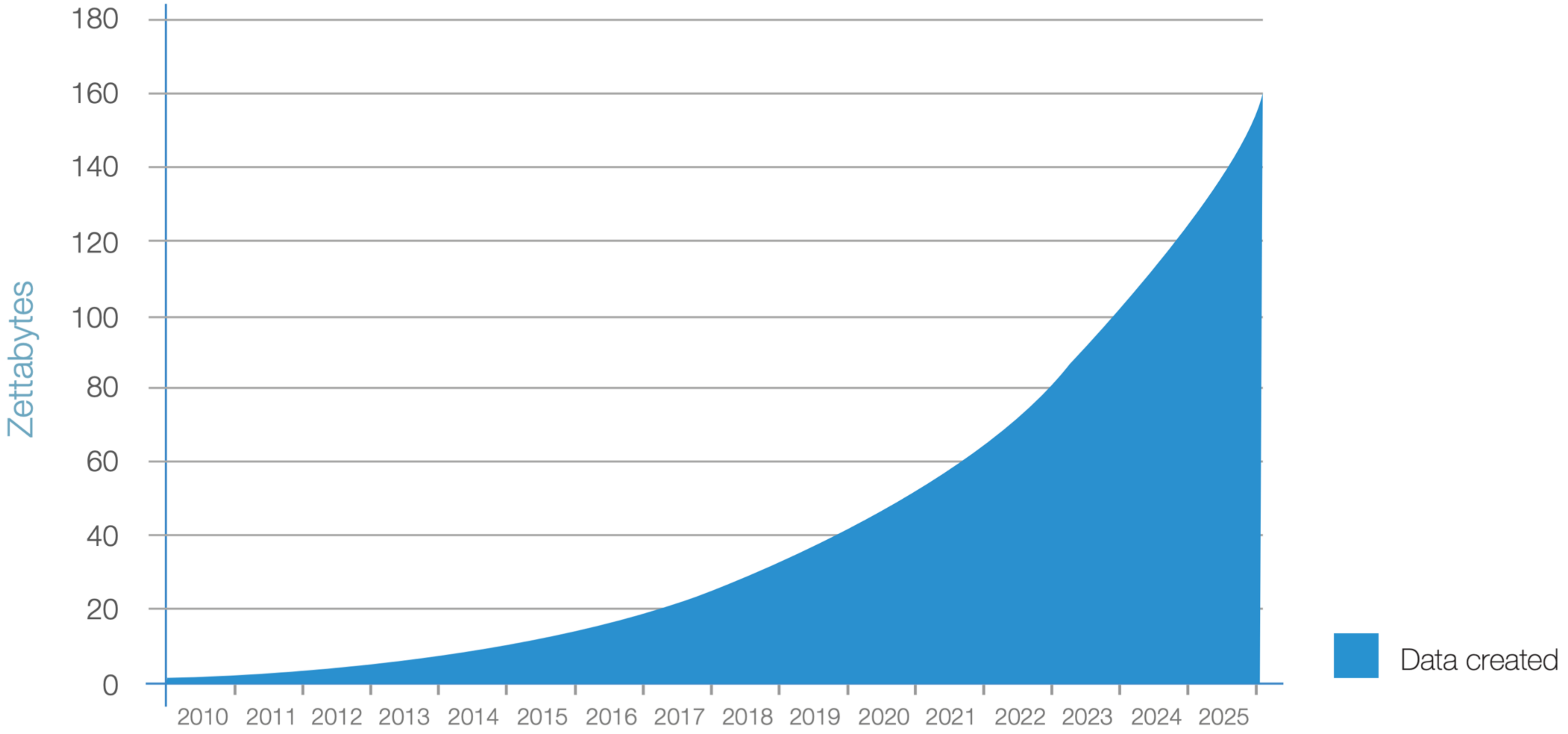
Dear AI, IT is Data

- Data-centric AI



Big Data : Big in Growth Too

1ZB = 1,000,000,000,000,000,000,000 bytes



Source: IDC's Data Age 2025 study, sponsored by Seagate, April 2017

Big Data : Big in Growth Too

1ZB = 1,000,000,000,000,000,000,000 bytes

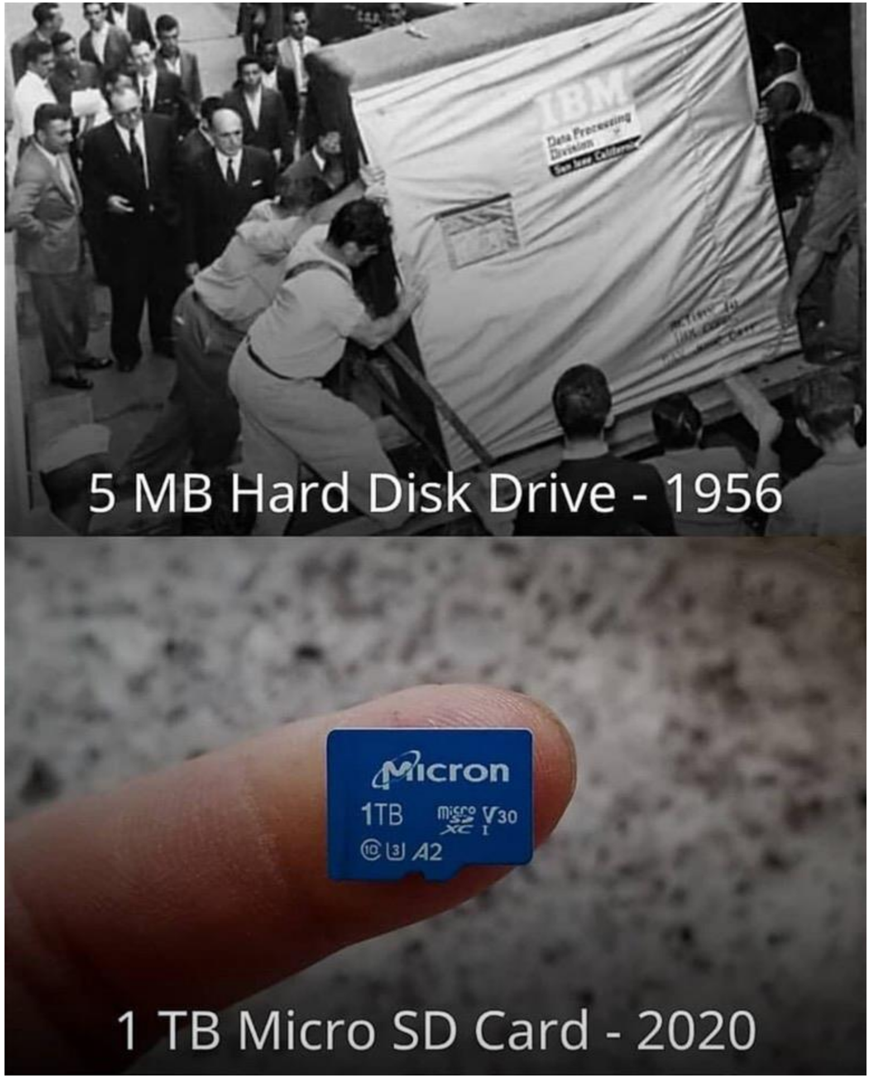
How to manage BIG DATA efficiently?

Speed, Durability, Cost

180
160
20
0
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
Data created

Source: IDC's Data Age 2025 study, sponsored by Seagate, April 2017

Storage Systmes

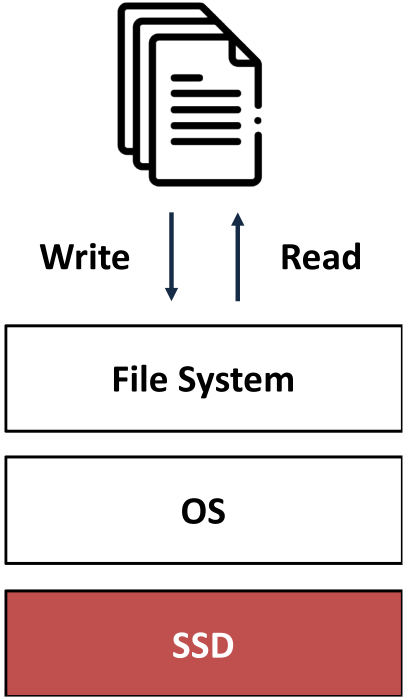


5 MB Hard Disk Drive - 1956

1 TB Micro SD Card - 2020

SSD (Solid State Drive)

- Charcteristics of Flash memory

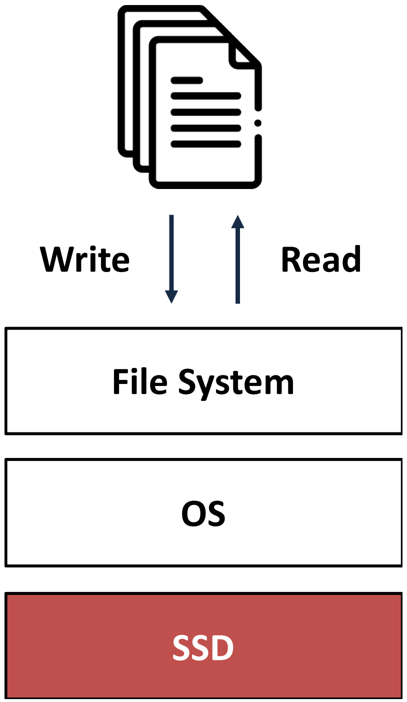


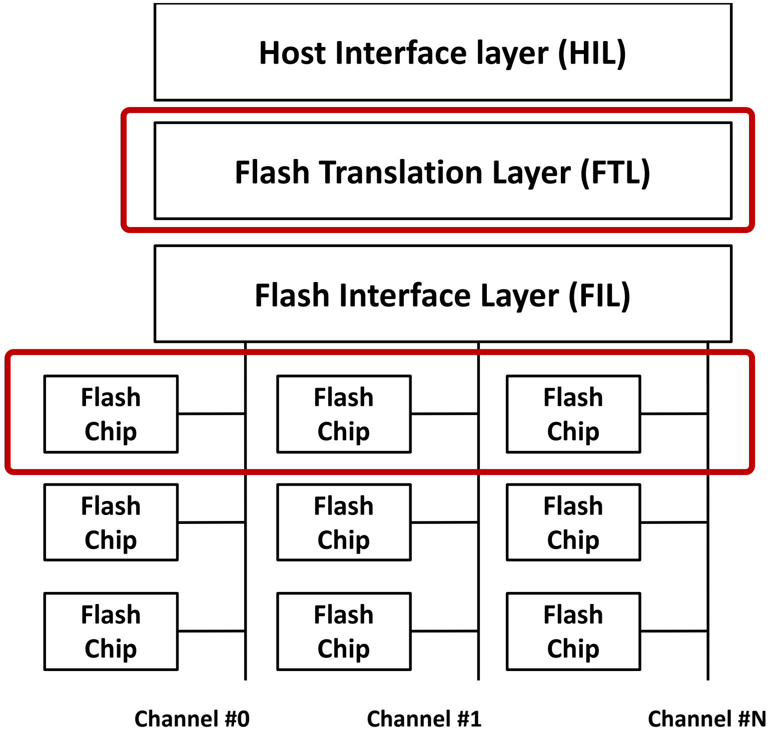
SSD (Solid State Drive)

1. Parallelism & Translation

Translate Logical address to Physical address!

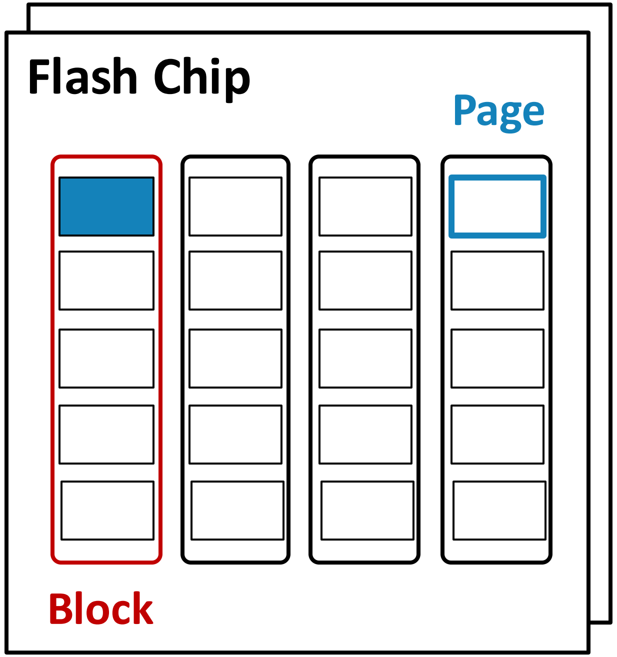
Logical Block Address → Physical Block Address





SSD (Solid State Drive)

2. Different operation granulality
   - Read and write operations are performed on a page level
   - Erase operation occurs at block level



LBA	PBA	Valid
0	(0,0)	Y

Mapping TBL

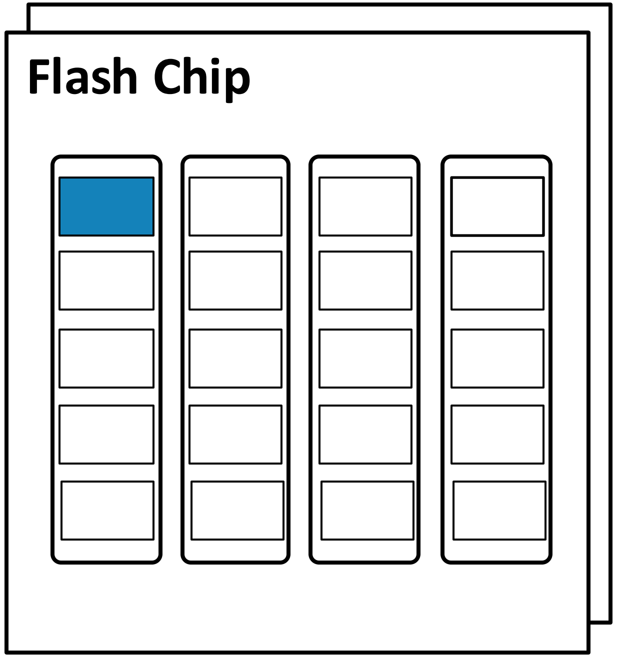
Translation

Write LBA 0 → **Write** Block #0, Page #0

Read LBA 0 → **Read** Block #0, Page #0

SSD (Solid State Drive)

3. No overwrite
   - Read and write operations are performed on a page level
   - Erase operation occurs at block level



LBA	PBA	Valid
0	(0,0)	Y

Mapping TBL

Translation

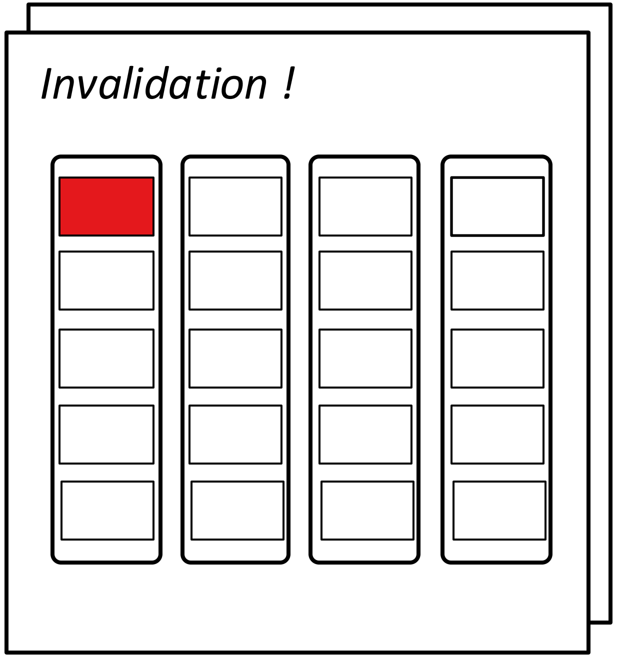
Write LBA 0 → Write Block #0, Page #0

Read LBA 0 → Read Block #0, Page #0

Write LBA 0 → Write Block #0, Page #1

SSD (Solid State Drive)

3. No overwrite
   - Read and write operations are performed on a page level
   - Erase operation occurs at block level



LBA	PBA	Valid
0	(0,0)	N

Mapping TBL

Translation

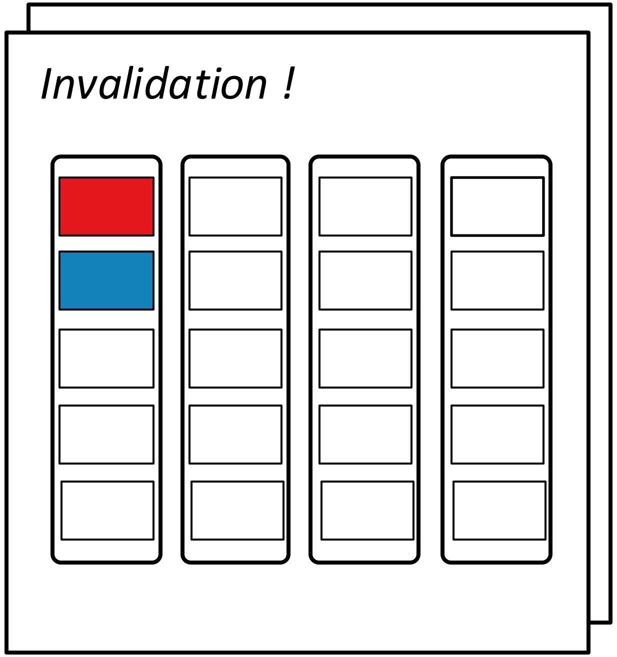
Write LBA 0 → **Write** Block #0, Page #0

Read LBA 0 → **Read** Block #0, Page #0

Write LBA 0 → **Write** Block #0, Page #1

SSD (Solid State Drive)

3. No overwrite
   - Flash chips have a finite lifespan



LBA	PBA	Valid
0	(0,1)	Y

Mapping TBL

Translation

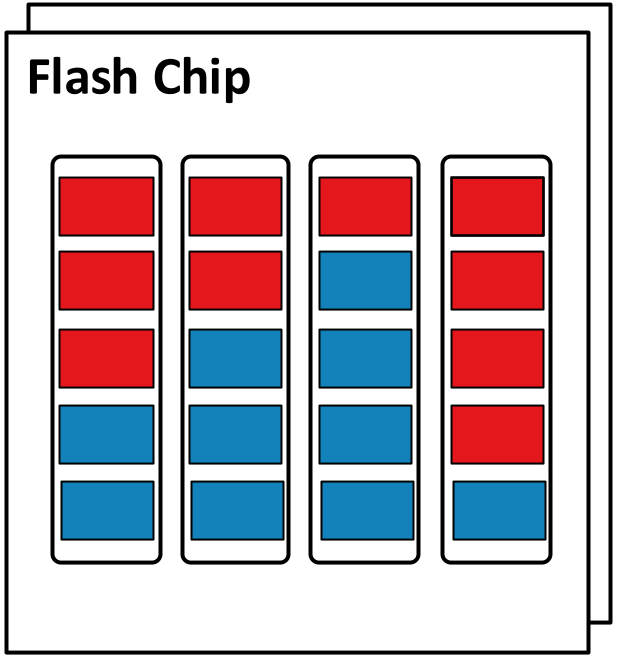
Write LBA 0 → **Write** Block #0, Page #0

Read LBA 0 → **Read** Block #0, Page #0

Write LBA 0 → **Write** Block #0, Page #1

SSD (Solid State Drive)

4. When the block is full
   - Select the victim block with fewest valid pages (Greedy Algorithm)

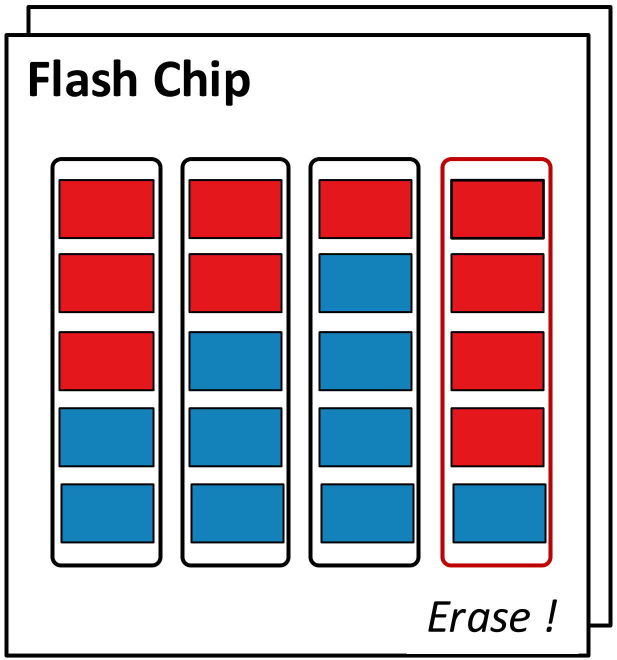


LBA	PBA	Valid
0	(0,1)	Y

Mapping TBL

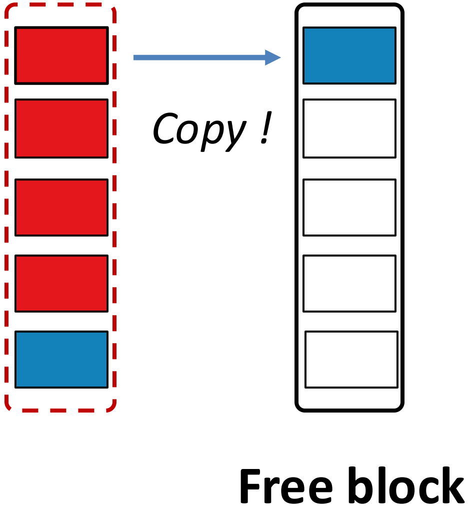
SSD (Solid State Drive)

4. When the block is full, we need GC (Garbage Collection)
   - Select the victim block with fewest valid pages (Greedy Algorithm)
   - Copy the valid pages to free block and then erase the original block



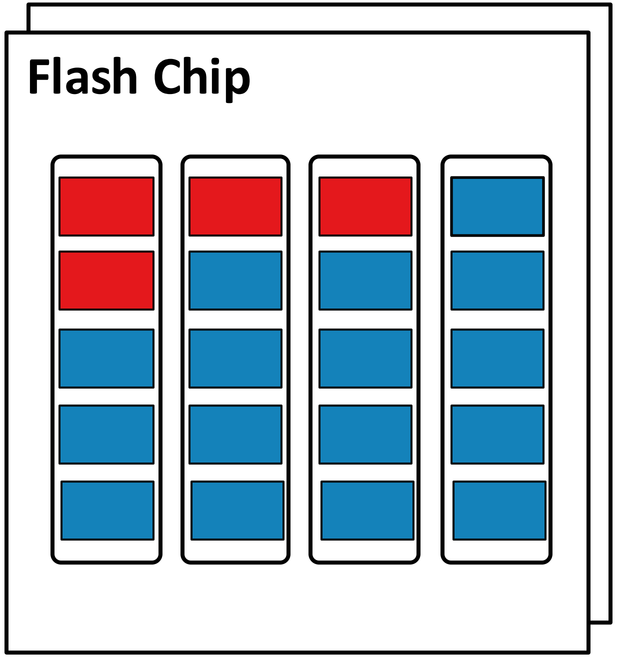
LBA	PBA	Valid
0	(0,1)	Y

Mapping TBL



The Achilles' heel of SSDs

- Write Amplification
  - Worsen the performance and shorten the lifespan of flash chips



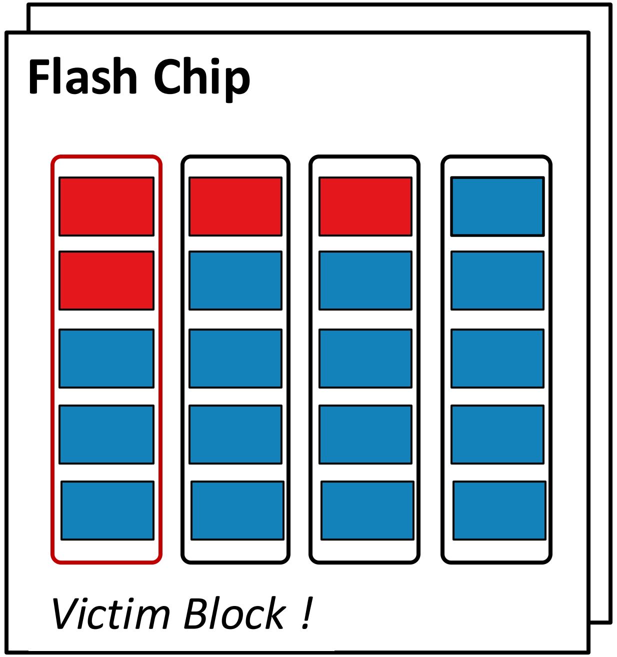
…

Write LBA 0 → **Write** Block #0, Page #0

GC is triggered

The Achilles' heel of SSDs

- Write Amplification
  - Worsen the performance and shorten the lifespan of flash chips



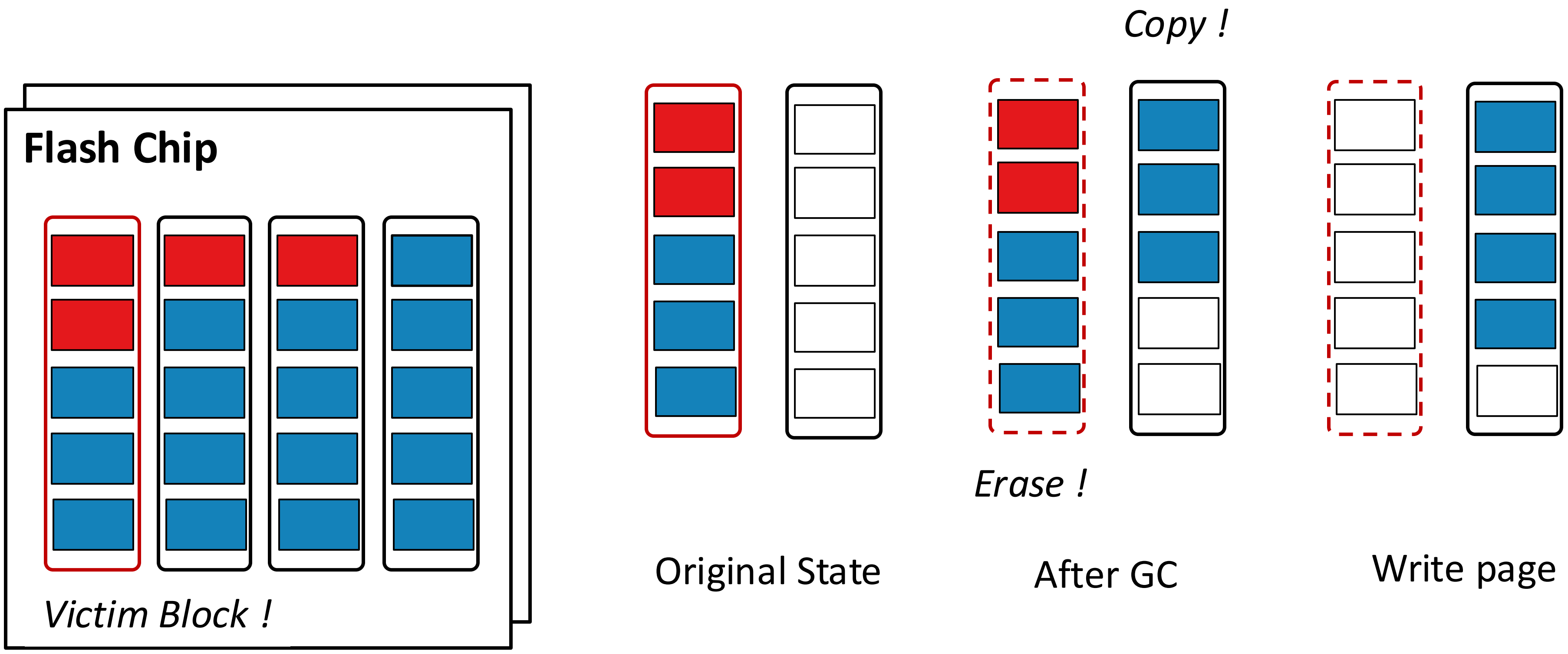
…

Write LBA 0 → **Write** Block #0, Page #0

GC is triggered

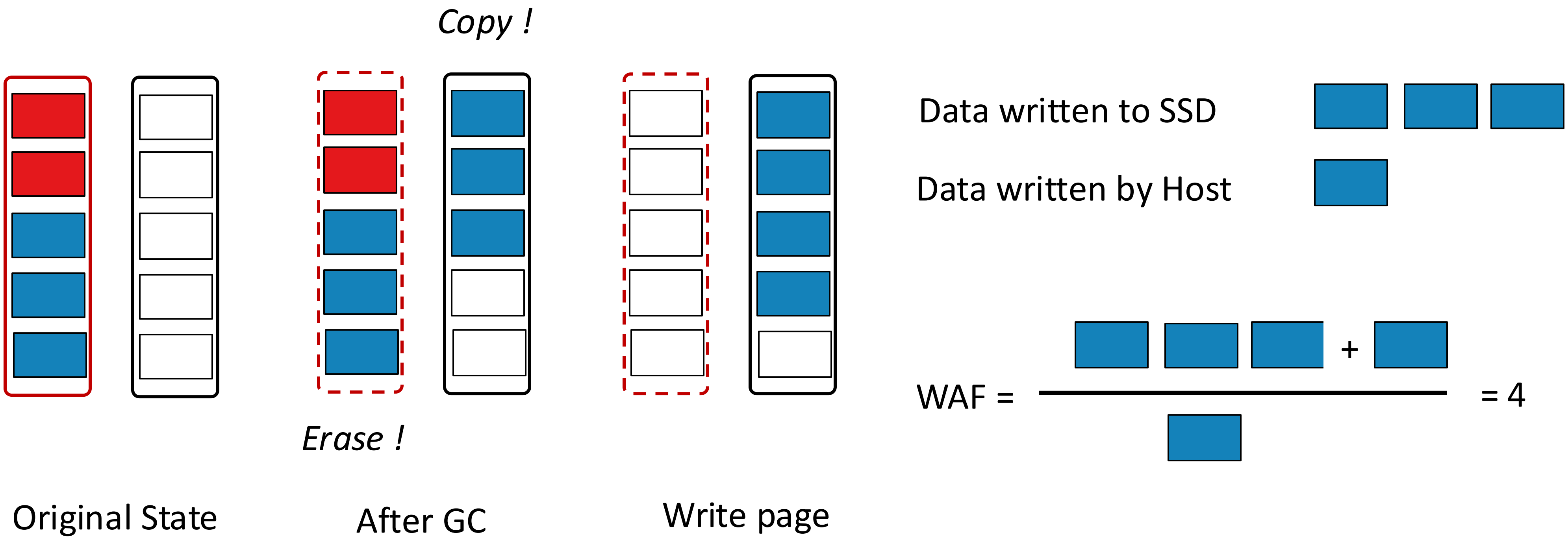
The Achilles' heel of SSDs

- Write Amplification
  - Worsen the performance and shorten the lifespan of flash chips



The Achilles' heel of SSDs

- Write Amplification
  - Worsen the performance and shorten the lifespan of flash chips



Copy !

Erase !

Original State

After GC

Write page

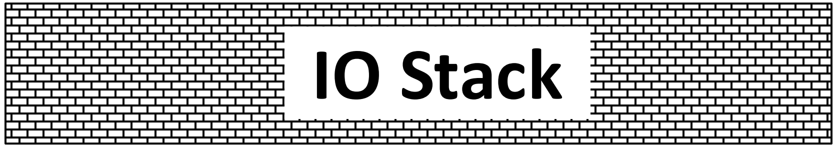
Data written to SSD

Data written by Host

WAF = = 4

We need the new storage

High Performance



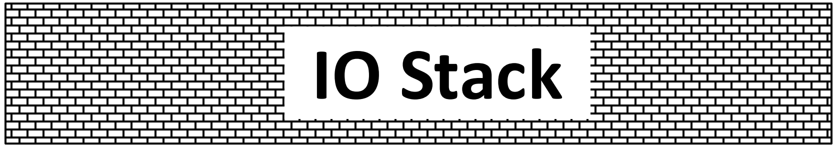
IO Stack

Bottleneck #2

Bottleneck #1

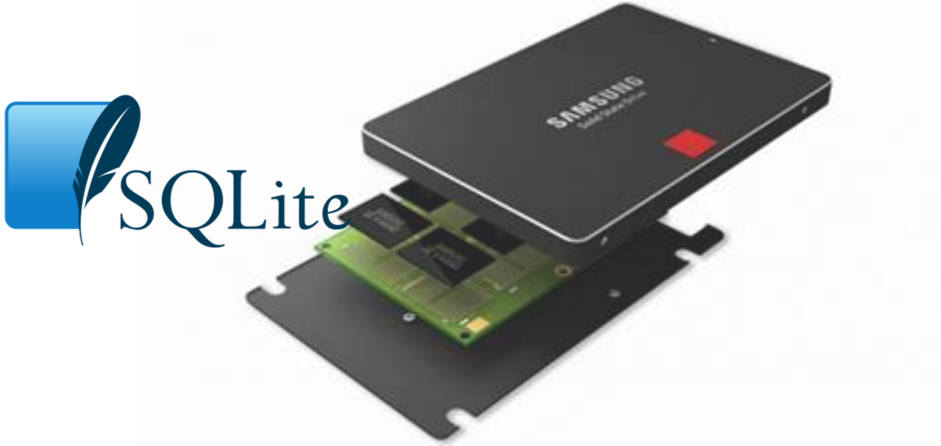
We need the new storage

PostgreSQL
amazon DynamoDB
OrientDB Community Edition
neo4j
SQLite
MySQL
redis
elasticsearch
mongoDB
cassandra
Microsoft SQL Server
ORACLE
IBM DB2
MariaDB
mobulous



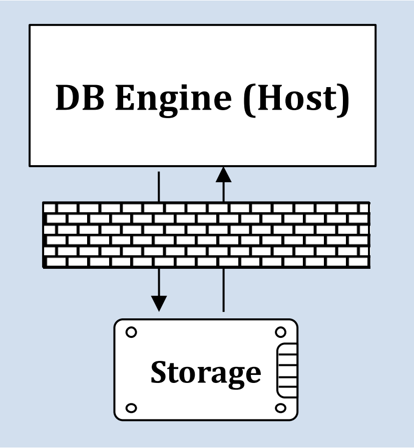
We need the new storage

- SaS: SSD as SQL Engine



Motivation

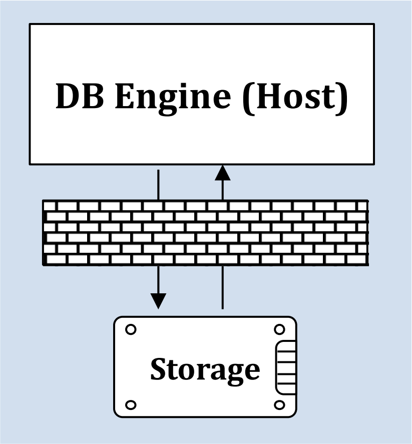
- **Databases: bedrock of modern service**
  - Meta, Google
  - MS Azure, Amazon Aurora, Data Bricks, Snowflake
- **Computer architecture**
  - Dichotomy of host and storage
  - In-host database engines (**IHDE**)
- **Era of flash memory SSDs**
  - "Flash is disk, disk is tape, and tape is dead." (Jim Gray)
- **DB computing paradigm**: host-centric → **SSD-centric**



IHDE

Why IHDEs are so Inefficient on SSDs (1)

- **Dichotomy** of Host and Storage
- In-Host Database Engine (IHDE)

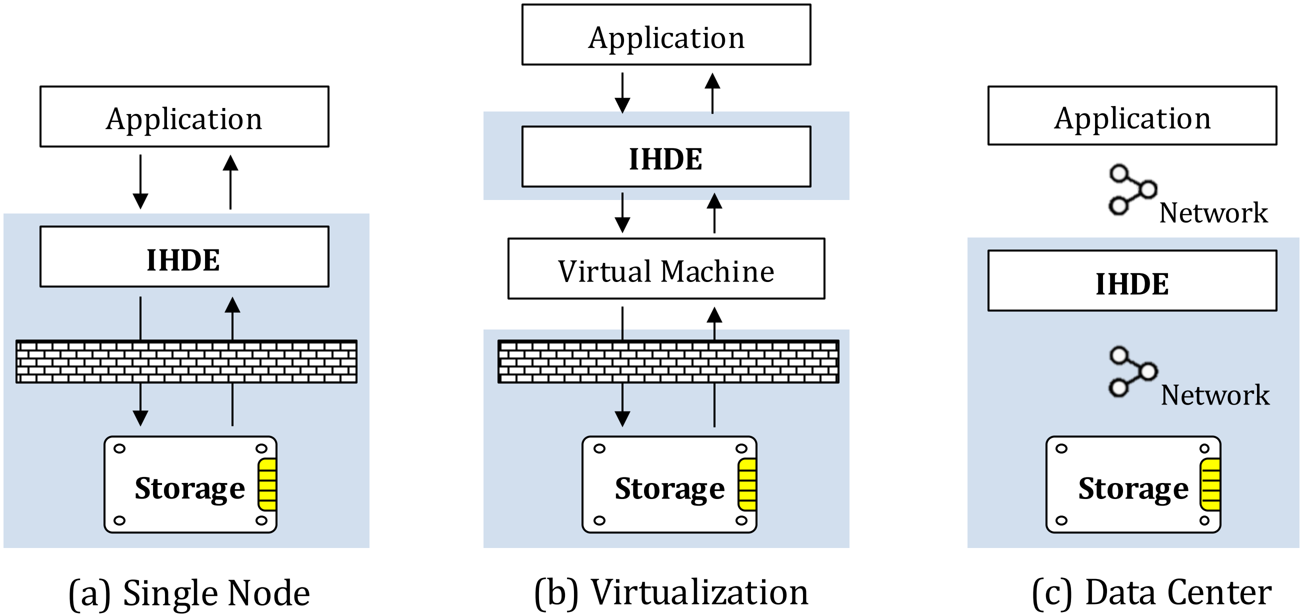


IHDE

1. IO Stack Overhead
2. Hinder Vertical Optimization
3. Architectural Inefficiencies

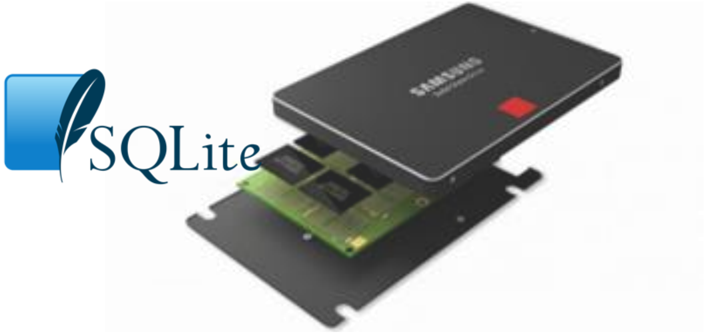
Why IHDEs are so Inefficient on SSDs (2)

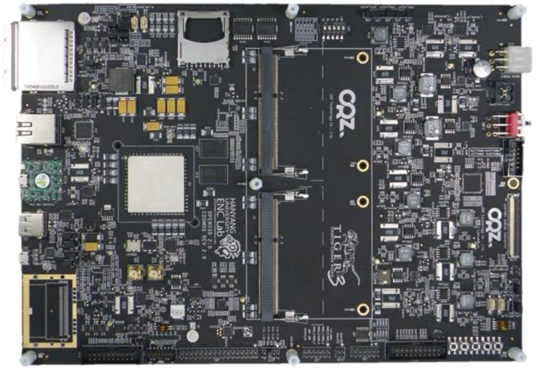
- **Legacy IO stack overhead**
  - Latency, CPU instructions, interrupt, etc.
  - (+) Virtualization barrier: Docker, Container
  - (+) Network latency barrier : Serverless / Disaggregation (e.g., Amazon Aurora)



Jim Gray's Vision and Ours

> *All storage systems will eventually evolve to be database systems*

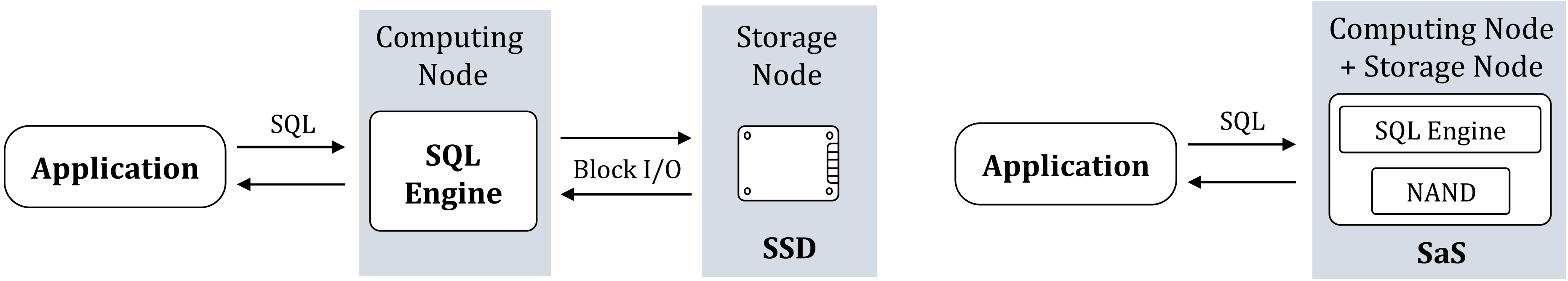




Cosmos+ Open SSD

SaS: SSD as SQL Engine

- **Let’s a full-blown DB engine run inside SSD**
  - Eliminate IO stack overhead
  - Enable seamless vertical optimizations
  - More elegant and economical architecture

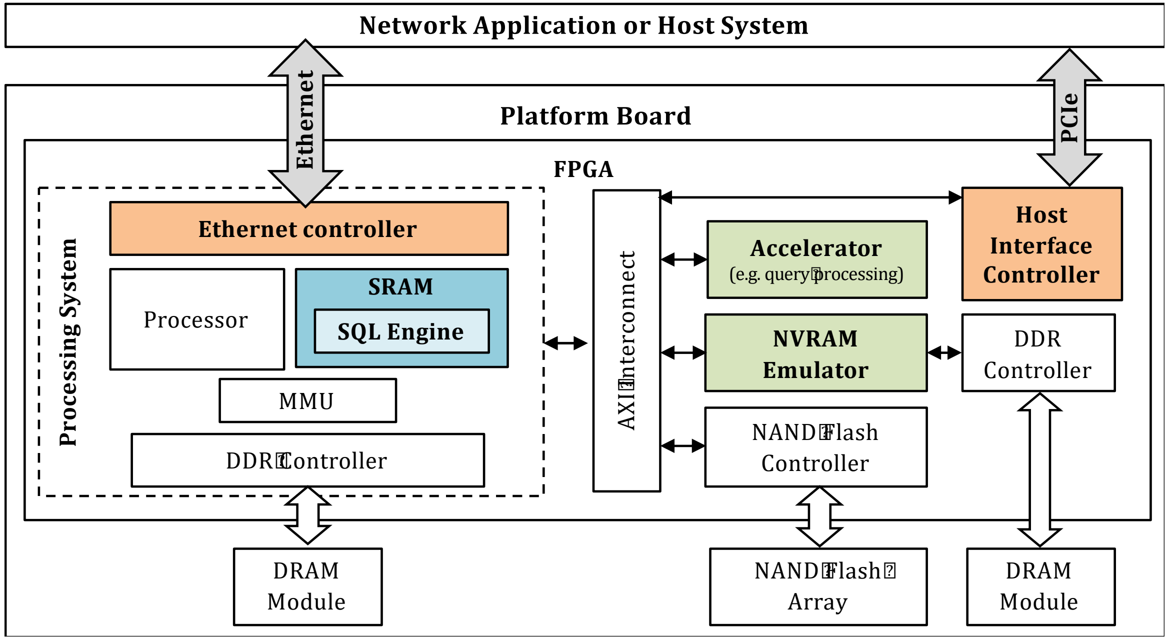


(a) IHDE Architecture

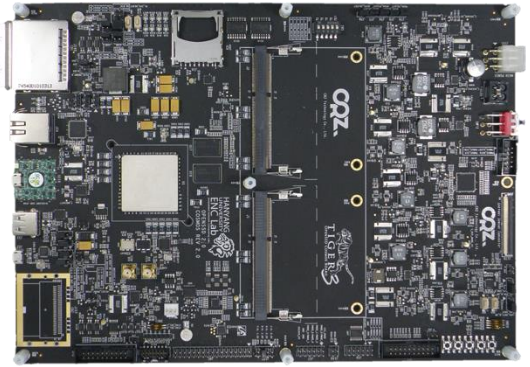
(b) SaS Architecture

SaS: SSD as SQL Database System

- SQL Interface
- Vertical IO Optimization
- Hardware-assisted Acceleration



SaS Architecture

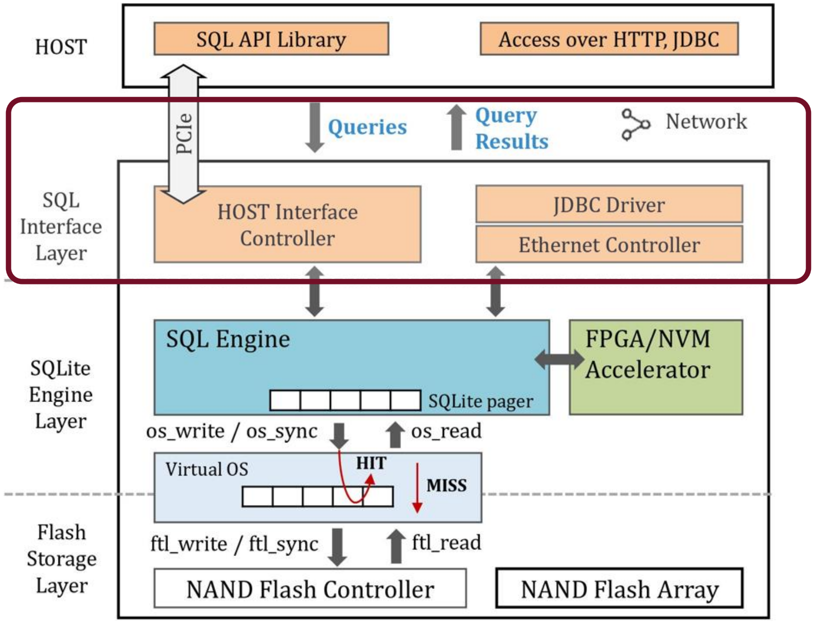


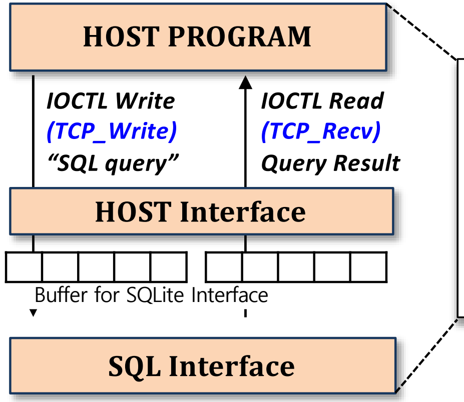
SaS : HW specification	
CPU	Dural ARM Cortex-A9 1 GHz
DRAM	1 GB
SRAM	256 KB

Cosmos+ Open SSD

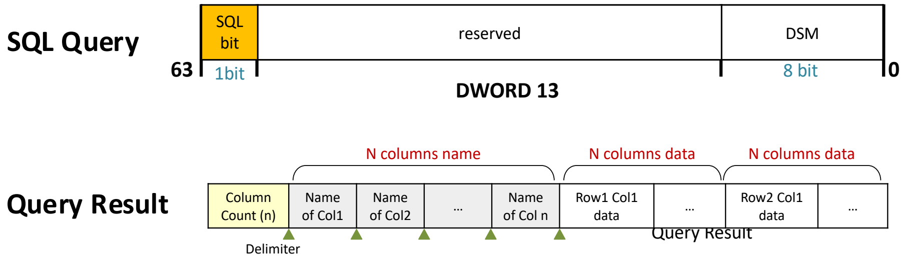
SQL Interface

- Support **tuple-oriented** SQL interface over **block-oriented** interface
- SQL Query IO command (NVMe command)
- Ethernet network Interface (LwIP)



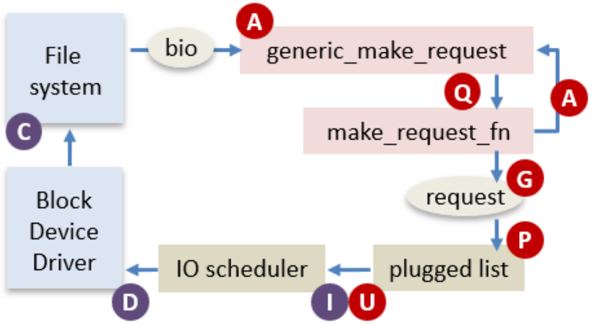


- HOST program issue R/W command with **SQL bit**
- SQL interface extract SQL from HOST Write Buffer
- Support Ethernet Interface via LwIP library

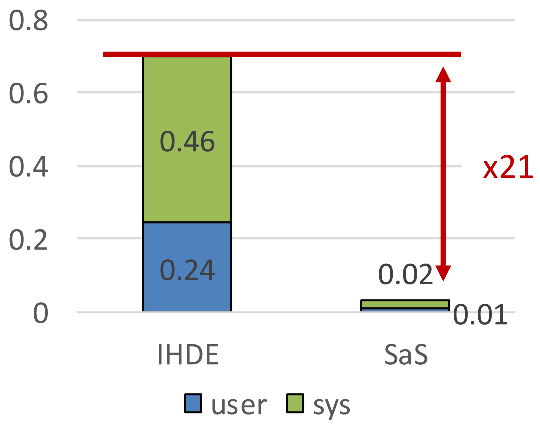


SQL Interface

- Host CPU Time (IHDE vs. SaS)



Block Trace Event Flow



CPU Time (IHDE vs. SaS)

IHDE

```
Event Process Length
--------------------
A                 4K
Q      sqlite3    4K
G      sqlite3    4K
P      sqlite3
A                 4K
Q      sqlite3    4K
G      sqlite3    4K
A                 4K
Q      sqlite3    4K
G      sqlite3    4K
U      sqlite3
D      sqlite3    4K
D      sqlite3    4K
D      sqlite3    4K
C                 4K
C                 4K
C                 4K
Q      sqlite3
G      sqlite3
D
C               flush
```

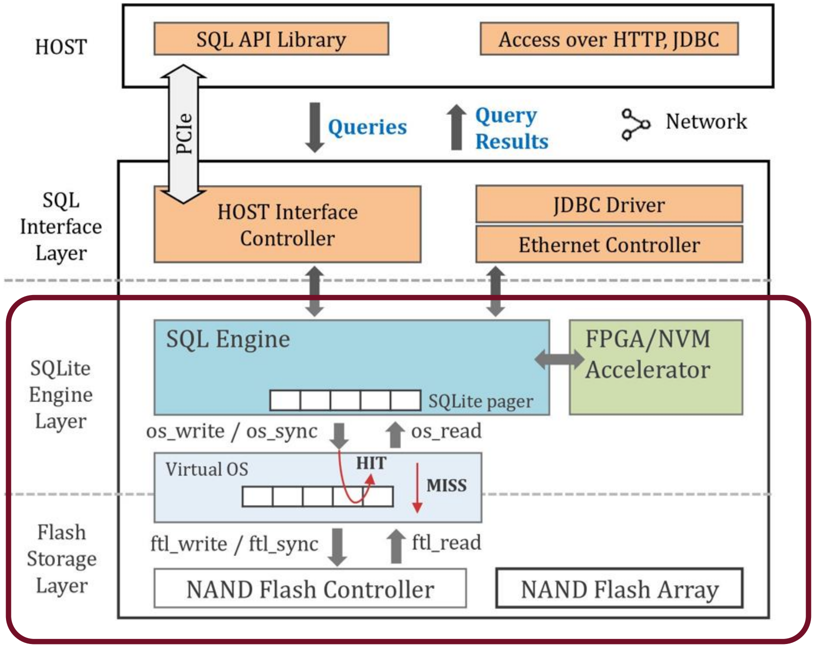
SaS

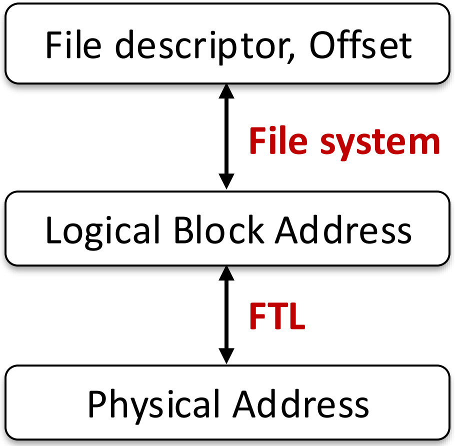
```
Event Process Length
--------------------
I      ioctl     512
D      ioctl     512
C                0
```

Block Trace for single *INSERT* query on IHDE vs. SaS

Vertical Optimizations

- **WITHOUT** operating system
  - Unified space management and transparent address translation





File descriptor, Offset

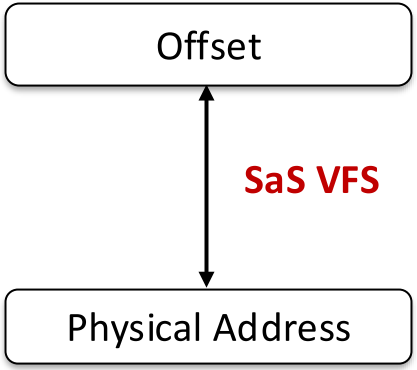
File system

Logical Block Address

FTL

Physical Address

IHDE



Offset

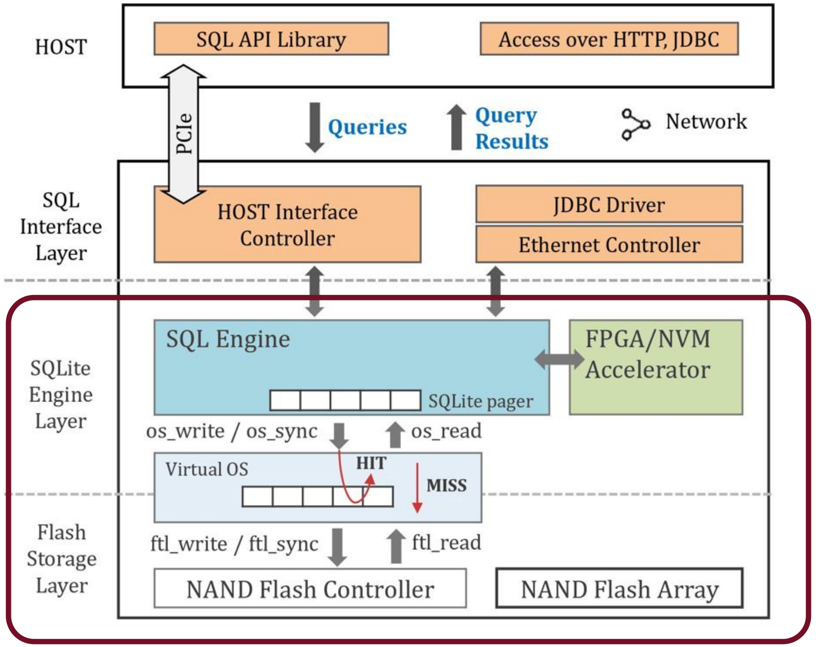
SaS VFS

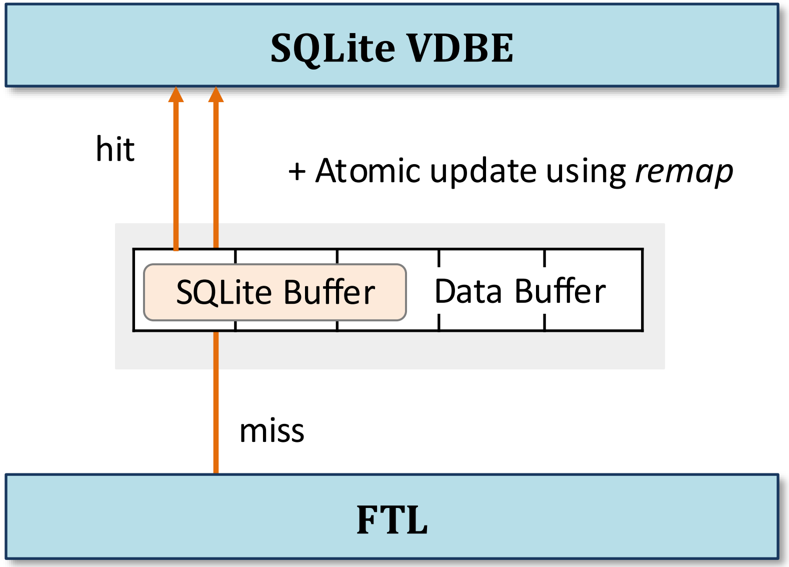
Physical Address

SaS

Vertical Optimizations

- **WITHOUT** operating system
  - Unified space management and transparent address translation
  - Unified memory management

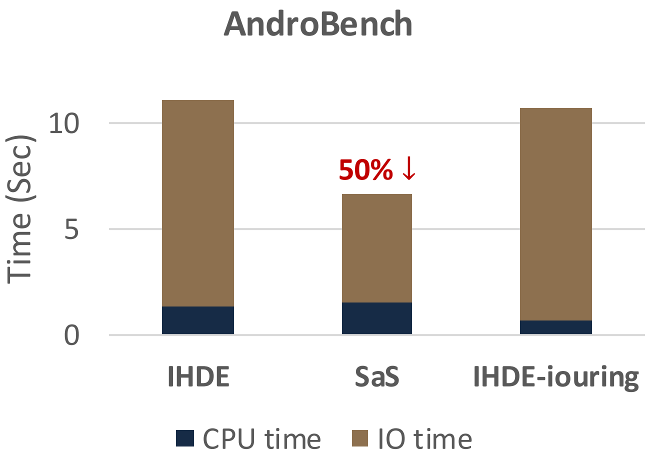


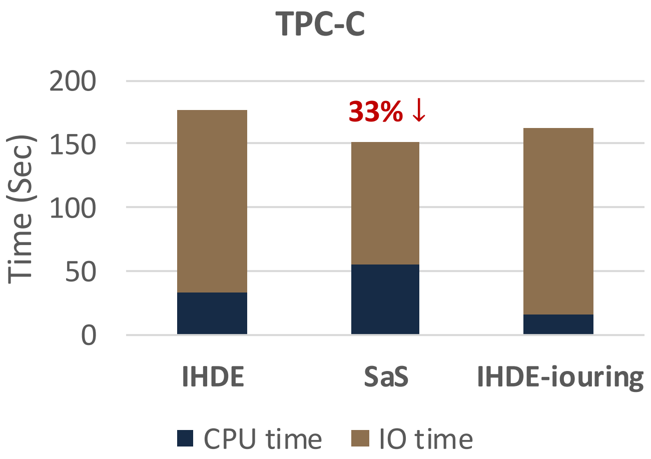


Performance Evaluation

- Source of Performance Gain in SaS
  - Bypass the kernel IO Stack
  - Memory copy reduction

- Challenges
  - Limited computing power (Intel vs. ARM)

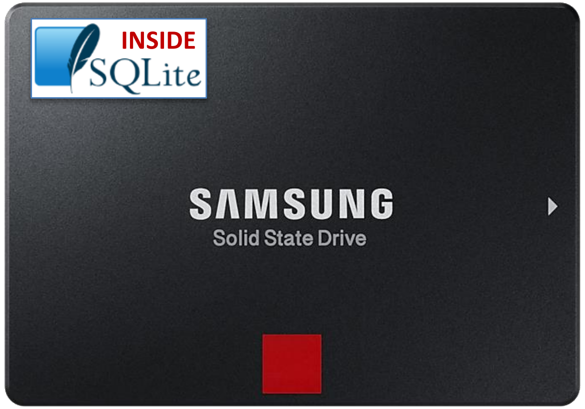




Summary

- New design alternatives
- Seamless vertical optimization
- HW-assisted accelerations

- **Target Applications**
  - Edge computing, IoT, Smart City
  - Serverless solution for small-scale blog DB



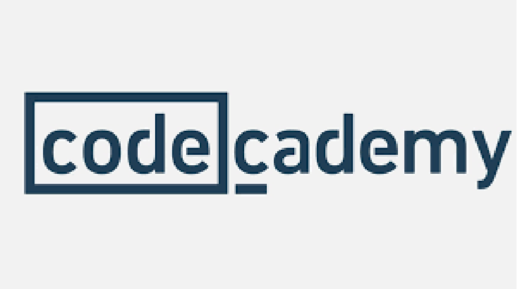
https://vldb.org/pvldb/vol14/p1481-lee.pdf

https://github.com/SSD-as-SQL-Engine/LibSaS

Some Advice …

- **WARNING!**
  - This is purely my personal opinion

Programming is not exclusive to CS people



부트캠프 광고 문구 모음.
무.조.건 취업
네카라쿠배
실용 중심의 인공지능 개발자 양성과정
영원히 나만
알고싶은 비법
지금 신청하러 가기
DevOps 부트캠프 17주 과정
전액 국비 지원 받아
커리어 전환!
#17주 완성
#전문 커리어 코칭
#100% 온라인
초봉
60,000,000
가고싶다... 네카라쿠배...!
꿈을 크게 꾸는 당신을 위한
IT기업 취보 프로젝트
고용노동부 지원 K-디지털 크레딧
<코알못을 위한 블록코딩으로 배우는 인공지능>
수강생 절찬 모집중!
선착순 모집중

Be the irreplacible Tenlent

- Devleoping your own unique skills
  - Programming (default)
  - Communication
  - Abstraction

- How..?

Do what others cannot do

- Normal users can not purchase these products!

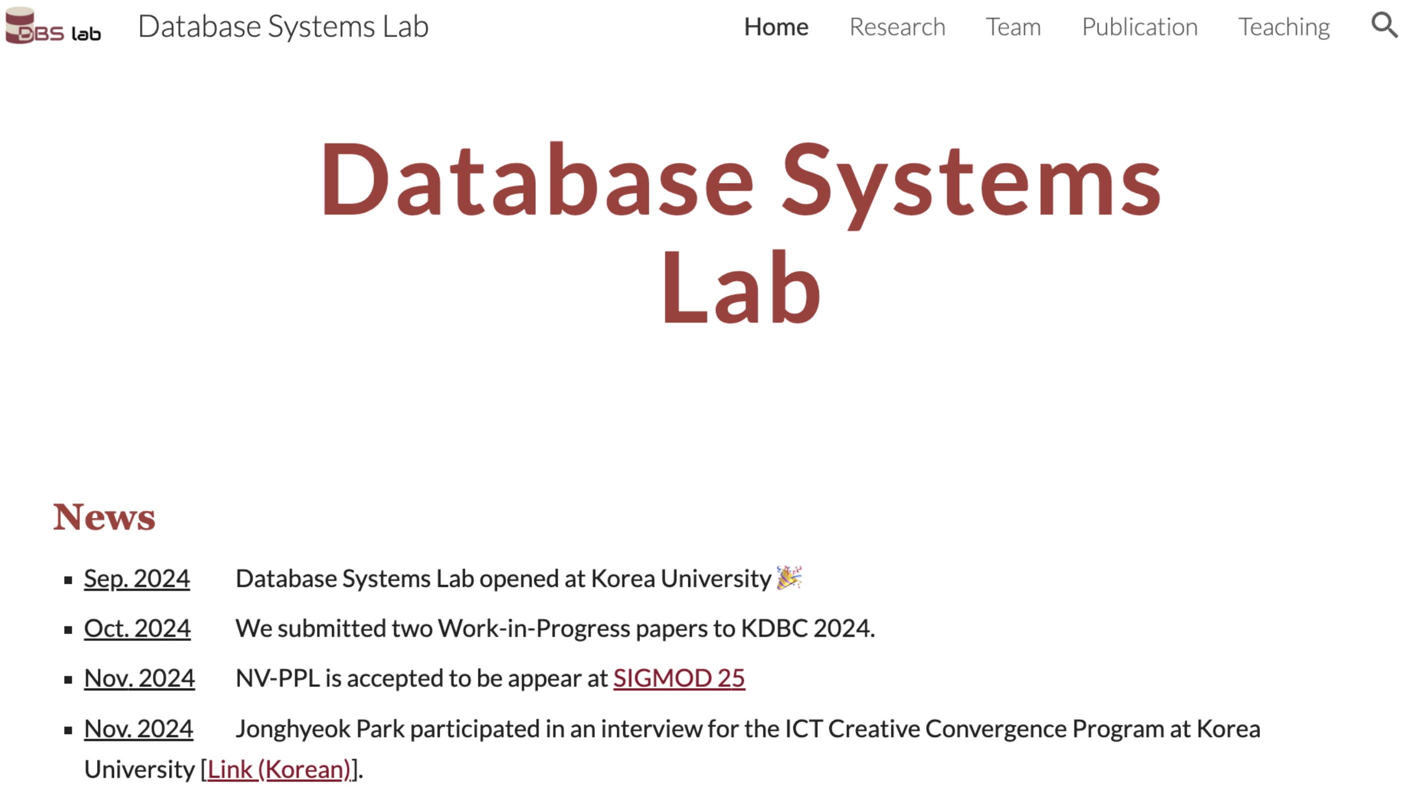
CXL (Compute Express Link)

Zoned Namespace SSD

Flexible Data Placement

DBS Lab.

- Database Systems Lab.



Database Systems Lab

News

- Sep. 2024 Database Systems Lab opened at Korea University 🎉
- Oct. 2024 We submitted two Work-in-Progress papers to KDBC 2024.
- Nov. 2024 NV-PPL is accepted to be appear at SIGMOD 25
- Nov. 2024 Jonghyeok Park participated in an interview for the ICT Creative Convergence Program at Korea University [Link (Korean)].

Research Collaborations

- SFU (NRF-Mitacs Visiting Research Internship)
- TUM
- UW
- Samsung
- SAP HANA
- Oracle

Thank You

- dbs.korea.ac.kr
- jonghyeok_park@korea.ac.kr

Join the DBS Lab.