

# LevelDB Study Introduction

2022. 07. 05

Presented by Min-guk Choi

koreachoi96@gmail.com

## **1. Study Introduction**

2. Why should we understand internal structure?

3. What is a key-value store?

4. Why open-source?

5. Job market

6. Lab research subject

7. Environment setup

8. Homework

### **■ Study Introduction**

#### **✓ Members**

- Professor, Assistant, Students

#### **✓ Schedule**

#### **✓ Goals**

#### **✓ What you can get**

# Professor

---



Jongmoo Choi

- Professor (Ph.D.), *Mar. 2003 ~ Present*
- Department of Computer Science and Engineering  
at Dankook University
- <https://sslab.dankook.ac.kr/~choijm/>



Seehwan Yoo

- Professor (Ph.D.), *Mar. 2014 ~ Present*
- Department of Mobile Systems Engineering  
at Dankook University
- <https://sites.google.com/site/dkumobileos/members/seehwanyoo>

# Assistant

---



Sounghyoun Lee

- Ph.D student
- KV Store, RT-KV Store
- wwbabaww@gmail.com



Hojin Shin

- Ph.D student
- KV Store, RT-KV Store
- ghwls03s@gmail.com



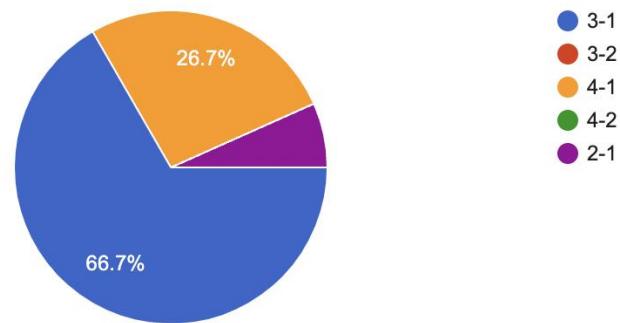
Min-guk Choi

- Undergraduate student
- KV Store, Caching
- koreachoi96@gmail.com

# Student

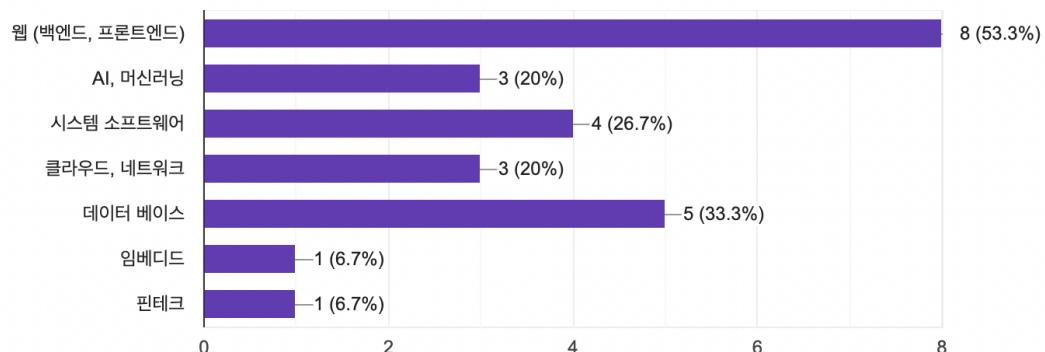
22-1학기 기준 본인의 학년을 알려주세요.

응답 15개



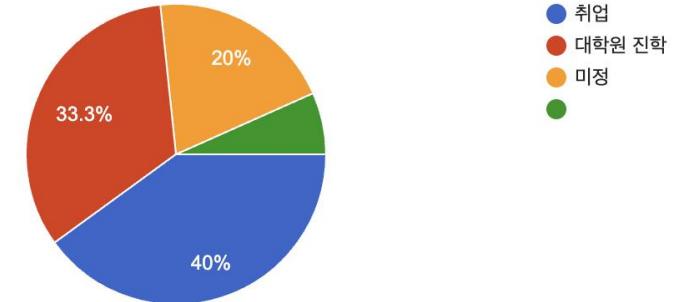
좋아하는 분야에 대해 알려주세요. (중복 가능)

응답 15개



졸업 이후 희망하는 진로에 대해 알려주세요.

응답 15개



스터디에 대해 기대하는 점이 있다면 알려주세요. (선택)

응답 6개

- 운영체제 수업에서 다소 어려웠던 개념을 다시 사용하고 새로운 지식을 얻어가는 것을 기대하고 있습니다.
- 기대하는 점은 없습니다. 잘 부탁드립니다.
- 많이 어려울 것 같지만, 그만큼 단단한 멘탈과 SW실력을 가지게 될 수 있을 것 같습니다!
- 앞으로의 진로에 관해 선배님이나 교수님들의 조언을 듣고싶습니다.
- 방학 중에 좋은 공부가 되었으면 합니다.
- 좋은 경험이 될 것 같아 기대가 됩니다!

# Schedule

- Date: Every Tuesday in July, August
- Time: 14:00 ~ 16:00
- Place: Dankook University Software ICT Hall Room 301

No	Date	Contents	
Week 1	22-07-05	LevelDB Introduction 1	What is key-value store? Why open-source?
Week 2	22-07-12	LevelDB Introduction 2	LevelDB basics Analytics tools
Week 3	22-07-19	LevelDB Analysis 1	
Week 4	22-07-26	LevelDB Analysis 2	
Week 5	22-08-02	LevelDB Analysis 3	
Week 6	22-08-09	LevelDB Analysis 4	
Week 7	22-08-16	Real-World Workload Optimization	YCSB Twitter Trace
Week 8	22-08-23	New Idea Implementation 1	
Week 9	22-08-30	New Idea Implementation 2	Write (undergraduate) research paper

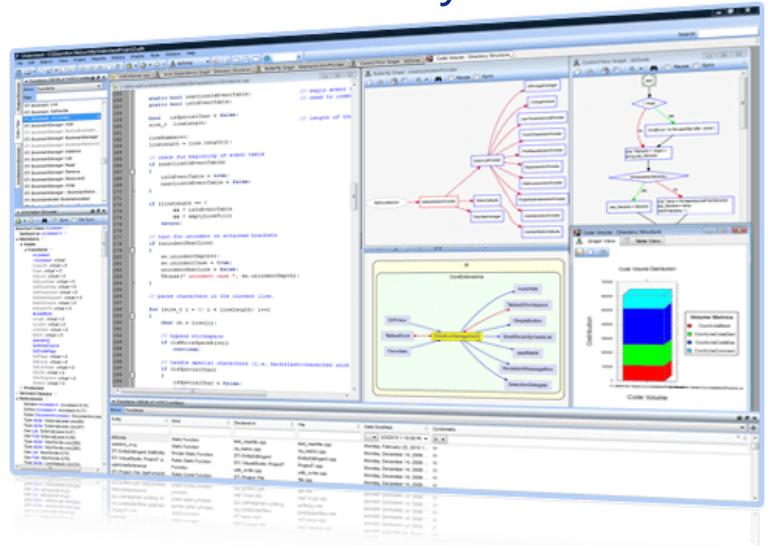
# Goals

---

- Analyze LevelDB.
- Write your LevelDB analysis into a git-book.
- Optimize LevelDB for real-world workload.
- Implement new ideas in LevelDB.
- Write a research (undergraduate) paper based on what you learned.

# Goal 1. Analyze LevelDB

- Analyze open-source project.
  - Start with studying documents and lectures
  - Learn and use analytics tools
  - Study code and remarks in small function and classes
  - Try test code
- Analytics tool
  - Debugger: gdb
  - Static analytics tools: Understand by Scitools
  - Dynamic analytics tools: utrace
- Draw figures
  - Code flowchart, Class diagram, Structure

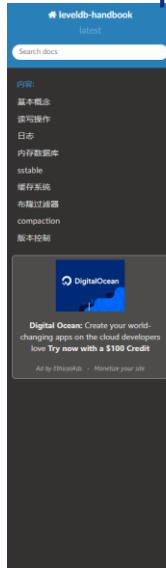


```
TOTAL TIME : FUNCTION
20.914 us : (1) a.out
20.914 us : (1) main
1.997 us :   (1) operator new
              :
4.170 us :   (1) std::shared_ptr::shared_ptr
3.797 us :   (1) std::shared_ptr::_shared_ptr
              (1) std::shared_count::_shared_count
              (1) operator new
              :
0.630 us :   (1) std::shared_count::_shared_count
              (1) operator new
              :
0.153 us :   (1) std::shared_count::_shared_count
              (1) std::enable_shared_from_this_helper
              :
10.027 us :   (1) std::shared_ptr::~shared_ptr
9.700 us :   (1) std::shared_ptr::~shared_ptr
9.374 us :   (1) std::shared_count::~shared_count
9.054 us :   (1) std::Sp_counted_base::~M_release
2.093 us :   (2) __gnu_cxx::__exchange_and_add_dispatch
0.287 us :   (2) __gthread_active_p
              :
0.260 us :   (2) __gnu_cxx::__exchange_and_add_single
              :
3.803 us :   (1) std::Sp_counted_ptr::~M_dispose
2.890 us :   (1) operator delete
              :
2.083 us :   (1) std::Sp_counted_ptr::~Sp_counted_ptr
1.697 us :   (1) std::Sp_counted_ptr::~Sp_counted_ptr
0.686 us :   (1) std::Sp_counted_base::~Sp_counted_base
0.143 us :   (1) std::Sp_counted_base::~Sp_counted_base
              :
0.284 us :   (1) operator delete

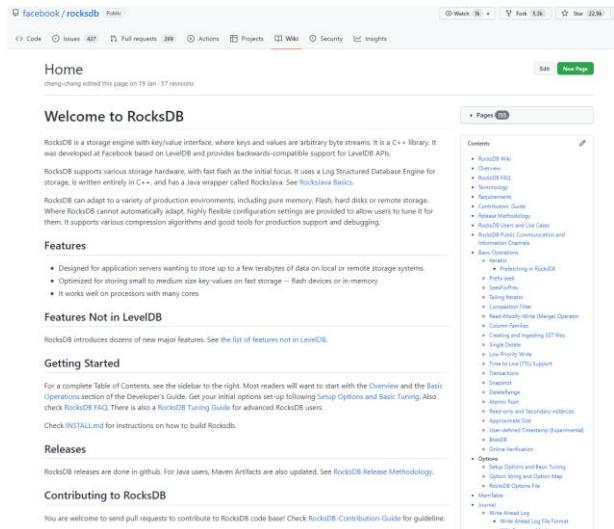
utrace graph: /home/hongyu/shared_ptr.cc [line:3]
```

# Goal 2. Git-book

- Start open-source contribution with writing a document
  - Write your LevelDB analysis into a git-book
  - Use Git-book as a portfolio
  
- Document what you have studied
  - How to organize and convey your thoughts?
  
- There are no LevelDB analysis documents written in English yet.
  - write documents in Korean during study.
  - translate documents in English after study.
  - When it finishes, promote git-book on reddit and open-source communities.



<https://leveldb-handbook.readthedocs.io/zh/latest/>



<https://dku-starlab.gitbook.io/leveldb-analysis/>

# Goal 3. Tuning

- Tune existing application/system
  - Optimize for your environment, workload
- Before tuning, you should understand
  - Whole structure
  - Relationships between parameters
  - System environment, Workload, Data set
- Tuning Contest
  - Tune LevelDB for real-world workload
  - Twitter Trace/YCSB

**kaggle**

**kakao**

MongoDB Engineer 모집

카카오 | 정규직 | 영입종료시 | 판교

- ◆ 업무내용
  - MongoDB 운영
  - MongoDB 운영 자동화 시스템 개발
  - MongoDB 장애 대응 및 트러블 투팅
  - MongoDB 모니터링/분석
  - MongoDB 쿼리/서버 설정 투팅
  - MongoDB Internal 연구
  - Database as a Service(DBaaS) 개발과 운영
  - 신기술 리서치 및 서비스 적용

<https://careers.kakao.com/jobs/P-12208>

**LINE**

Social Platform Backend Engineer

Bundang | LINE Plus | Engineering | Server-side | Full-time

**우대사항**

- 글로벌 서비스 운영 및 개발 경험이 있으신 분
- k8s 환경에서 서비스를 개발한 경험이 있으신 분
- Kotlin, GO 언어 등의 기타 언어에도 익숙하거나 또는 도입에 거부감이 없으신 분
- 성능 개선을 위해 어플리케이션 혹은 시스템 투팅에 도전해보셨거나 관심이 많으신 분
- 외국어로 업무 진행이 가능하신 분 (영어, 일본어, 중국어)

<https://careers.linecorp.com/ko/jobs/1078>

# Goal 4. New Idea Implementation

---

- Implement on existing project
  - Your own idea, optimization, customization
  - Idea from papers
  
- How many people do companies need?
  - Many
  
- How many people have done this?
  - Not many
  
- Who can do this?
  - Someone just like you.
  - The professors and assistants will support you.



**nbase-arc**

Distributed memory store based on Redis

[naver / nbase-arc](#) Public

nbase-arc is an open source distributed memory store based on Redis

Apache-2.0 license

162 stars 38 forks

★ Star Watch ▾

<https://naver.github.io/>

<https://github.com/naver/nbase-arc>

# Goal 5. Research Paper

---

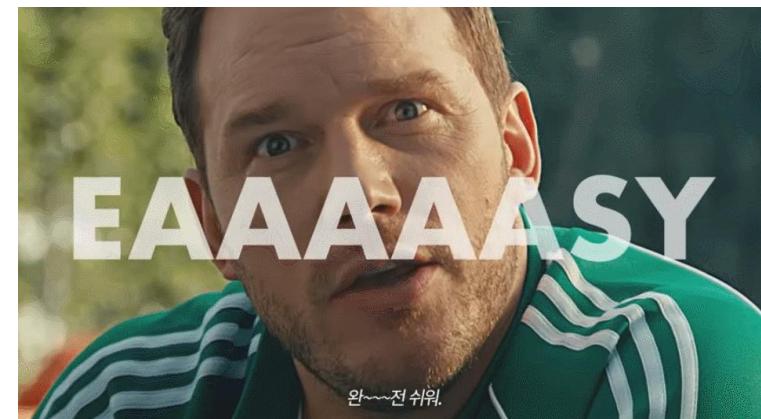
- How can I appeal my research experience?
  - Git-hub and blog are ok.
  - Nevertheless, writing paper is the best.
  
- Write (an undergraduate) research paper
  - It's is not as hard as you think.
  - Just write it like a homework report.
  - Check how seniors wrote paper before.
  
- The professors and assistants will support you.



KSC2021

2021년 12월 20일(월)-22일(수), 휘닉스 평창 호텔&온라인

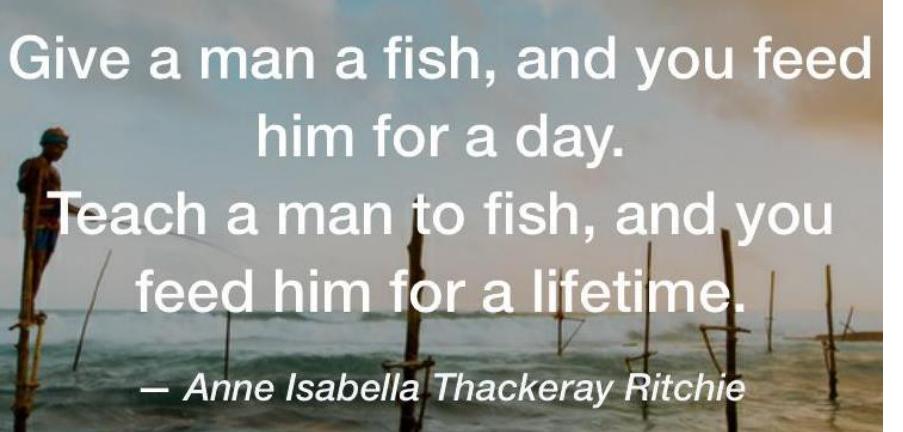
"코로나 이후의 세상: 소프트웨어 중심 사회"



[https://www.youtube.com/watch?v=Q39VK9QAbG4&ab\\_channel=FortniteKR](https://www.youtube.com/watch?v=Q39VK9QAbG4&ab_channel=FortniteKR)

# What you can get

- KV store
  - Knowledge, Tuning, Implementation
- Open-source
  - Learn how to analyze and document the project
  - Start open-source contribution by writing a git-book
- Portfolio
  - Git-book, Tuning Contest, Research paper



Give a man a fish, and you feed  
him for a day.  
Teach a man to fish, and you  
feed him for a lifetime.

— Anne Isabella Thackeray Ritchie

<https://unquote.li/>

1. Study Introduction

**2. Why should we understand internal structure?**

3. What is a key-value store?

4. Why open-source?

5. Job market

6. Lab research subject

7. Environment setup

8. Homework

- Why should we understand internal structure?
  - ✓ Indexing in RDB
  - ✓ Coding interview question
  - ✓ B+-tree vs LSM-tree

# Indexing in RDB

- Indexing in RDBMS

Mem (B+-tree)

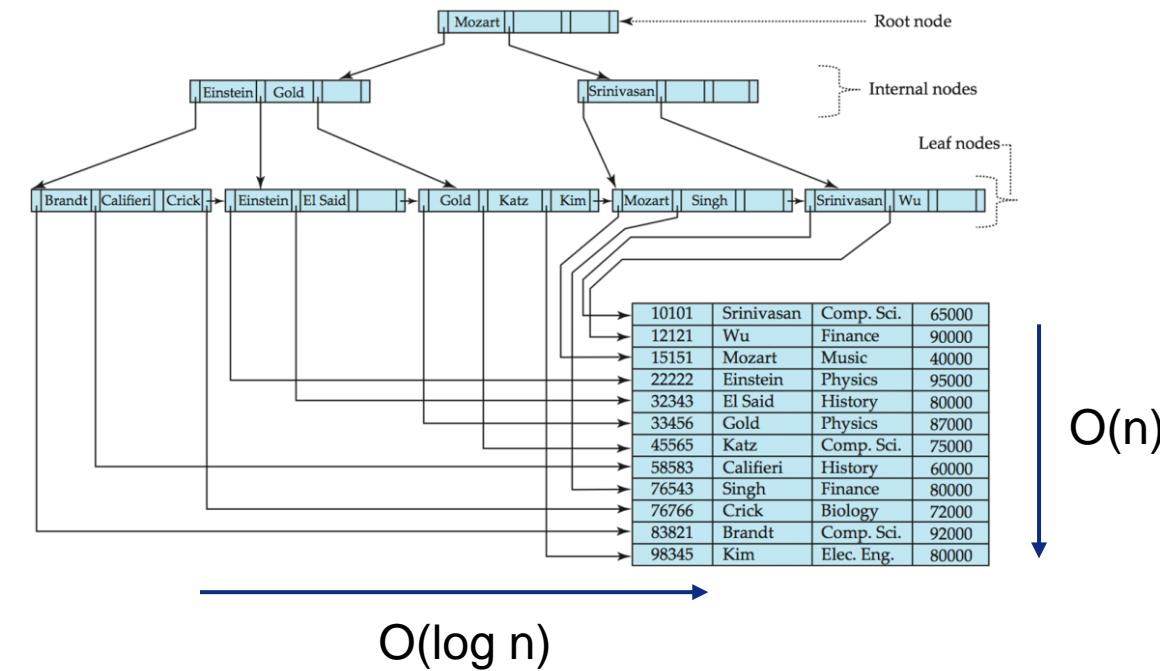
Index Table

Biology
Comp. Sci.
Elec. Eng.
Finance
History
Music
Physics

Pointers

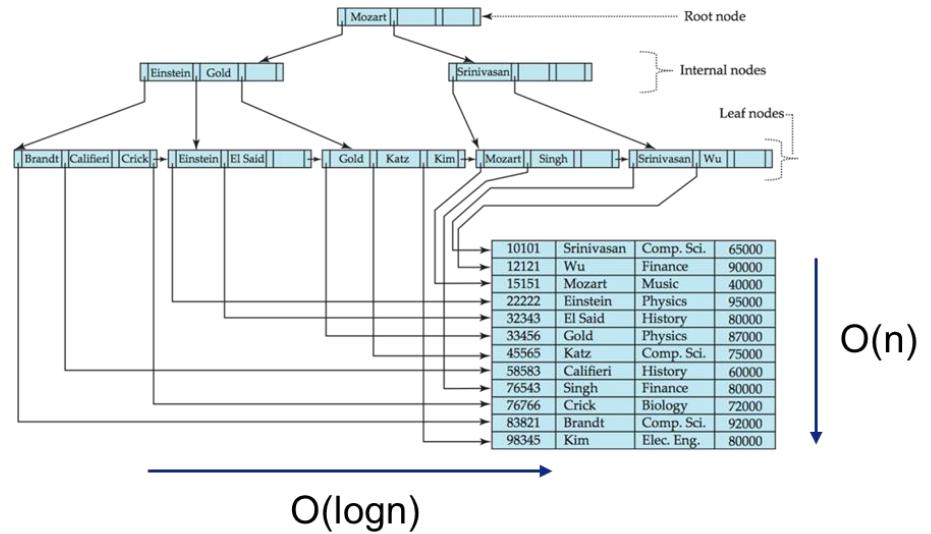
ID	Name	Discipline	Salary
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
33465	Gold	Physics	87000

Disk (File)

Database System Concept 7<sup>th</sup> Edition, Avi SilberschatzDatabase System Concept 7<sup>th</sup> Edition, Avi Silberschatz

# Coding interview question

- Q1. Which tables should be indexed?
  - Hint. Write (*INSERT, UPDATE, DELETE*) / Read (*SELECT*)
  - Hint. B-tree, Rebalancing
  
- Q2. Which column should be indexed?
  - Hint 1. *SELECT \* WHERE Name = "Giraffe"*;
  - Hint 2. Join/Where/Order by, Cardinality
  
- Q3. Should we use index or not? (Index: weight)
  - 1) *SELECT \* WHERE Weight <=30 OR Height <= 40;*
  - 2) *SELECT \* WHERE Weight >=100;*
    - Hint. Disk Random/Sequential Access
  - 3) *SELECT AVG(Weight) WHERE Weight >= 30;*
    - Hint. Disk/Mem Access speed



Database System Concept 7<sup>th</sup> Edition, Avi Silberschatz

<b>Id</b>	<b>Name</b>	<b>Height</b>	<b>Weight</b>	<b>Age</b>
1	Cat	30	5	3
2	Dog	30	10	2
3	Bear	100	100	5
4	Rabbit	30	20	1
100,000	Giraffe	200	200	10

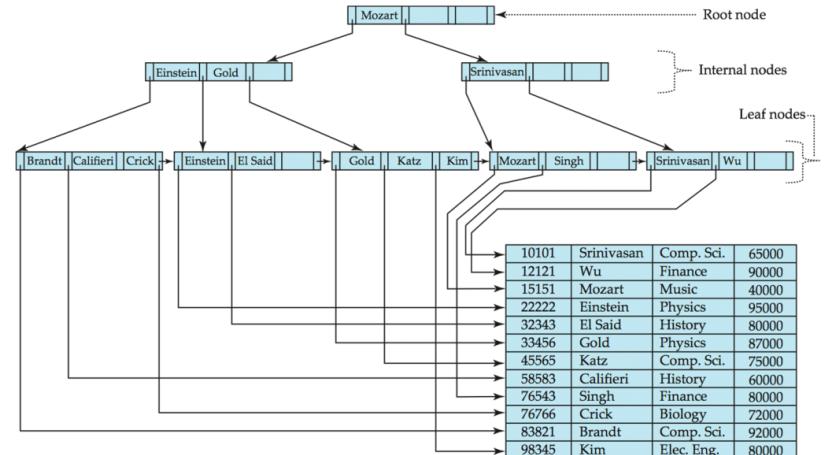
# Coding interview question

- How should we answer the previous question?
  - Based on structure/architecture (B+-tree)
  - OS/DB kernel knowledge
  
- Why does the interviewer ask such questions?
  - Performance, efficiency, cost
  
- So, understanding how internal works is very important.

Index Table

	ID	Name	Discipline	Salary
Biology	76766	Crick	Biology	72000
Comp. Sci.	10101	Srinivasan	Comp. Sci.	65000
Elec. Eng.	45565	Katz	Comp. Sci.	75000
Finance	83821	Brandt	Comp. Sci.	92000
History	98345	Kim	Elec. Eng.	80000
Music	12121	Wu	Finance	90000
Physics	76543	Singh	Finance	80000
	32343	El Said	History	60000
	58583	Califieri	History	62000
	15151	Mozart	Music	40000
	22222	Einstein	Physics	95000
	33465	Gold	Physics	87000

Pointers



# B+-tree vs LSM-tree

- Study how LSM-tree based DB works!

## DATABASE STORAGE ENGINES

### B-TREE



### LSM TREE



<https://blog.yugabyte.com/a-busy-developers-guide-to-database-storage-engines-the-basics/>

1. Study Introduction

2. Why should we understand internal structure?

3. *What is a key-value store?*

4. Why open-source?

5. Job market

6. Lab research subject

7. Environment setup

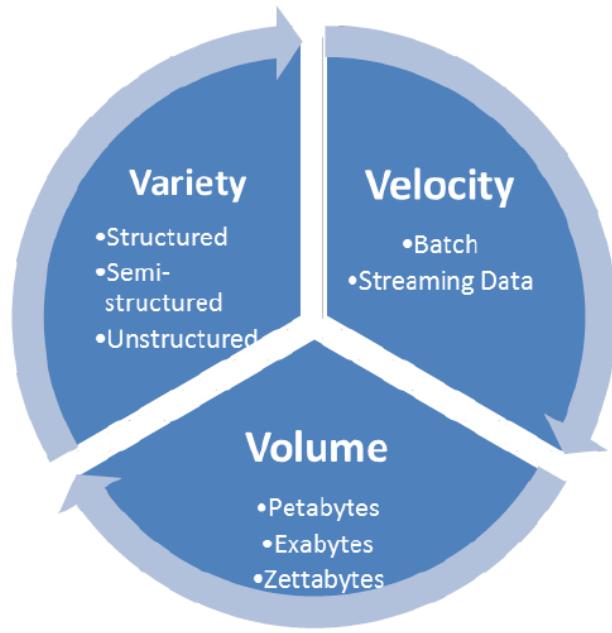
8. Homework

- **What is a key-value store?**

- ✓ What is Big Data?
- ✓ Structured vs Unstructured data
- ✓ Key-Value Store
- ✓ Level DB

# What is Big Data

- 3 Vs of Big Data: Volume, Velocity, Variety
  - Extended to 5 or 8 Vs



[https://www.researchgate.net/figure/BM-Big-Data-characteristics-3V-Adopted-from-Zikopoulos-and-Eaton-2011\\_fig1\\_258247680](https://www.researchgate.net/figure/BM-Big-Data-characteristics-3V-Adopted-from-Zikopoulos-and-Eaton-2011_fig1_258247680)



<https://shreyanshmathur1998.medium.com/big-data-with-8vs-and-todays-challenges-9938f00363c5>

# What is Bigdata

- Volume and Velocity Perspective
  - Data gathered while driving



✓ 25GB per hour (from IEEE AutoSafety, 2020)

# What is Bigdata

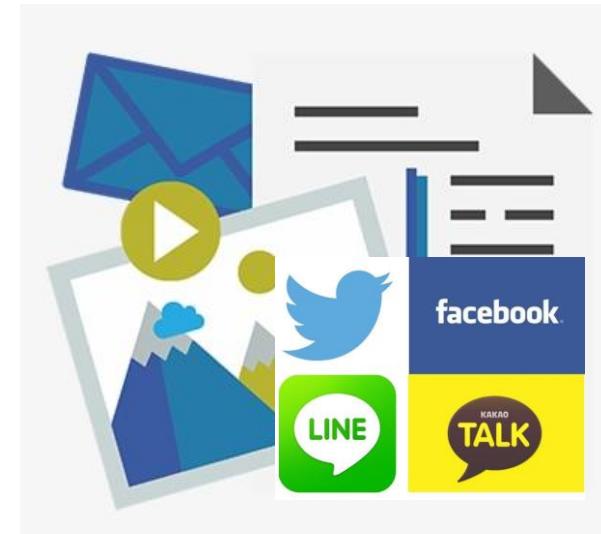
---

- How about Variety?
  - Conventional data: well-defined format
  - Big data: various and diverse
    - Various fields/frequencies in each record, Diverse format ...

Column ↓

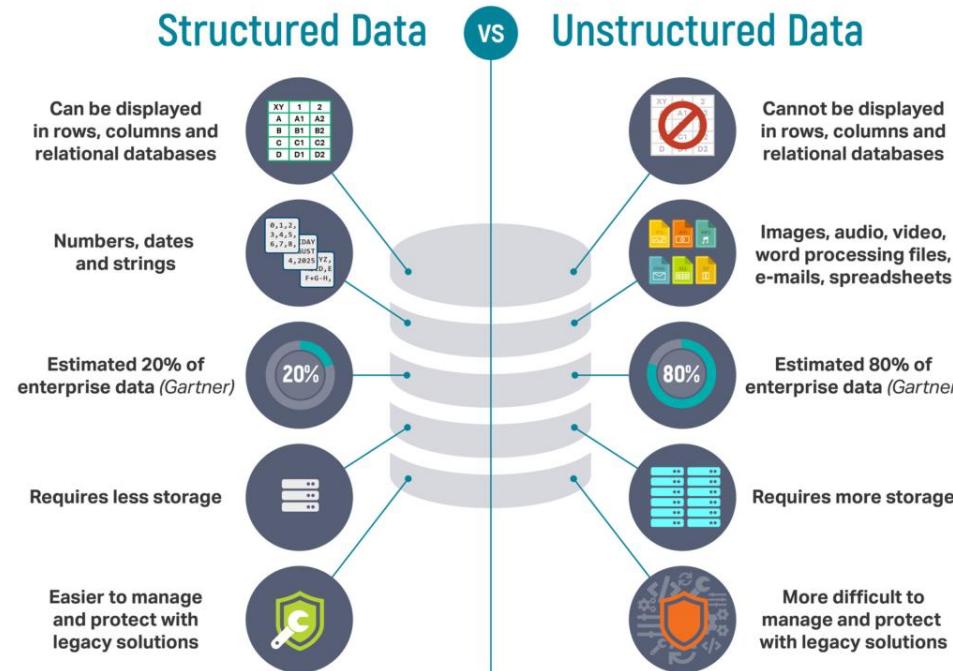
Attributes →	Roll_No	Name	Age	GPA
Tuple →	1	Arya	21	4
	2	Bran	19	3
	3	John	24	4.3
	4	Max	24	1

<https://prepinsta.com/dbms/rdbms/>

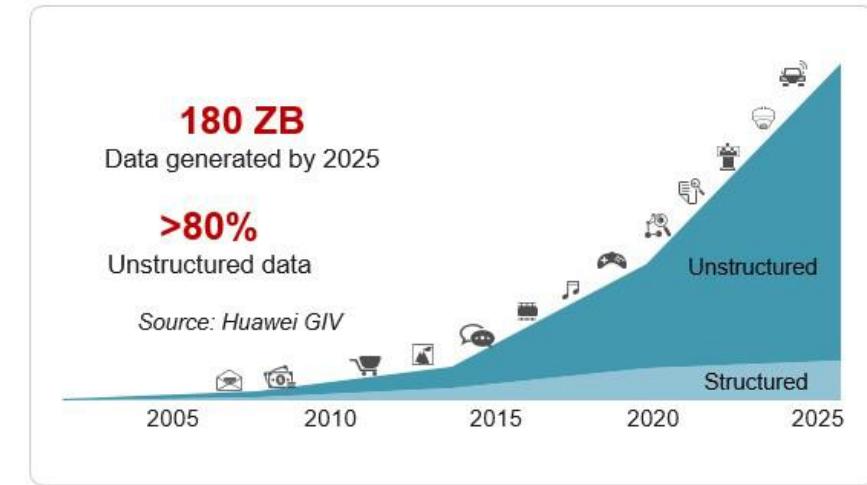


# Structured vs. Unstructured data

- Types of Data



## New Connections Drive Explosive Data Growth

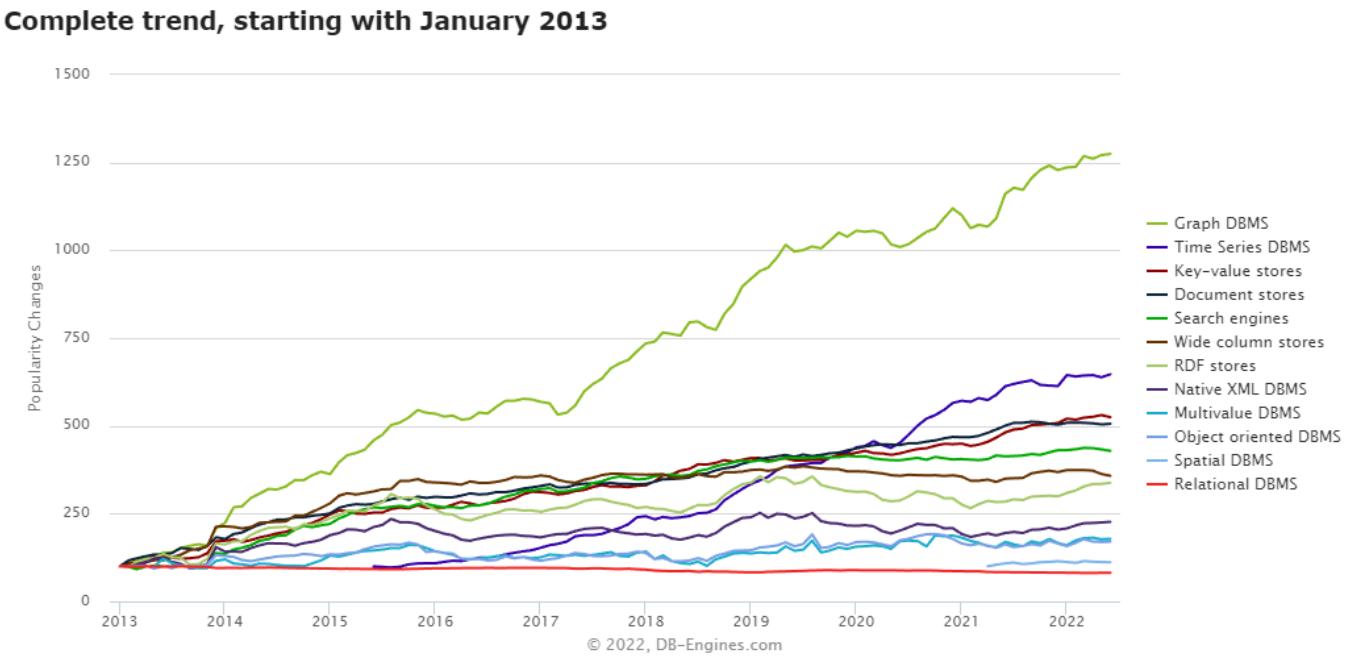


<https://lawtomated.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/>

<https://e.huawei.com/kr/products/storage/topic/2021/all-flash-data-center-green-energy>

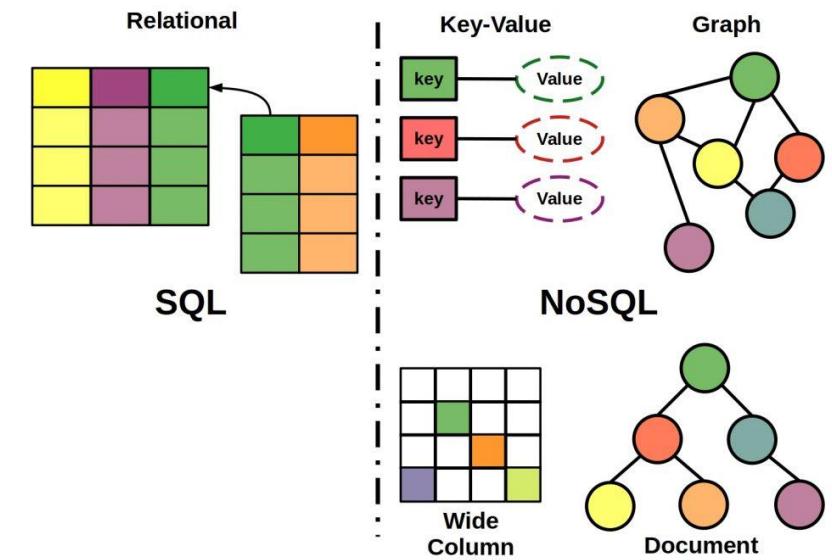
# Structured vs. Unstructured data

- Types of Database
  - SQL DB, RDB (Relational DB) for structured data
  - NoSQL(Not-Only SQL) DB, Non RDB for unstructured data



# Structured vs. Unstructured data

- Types of NoSQL Database
  - Key-Value DB
    - Manage key and value pair
    - Ex. LevelDB, RocksDB, Redis ...
  - Document DB
    - Extend KV store, KVs are stored in organized format (like XML)
    - Ex. MongoDB, Couchbase
  - Column-family (wide column) DB
    - Column-oriented KV management (sparse)
    - Ex. Bigtable, Hbase, Cassandra
  - Graph DB
    - KV pairs and their relationships
    - Ex. Neo4j, JanusGraph
  - Time Series DB
    - time series through associated pairs of time(s) and value(s)
    - Ex. Influx DB, Apache Druid



<https://starship-knowledge.com/when-to-use-nosql-vs-sql>

# Key-Value Store

- Key-Value Stores (a.k.a Key-Value DB)
  - A de-facto standard DB for unstructured data
  - Google, Facebook, Amazon, Microsoft, MongoDB, Yahoo, Hbase, LinkedIn, Oracle, Baidu, Basho, In Memory DB (Memcached), ...



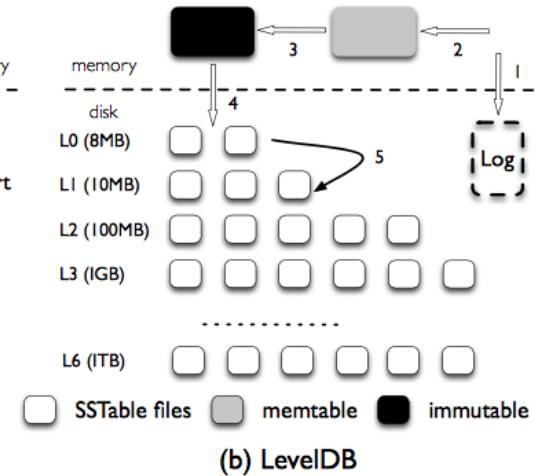
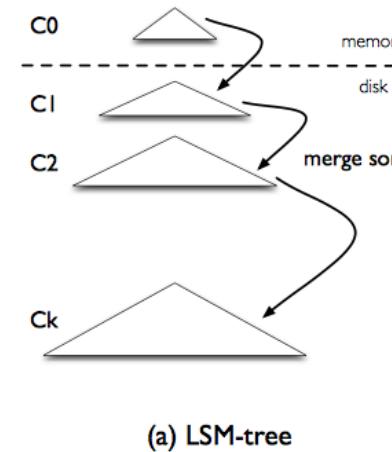
# Key-Value Store

- Key-Value Stores: some examples
  - LevelDB
    - By Google, 2011, a subset of Bigtable (Column-oriented DB, OSDI, 2006)
    - Level compaction, Open-source
  - RocksDB
    - By Facebook, 2012, a fork of LevelDB
    - Various algorithms (e.g. Tier compaction), High performance, Diverse applications
  - Hbase
    - By Apache, 2008, motivated by Google's Bigtable
    - A distributed data storage system for the Hadoop ecosystem
  - Cassandra
    - By Apache (Facebook), 2008, motivated by Amazon's Dynamo
    - A decentralized architecture, each node is powered by LST-tree based DB



# Key-Value Store

- LevelDB
  - an open source on-disk key-value store
  - written by Google fellows  
Jeffrey Dean and Sanjay Ghemawat
  - By Google, 2011, a subset of Bigtable
    - minimal dependencies, suitable for open sourcing
  - a light-weight, single-purpose library  
for persistence with bindings to many platforms. – [leveldb.org](http://leveldb.org)



"Wisckey: Separating keys from values in ssd-conscious storage." TOS '17

# Key-Value Store

- Projects using LevelDB

Google Cloud Bigtable



1. Study Introduction

2. Why should we understand internal structure?

3. What is a key-value store?

**4. Why open-source?**

5. Job market

6. Lab research subject

7. Environment setup

8. Homework

## ■ Why Open-Source?

- ✓ Trend of open-source
- ✓ Company
- ✓ Developer
- ✓ How to start open-source
- ✓ Most important language for open-source

# Trend of open-source

- 2016
  - Alpha Go vs Lee Sedol
- 2015
  - Many Company have released open-source machine learning framework
  - Google, Facebook, Samsung, IBM, MS ...
- Machine learning framework which is not open-source?
  - No one uses it

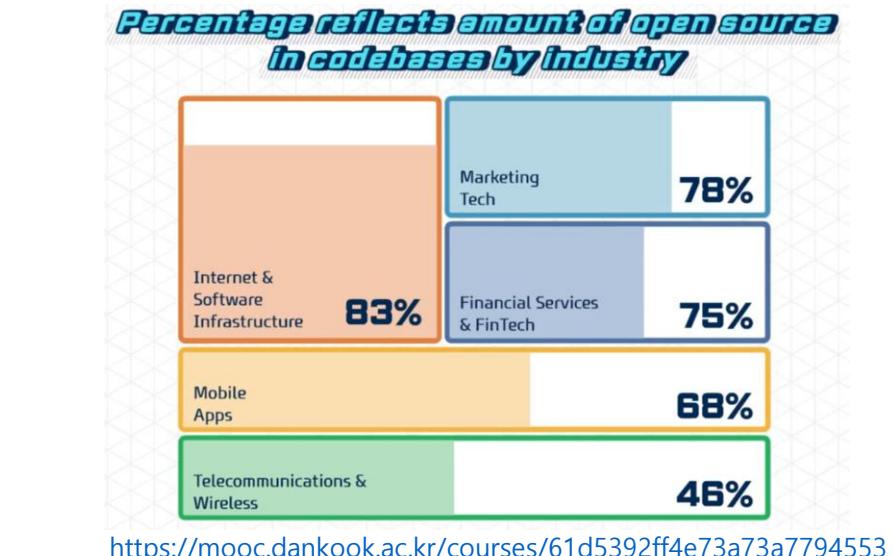
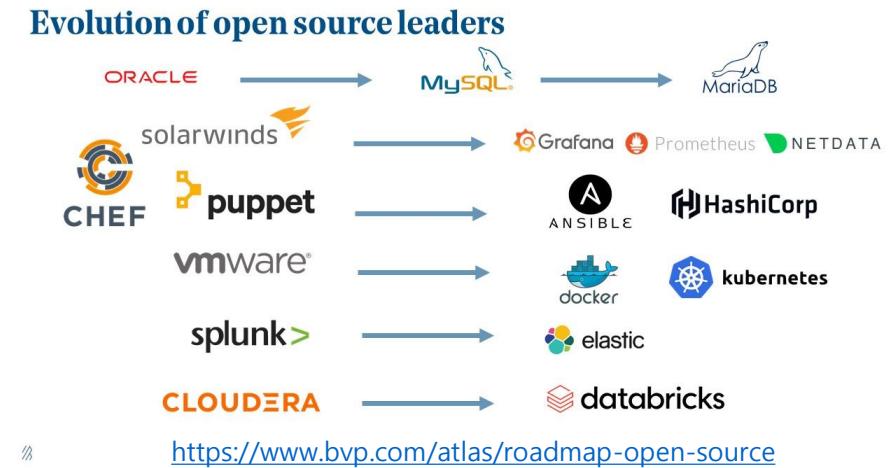


<http://m.baduk.or.kr/>



# Trend of open-source

- Past
  - Cheaper version of closed source software
  
- Present
  - Superior alternative offering higher quality
  - Better support, and more flexibility.
  - Started in infra structure, data analytics and AI
  
- In emerging industries,
  - Companies open-source their projects
  - Companies use and contribute to open-source



# Company

---

- Big Tech
  - Software/cloud platform
  
- Service Startup
  - Fast development
  - Cloud service
  
- Manufacturer
  - Linux kernel

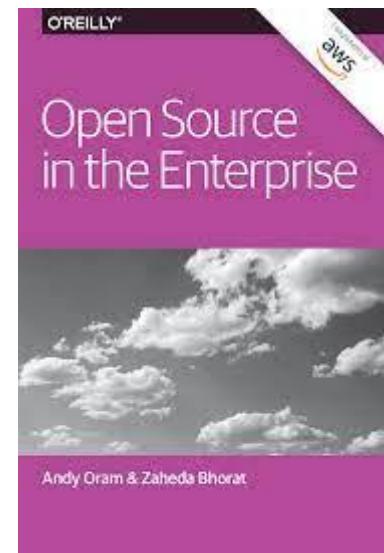
RANK	ORGANIZATION	ACTIVE CONTRIBUTORS	TOTAL COMMUNITY	
1	Google	4643 +414	9706 +689	>
2	Microsoft	4394 +486	9515 +903	>
3	Red Hat	3298 +198	4956 +226	>
4	Intel	1837 +172	4027 +339	>
5	IBM	1720 +188	4129 +414	>
6	Amazon	1512 +187	3869 +420	>
7	Facebook	1140 +103	4288 +567	>
8	GitHub	975 +120	2460 +270	>
9	VMware	874 +72	1587 +112	>
10	SAP	739 +76	1519 +148	>
37 ↘ 2	Samsung	217 +23	610 +60	>
89 ↗ 4	LG Electronics	62 +9	199 +29	>
103 ↗ 5	LINE Corporation	53 +9	171 +14	>

Open-Source Contributor Index  
[\(https://opensourceindex.io/\)](https://opensourceindex.io/)

# Company

---

- Multiplying the company's investment
- Benefiting from the most recent advancement
- Spreading knowledge of the software
- Increasing the developer base
- Upgrading internal developer skills
- Building reputation
- Recruiting and retaining developer

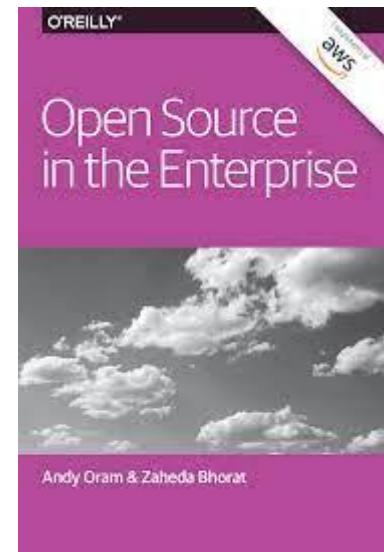


Ref. Open Source in the enterprise,  
Andy Oram & Zaheda Bhorat

# Developer

---

- Fun and happy
- Learn to code
- Gain early experience
- Experiencing critical mass
- There are good eyes on who is contributing what
- Positions of public trust
- Learn to communicate positively
- It's idealistic and the way of the future

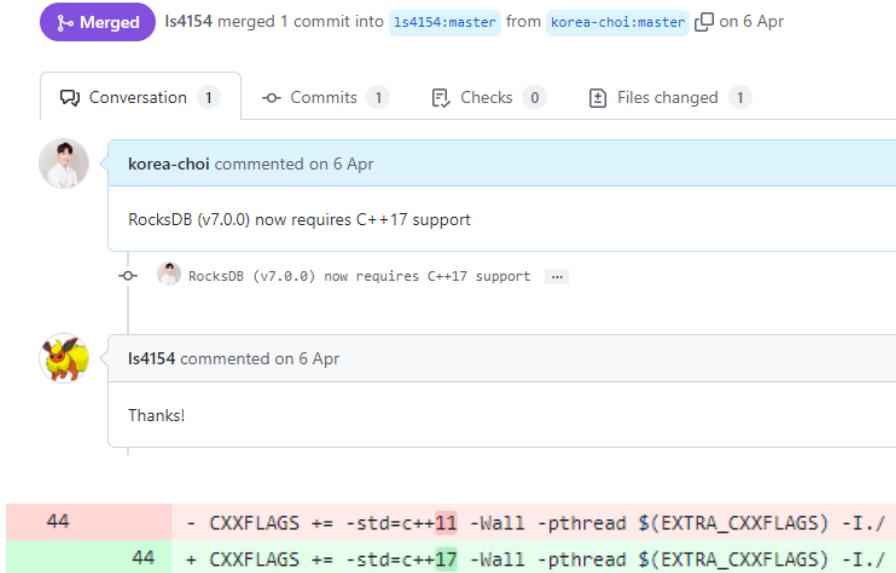


Ref. Open Source in the enterprise,  
Andy Oram & Zaheda Bhorat

# How to start open-source?

- You don't have to contribute code

RocksDB (v7.0.0) now requires C++17 support #1



Fix typos in remarks or documentation

<https://github.com/ls4154/YCSB-cpp/pull/1>

```

+ # Add -pg option for uftrace
+ set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -pg")

if(CMAKE_CXX_COMPILER_ID STREQUAL "MSVC")
    # Disable C++ exceptions.

```

Giving guidance on a specific project and how to use it

<https://github.com/namhyung/uftrace/wiki/uftrace-for-LevelDB-RocksDB-with-YCSB>

# How to start open-source?

- You don't have to contribute code

The screenshot shows the 'LevelDB Analysis' page from the LevelDB WiKi website. The page title is 'LevelDB Analysis'. On the left, there is a sidebar with a navigation menu containing links to 'Overview', 'Basic Operations', 'Memtable', 'WAL', 'Compaction', 'SSTable', 'Bloom Filter', and 'Cache'. Below the sidebar, there is a section titled 'Next Overview' and a note indicating the page was last modified 3 days ago.

Write Document

The screenshot shows a GitHub pull request titled 'leveldb properties option bug fix #2'. The pull request has been merged. The commit message is: 'Is4154 merged 1 commit into ls4154:master from korea-choi:develop' on 20 May. The commit author is 'korea-choi' and the committer is 'ls4154'. The commit message details a bug in the LevelDB properties options where an integer underflow occurred on default values. It includes a code snippet: `size_t cache_size = std::stol(props.GetProperty(PROP_CACHE_SIZE, PROP_CACHE_SIZE_DEFAULT))`. The pull request also includes a comment from 'ls4154' thanking 'korea-choi' for the PR and deleting the `leveldb.properties` file from the commit.

Bug Fix

<https://github.com/ls4154/YCSB-cpp/pull/2>

# Most important language for open-source

4. Why open-source? (8/8)

- English

[잡담] 리눅스 배우는데 가장 도움이 되는 언어 | 자유 게시판

2009.05.21

리눅스를 배우는데 또는 사용하는데 알고 있으면 도움이 되는 언어가 참 많지요.

shell script, perl, python, c, c++ 등등...

그런데 그 중에서도 가장 도움이 되는 건(아마도 거의 필수적일 것 같습니다)

영어인 것 같습니다.

- Don't have to be perfect

- Writing a correct code is more important

최소한 man page 만 읽어봐도, 아니면 googling만 해봐도 금방 답을 찾을 수 있는 질문들이 여러 사람들에 의해 반복적으로 올라오는 걸 보고 혼자 놔두리를 적어봅니다.

문제해결을 위해 혼자서 관련 문서를 찾아보지 않고 그냥 게시판에 틀어 보는 분들은 아마도 리눅스 한번 설치 해보고 금방 그만 두실 가능성이 높으리라 생각합니다. 그러다 몇달 후에 다시 생각나서 요즘 어느 배포판이 좋냐는 질문부터 시작해서 X가 설치 안되요라는 질문으로 무한 루프를 돌게 되는 거죠.

- Just don't be afraid

- Google Translator, Papago
- Ph.D. Charles will help you!



LG전자 거쳐 구글로 간 김남형 씨  
오픈소스 프로젝트 '리눅스 커널' 개발 경험이 취업에 도움돼  
소프트웨어 개발과 불가분...기업 최신 기술 도입 통로  
"완벽한 영어 구사할 필요없어, 올바른 코드 작성이 더 중요"

1. Study Introduction

2. Why should we understand internal structure?

3. What is a key-value store?

4. Why open-source?

**5. Job market**

6. Lab research subject

7. Environment setup

8. Homework

- **Job market**

- ✓ KCC X KAKAO
- ✓ Samsung DS
- ✓ SK Hynix
- ✓ Naver
- ✓ LINE

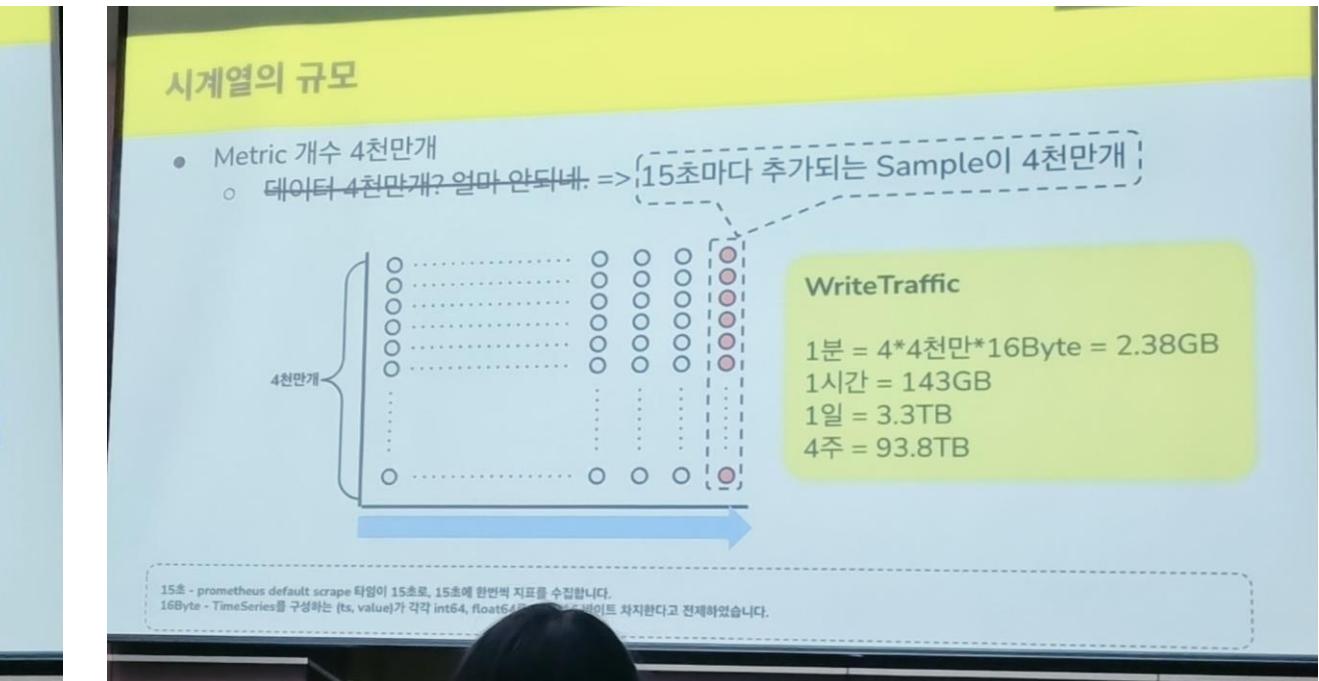
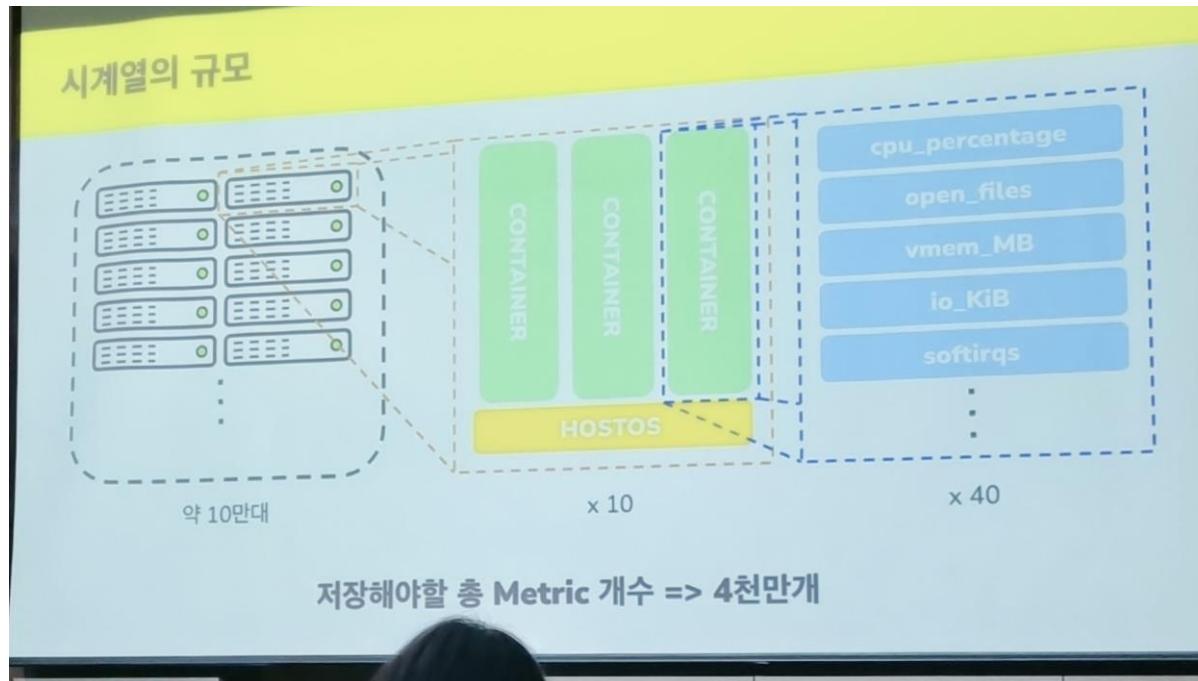
# KCC 2022 X KAKAO

15:00 ~ 15:30	SPDK를 사용한 클라우드 볼륨 스토리지 가속화	손성국
15:30 ~ 16:00	클라우드 지표 저장을 위한 대용량 TSDB 만들기	김연우
16:00 ~ 16:30	카카오 대용량 로그 파이프라인을 위한 시스템	윤상민

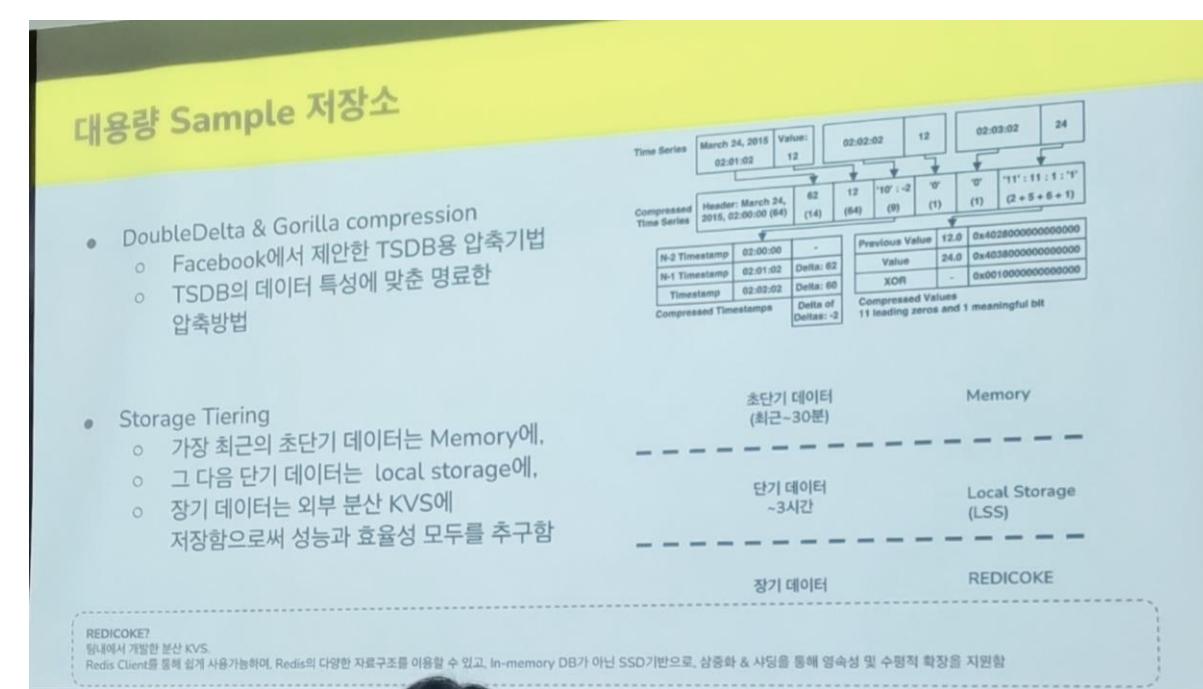
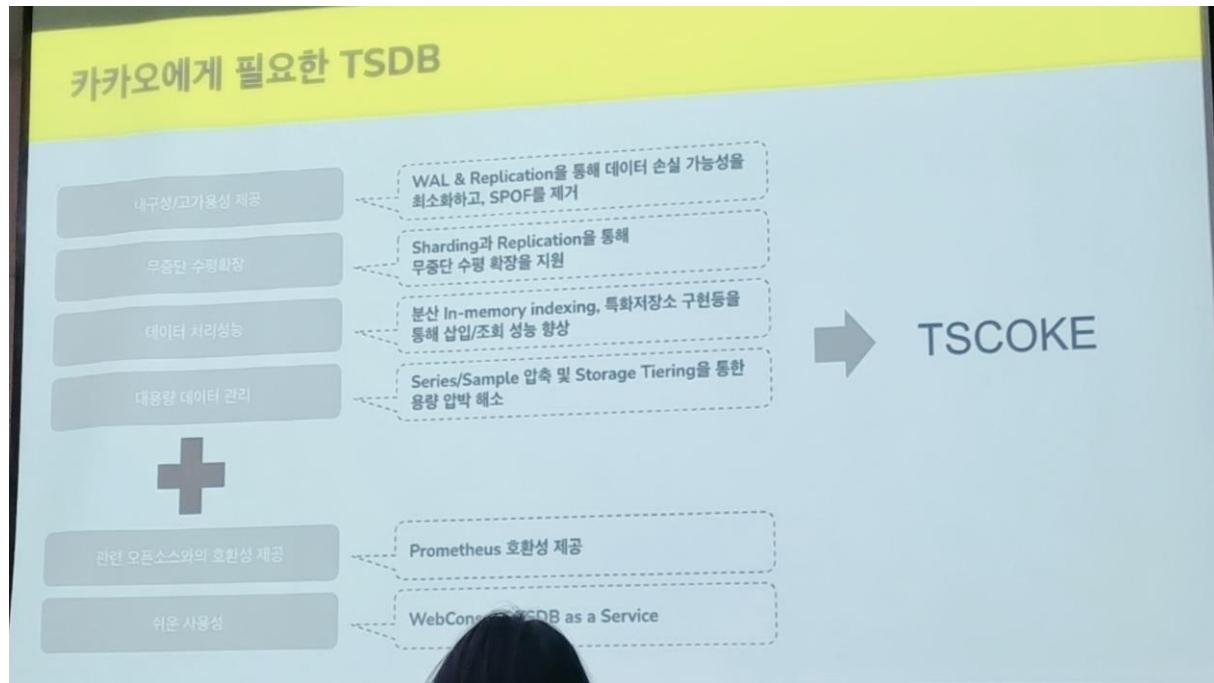


[https://www.kiise.or.kr/conference/main/getContent.do?CC=KCC&CS=2022&PARENT\\_ID=011600&content\\_no=1595](https://www.kiise.or.kr/conference/main/getContent.do?CC=KCC&CS=2022&PARENT_ID=011600&content_no=1595)

# KCC 2022 X KAKAO



# KCC 2022 X KAKAO



# KCC 2022 X KAKAO

**분산 In-memory Series 인덱스**

**Dictionary Coding**  
Series Name의 중복된 Symbol들을 Dictionary Coding  
-> 공간 절약 및 성능 향상

**ART(Adaptive Radix Tree)도입**  
ART(Adaptive Radix Tree)도입  
-> Dictionary Coding을 통해, Unbalance Tree인 Radix Tree의 단점이 최소화됨

mem{method=free,instance=10.0.20.3}  
mem{method=pgfault,instance=10.0.20.3}  
disk{method=free,instance=10.0.20.3}  
mem{method=pgfault,instance=10.0.20.4}  
mem{method=free,instance=10.0.20.4}  
disk{method=free,instance=10.0.20.4}

1(>=3,4=5)  
1(>=6,4=5)  
7(>=3,4=5)  
7(>=6,4=5)  
1(>=6,4=9)  
1(>=6,4=9)  
7(>=3,4=9)  
7(>=6,4=9)

mem=1,method=2,free=3,instance=4,10.0.20.3=5,  
pgfault=6,disk=7,used=8, 10.0.20.4=9

Fig. 1. Adaptively sized nodes in our radix tree.

**대용량 Sample 저장소**

- LSS(Log-structured Segment Storage)
  - SSD에 최적화된 Append only storage
  - 비슷한 생존시간을 가진 Segment들을 Block에 묶어 저장하고 이를 Rollout시킴으로써, 안정적이고 빠른 성능을 제공함

Segment - Sample을 압축해 저장하는 고정크기(512Byte) ByteArray

# Samsung DS

## ■ Jobs

- Software architecture
- System software
- System development

## ■ '22 Open Recruitment

- [https://www.samsung-dsrecruit.com/tnc\\_2022\\_6\\_jd](https://www.samsung-dsrecruit.com/tnc_2022_6_jd)

## Storage System SW

담당 업무	필요 역량	우대 역량
<ul style="list-style-type: none"> <li>■ <u>PCIe, NVMe 등 Storage Protocol Spec 연구/개발</u></li> <li>■ Data Center 표준 Spec 연구/개발</li> <li>■ <u>Storage 시스템, 분산시스템, 네트워크, 클라우드, 서버 연구/개발</u></li> <li>■ <u>SSD 성능 분석 (host-to-device) 연구/개발</u></li> <li>■ Data-Intensive 기술</li> <li>■ <u>Linux Kernel / Device Driver 연구/개발</u></li> <li>■ <u>Server / Client Application 연구/개발</u></li> </ul>	<ul style="list-style-type: none"> <li>■ 전공 관련           <ul style="list-style-type: none"> <li>- 컴퓨터공학 전공, 스토리지 시스템/분산 시스템 분야 Lab</li> <li>- OS/컴퓨터구조/알고리즘 교과목 이수</li> </ul> </li> <li>■ Knowledge 관련           <ul style="list-style-type: none"> <li>- <u>스토리지 시스템/분산 시스템 개발</u> 경력 보유</li> <li>- 클라우드 서비스 개발 경력 보유</li> </ul> </li> <li>■ Skill 관련           <ul style="list-style-type: none"> <li>- C/C++/Rust 프로그래밍 활용</li> <li>- 영어회화 중급자</li> <li>- Linux 개발 환경</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>■ Knowledge 관련           <ul style="list-style-type: none"> <li>- <u>오픈소스 Contribution 경험자</u></li> </ul> </li> </ul>

# SK Hynix

---

- Jobs
  - System software
  - Solution software
  - Application Engineering

- Job Introduction

- [https://recruit.skhynix.com/servlet/reab\\_jobIntroduce1.view](https://recruit.skhynix.com/servlet/reab_jobIntroduce1.view)



## System-level Architecture Analysis

주요수행업무

- 주요 수행 업무
  - 데이터센터/클라우드 컴퓨팅 환경 상에서의 벤치마크 기반 메모리 성능 분석
  - 차세대 응용 기반 워크로드/벤치마크 발굴 및 특성 분석
  - 차세대 메모리 솔루션 아키텍처 개발
  - 아키텍처 분석/검증을 위한 차세대 메모리 모델링 및 성능 분석
  
- 요구 역량
  - 오픈 소스 기반의 데이터센터 SW 프레임워크 구축 및 평가 경험 (e.g. In-memory DB, Storage DB, Multi-tenancy, Spark, Hadoop, KVM 관련)
  - 컴퓨터 구조, 운영체제, 메모리 시스템에 대한 깊이 있는 이해
  - 시뮬레이터 기반 성능/파워 분석 경험
  - 프로그래밍/스크립팅 경험

[https://recruit.skhynix.com/servlet/reap\\_rcrtSearchForm.view](https://recruit.skhynix.com/servlet/reap_rcrtSearchForm.view)

## ■ Jobs

- Database Engineer
- Cloud Platform Engineer
- Object Storage Engineer

### [NAVER Cloud] Object Storage 서비스 개발 D-71

#### [역할]

- 네이버 클라우드 플랫폼 Object Storage 서비스 개발 및 운영

#### [필요역량]

- 관련 경력 3년 이상
- 분산환경에서 개발 & 디버깅
- 분산시스템 성능 테스트
- 운영체제 및 Network에 대한 이해
- Java, C/C++ 등 중급 이상
- Test Code 작성, Test 자동화
- 클라우드 서비스에 대한 기본적인 이해 및 활용 경험
- RDBMS, NoSQL 등 데이터베이스 활용 경험 및 성능 튜닝 경험
- 웹 서비스의 일반적인 아키텍처에 대한 이해 및 RESTful API 설계 역량

#### [지원자격/우대사항]

- 리눅스 커널, 파일시스템 관련 개발 경험이 있으신 분
- 스토리지 관련 오픈 소스 운영 및 비교 경험 / 연구 경험 보유하신 분
- Distributed Lock, Consistency, Replication 등 분산 시스템이나 파일시스템 관련 실무 또는 연구 경험 보유하신 분
- 대규모 요청에 대한 아키텍처 설계 및 성능 튜닝 경험을 보유하신 분
- 리눅스 환경에서 개발 및 운영 경험을 보유하신 분
- Zookeeper, Redis, Kafka 등 분산 플랫폼 사용 경험을 보유하신 분

<https://career.navercorp.com/naver/job/detail/1000122/?annold=20005410&classId=&jobId=>



## ■ Jobs

- Database Engineer
- Cloud Platform Engineer
- DevOps Software Engineer

## ■ '21 Open Recruitment

- <https://www.careers-2021-liner.com>

## ■ '22 Open Recruitment

- <https://careers.linecorp.com/ko/jobs/1165>

### DevOps Software Engineering

LINE의 다양한 서비스들의 문제점들을 인지하고 해결하는 과정에서 안정적인 운영 기반이 되는 Observability Platform 개발을 담당하고 있습니다. 필연적으로 대용량 데이터를 처리해야 하는 모니터링 기능 개발을 위한 다양한 도전과 경험을 할 수 있습니다.



[신입 LINER가 되시면 이런 업무를 담당할 수 있습니다]

- Metric, Log, Tracing 등의 대용량 데이터 전달을 위한 클라이언트 라이브러리 개발
- 대용량 데이터 수집을 위한 API 서버/파이프라인 개발
- 사내 인프라/서비스에 특화된 모니터링 시스템 개발
- 자체 개발되는 대규모 수준의 Time Series DB 개발

[신입 LINER에게 바랍니다]

- 컴퓨터 과학의 기초 지식(자료 구조/알고리즘/데이터베이스/네트워크/운영 체제 등)을 이해하고 있는 분이면 좋습니다.
- Java, Go 등의 프로그래밍 언어를 활용한 서버 개발 경험이나 관심이 많은 분이면 좋습니다.
- 대용량의 데이터를 다루는 것을 즐길 줄 아는 분이면 좋습니다.
- 발생한 이슈들에 대해 분석하고 해결하는 것에 관심이 있는 분이면 좋습니다.

[필수는 아니지만 아래의 경험이 있으시면 좋습니다]

- 대용량 Storage 오픈소스/기술에 관심이 있는 분이면 좋습니다.
- 성능 개선을 위한 어플리케이션/시스템 튜닝 경험이나 관심이 많은 분이면 좋습니다.

<https://www.careers-2021-liner.com>

1. Study Introduction

2. Why should we understand internal structure?

3. What is a key-value store?

4. Why open-source?

5. Job market

***6. Lab research subject***

7. Environment setup

8. Homework

- **Lab research subject**

- ✓ Summary
- ✓ StarLab
- ✓ NRF
- ✓ SK Hynix

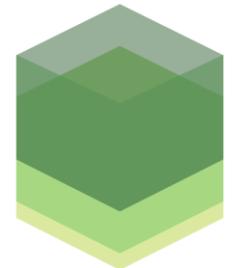
# Lab research subject



Linux Kernel



redis



LEVELDB



RocksDB

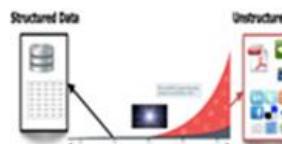


Next Generation SSD

<https://www.aspsys.com/solutions/storage-solutions/flash-storage-nvme/>

# StarLab



Objective	 <b>키-밸류 DB와 차세대 SSD, 오픈 소스에 정통한 빅데이터 전문가 양성</b> 														
Approach	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <b>Key-Value DB Optimization</b> <ul style="list-style-type: none"> <li>• User-level Direct Access</li> <li>• Hot-cold separation</li> <li>• New compaction algorithm</li> </ul> </div> <div style="text-align: center;"> <b>HW-SW Codesign</b> <ul style="list-style-type: none"> <li>• Predictable I/O on ZNS</li> <li>• Host-SW for KVSSD</li> <li>• SW-enabled WAF reduction</li> </ul> </div> <div style="text-align: center;"> <b>Integrating New Domain</b> <ul style="list-style-type: none"> <li>• Learned index (ML)</li> <li>• BlueStore (Ceph)</li> <li>• HW-assisted Transaction</li> </ul> </div> </div>														
Necessity	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <b>Software Semantic Gap</b>  <ul style="list-style-type: none"> <li>• Compaction/Lookup issue</li> <li>• Duplication/Unawareness</li> </ul> </div> <div style="text-align: center;"> <b>HW features Oblivious</b>  <ul style="list-style-type: none"> <li>• Storage-aware DB mgmt.</li> <li>• Make use of ISP</li> </ul> </div> <div style="text-align: center;"> <b>Lack of System Level View</b>  <ul style="list-style-type: none"> <li>• Unpredictable latency spike</li> <li>• Storage level vs. Client level</li> </ul> </div> </div>														
Background	<b>Growth of unstructured data</b> 	<b>Diverse Key-Value DB</b> 	<b>Next Generation SSDs</b> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Zone IF</th> <th>KV IF</th> <th>FTL (Flash Translation Layer)</th> </tr> </thead> <tbody> <tr> <td>NAND (channel, wip, ...)</td> <td>NAND (channel, wip, ...)</td> <td>PCM (NVM)</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td>(ZNS SSD)</td> <td>(KVSSD)</td> <td>(Optane SSD)</td> </tr> </tbody> </table>	Zone IF	KV IF	FTL (Flash Translation Layer)	NAND (channel, wip, ...)	NAND (channel, wip, ...)	PCM (NVM)				(ZNS SSD)	(KVSSD)	(Optane SSD)
Zone IF	KV IF	FTL (Flash Translation Layer)													
NAND (channel, wip, ...)	NAND (channel, wip, ...)	PCM (NVM)													
															
(ZNS SSD)	(KVSSD)	(Optane SSD)													

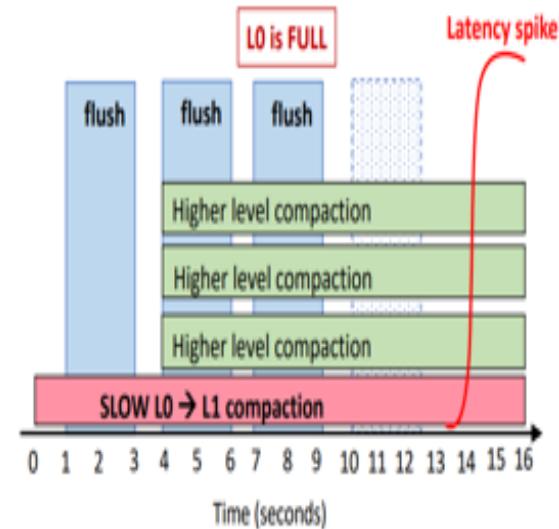
[http://www.swstarlab.kr/page/page02\\_02.php?c=lab37](http://www.swstarlab.kr/page/page02_02.php?c=lab37)

과제명(국문)	응답 시간 예측성을 보장하는 새로운 키-밸류 스토어 개발
과제명(영문)	Development of a Novel Key-Value Store that Supports Predictable Latency

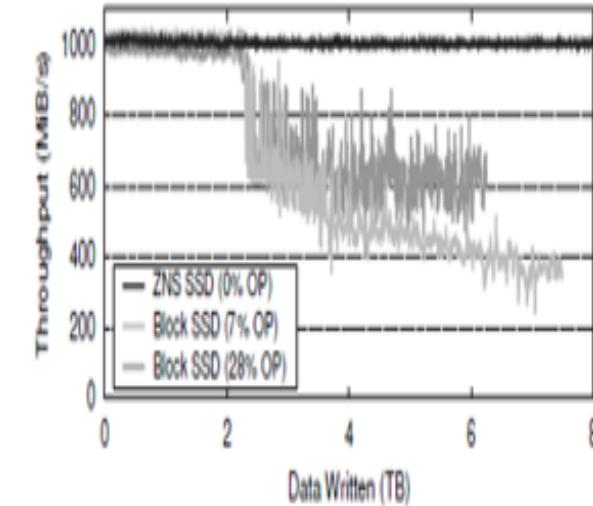
Tail Latency



(a) Latency spike (from L사 talk)



(b) Latency spike (from Silk paper)



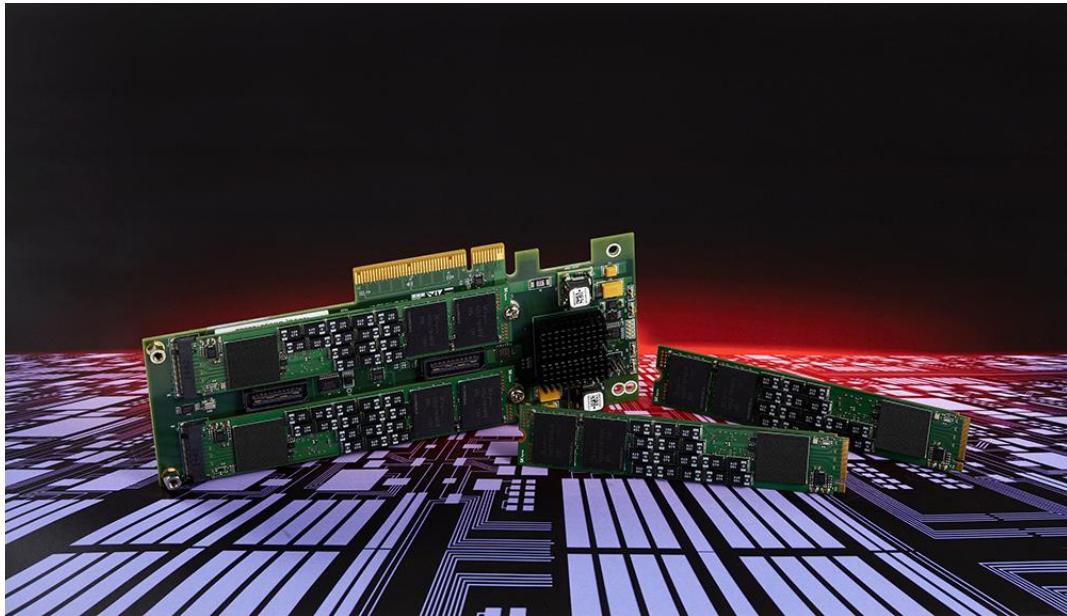
(c) SSD GC effect (from ZNS paper)

# SK Hynix



## SK hynix Demonstrates Industry's First ZNS-based SSD Solution for Data Centers

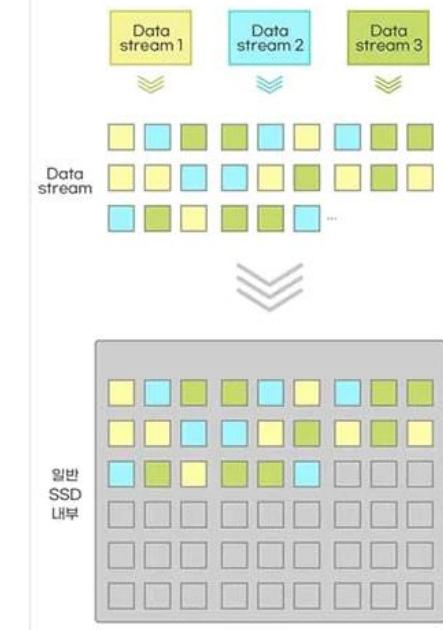
March 25, 2019



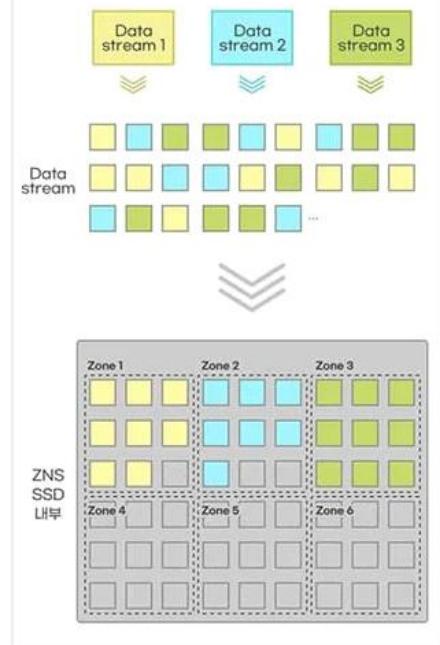
<https://news.skhynix.com/sk-hynix-demonstrates-industrys-first-zns-based-ssd-solution-for-data-centers-2/>

## 일반 SSD와 ZNS SSD의 데이터 저장 방식 비교

### I 일반 SSD



### I ZNS SSD



1. Study Introduction

2. Why should we understand internal structure?

3. What is a key-value store?

4. Why open-source?

5. Job market

6. Lab research subject

**7. *Environment setup***

8. Homework

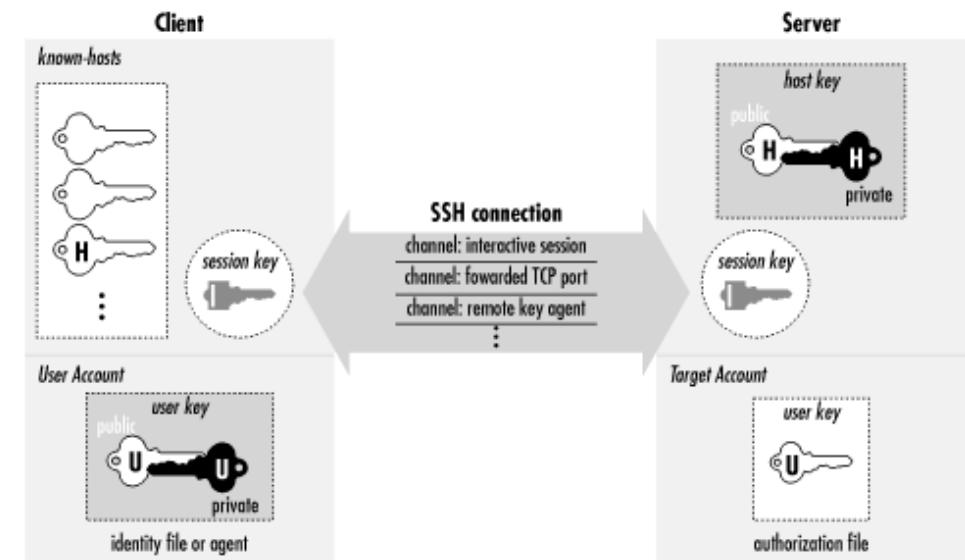
- **Environment setup**

- ✓ SSH
- ✓ Linux
  - SSH Server
- ✓ VS code
  - SSH Client App

# SSH

---

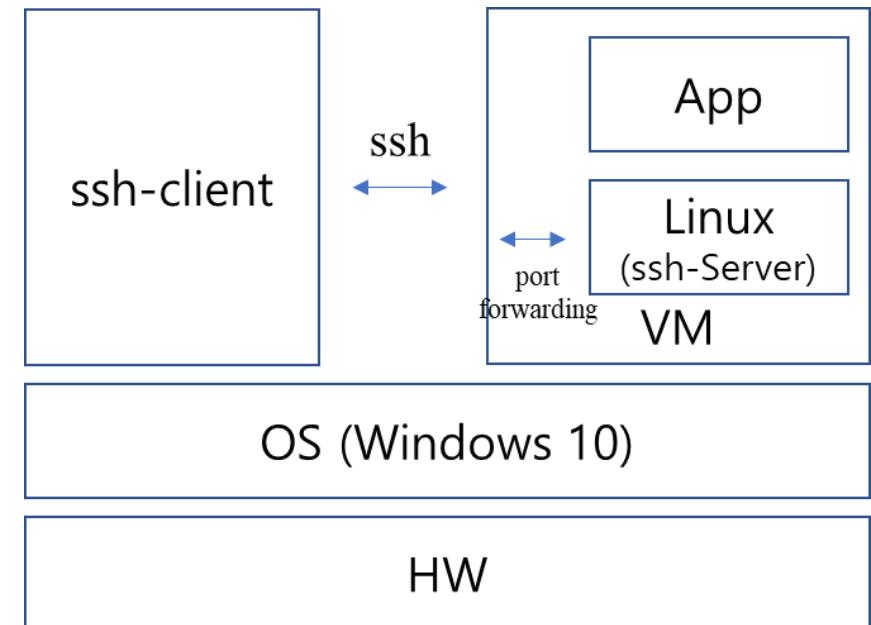
- SSH (Secure Shell)
  - Network protocol
  - Client-Server architecture
  
- Why?
  - Linux env. for LevelDB
  - It's inconvenient to use Linux alone
  
- Linux Server <-> Windows Client



[https://docstore.mik.ua/orelly/networking\\_2ndEd/ssh/ch03\\_03.htm](https://docstore.mik.ua/orelly/networking_2ndEd/ssh/ch03_03.htm)

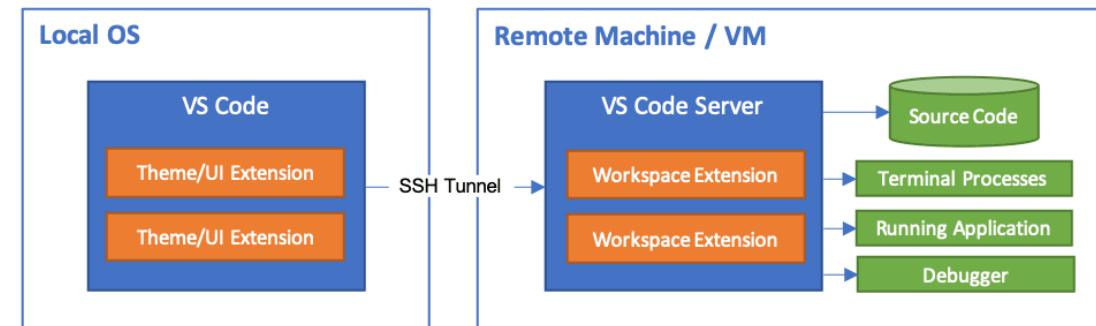
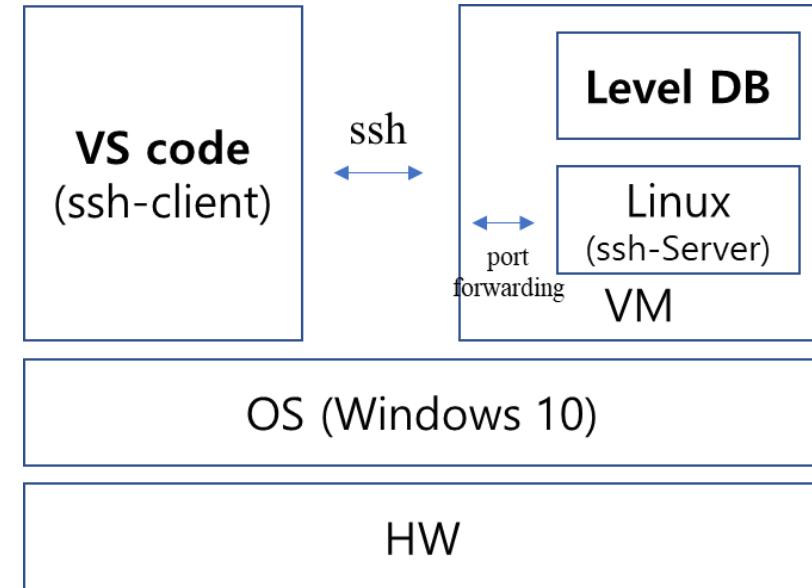
# Linux

- Ubuntu
  - 20.04 <https://releases.ubuntu.com/20.04/>
- Machine
  - Remote Machine
  - VM (Virtual Box) : <https://www.virtualbox.org/wiki/Downloads>
  - WSL (not recommended)
- SSH Server setup
  - <https://ubuntu.com/server/docs/service-openssh>
- Virtual Box port forwarding
  - <https://medium.com/platform-engineer/port-forwarding-for-ssh-http-on-virtualbox-459277a888be>



# VS code

- SSH Client App (+ Text Editor/IDE)
  - Putty + Vim
  - Xshell + Vim
  - VS code (+ Remote Development Extension)
  
- VS code
  - Download (<https://code.visualstudio.com/>)
  - Remote Development Extension-SSH
    - <https://code.visualstudio.com/docs/remote/ssh>



<https://code.visualstudio.com/docs/remote/ssh>

1. Study Introduction

2. Why should we understand internal structure?

3. What is a key-value store?

4. Why open-source?

5. Job market

6. Lab research subject

7. Environment

**8. Homework**

## ■ Homework

- ✓ Jongmoo Choi, 『Key-Value DB for Unstructured data』, 2021
  - <https://mooc.dankook.ac.kr/courses/61d537a3b6b71841651153b3>
- ✓ Environment setup

## ■ Photo time

# Thank you