RocksDB Festival

What is the RocksDB

Supported by IITP, StarLab.

July 5, 2021

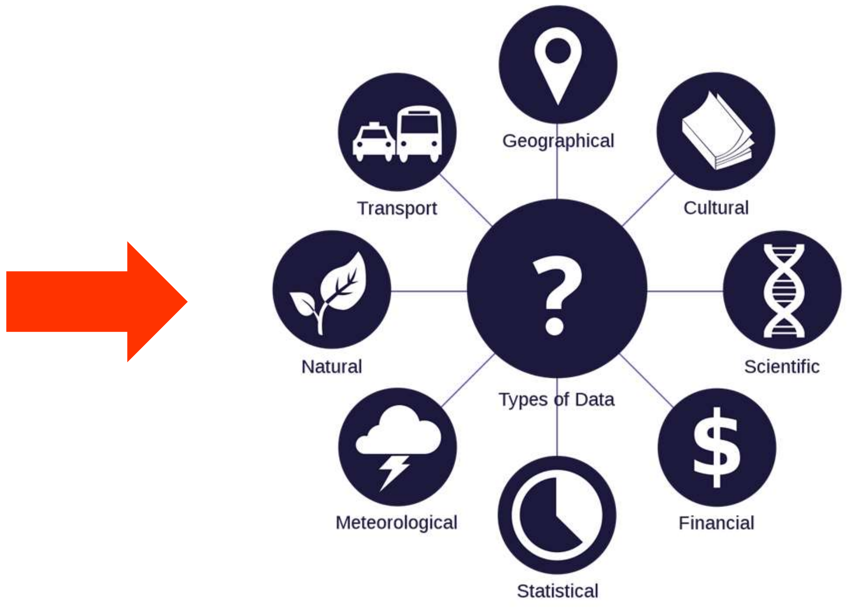
Hojin Shin, Jongmoo Choi
choijm@dankook.ac.kr
http://embedded.dankook.ac.kr/~choijm

RocksDB Festival: Contents

- Contents
  - ✓ What is data
  - ✓ How to manage data
  - ✓ This is Key-Value Store
  - ✓ The basics of key value: LevelDB
  - ✓ Our Goal: RocksDB

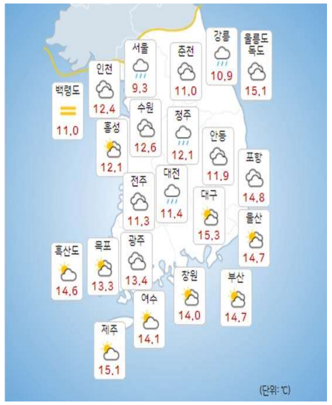
RocksDB Festival: What is the Data

- Data
  - ✓ 1) Units of information, often numeric, that are collected through observation.
  - ✓ 2) Fact on which a theory is based
  - ✓ 3) Data in the form of letters, numbers, sounds, pictures that a computer can process



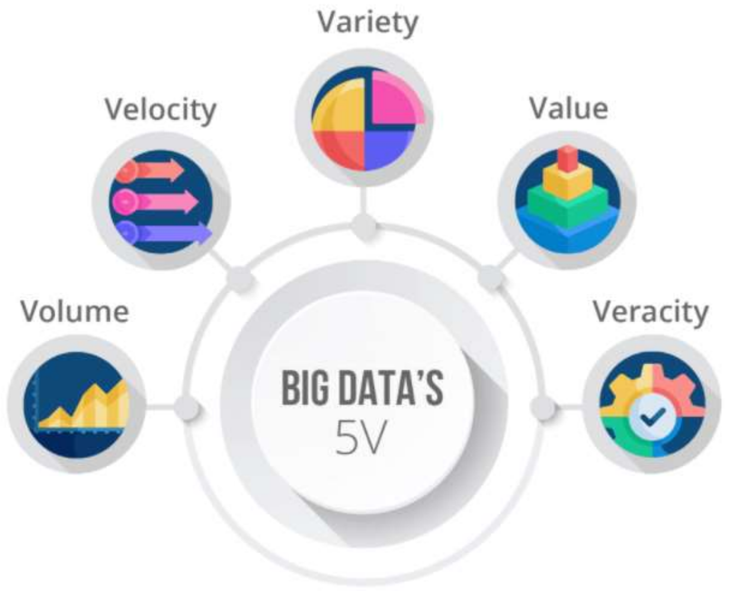
RocksDB Festival: What is the Data

- Information
  - ✓ Information is obtained by processing data
  - ✓ A form in which data is processed according to its meaning and purpose for specific decision-making



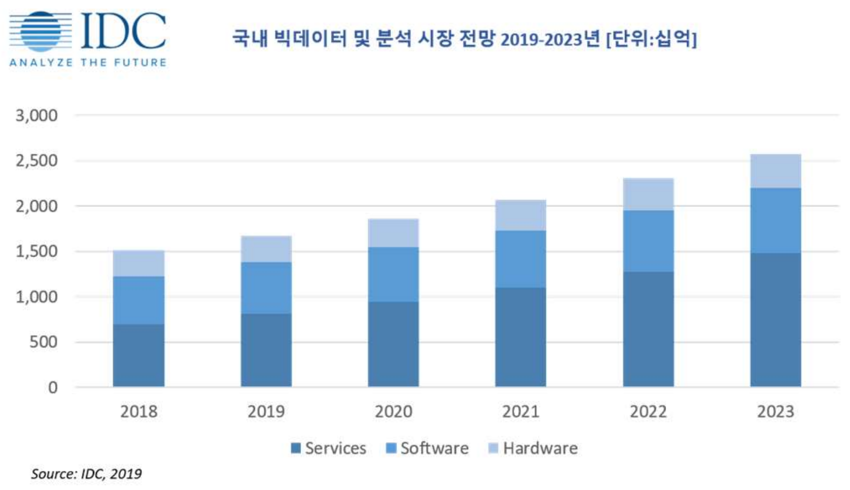
RocksDB Festival: What is the Data

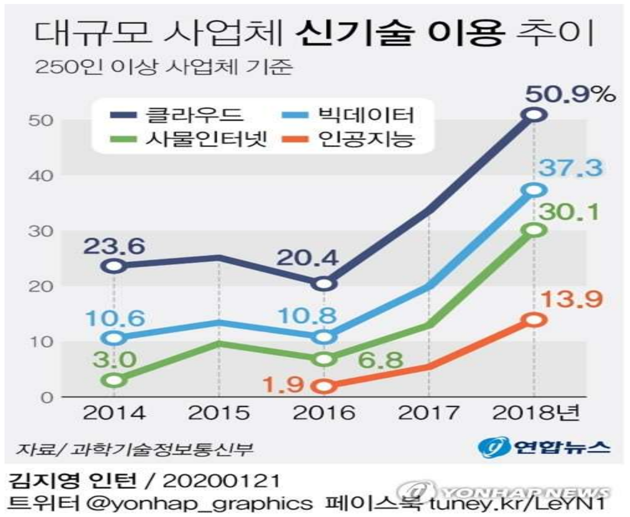
- Big Data
  - ✓ A large amount of structured data that exceed existing DB management tools
  - ✓ Set of unstructured data that is not in the form of data
  - ✓ Features = 5V

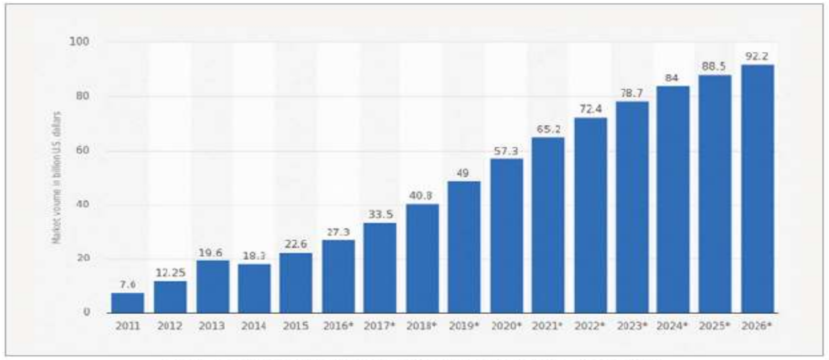


RocksDB Festival: What is the Data

- Big Data Prospects







2011~2026년간 빅데이터 시장 규모 전망 (단위: 10억 달러)

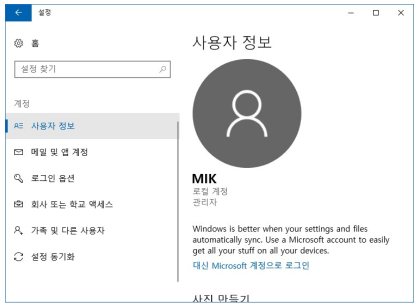
〈출처: statista〉

RocksDB Festival : What is the Data

- Kind of Data
  - ✓ Structured Data
    - Data organized and processed into a form suitable for immediate statistical analysis
    - Data stored in fixed fields
  - ✓ Unstructured Data
    - One piece of data, not a set of data, is objectified as collected data
    - Difficult to understand the meaning of a value because there is no set rule
  - ✓ Semi-structured Data
    - File type, metadata (schema of structured data inside data)

RocksDB Festival: How to manage Data

- Various Data



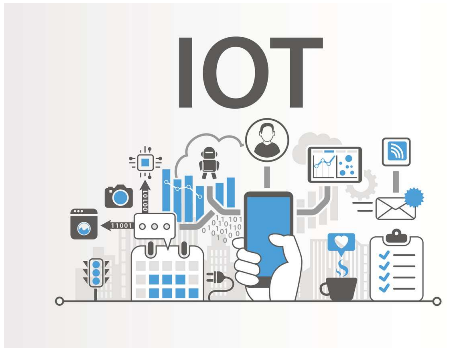
사용자 정보

MIK
로컬 계정
관리자

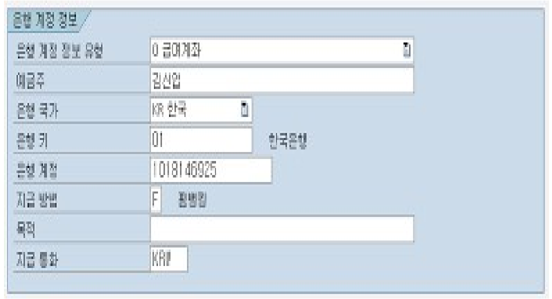
Windows is better when your settings and files automatically sync. Use a Microsoft account to easily get all your stuff on all your devices.
대신 Microsoft 계정으로 로그인

사진 만들기

{JSON}

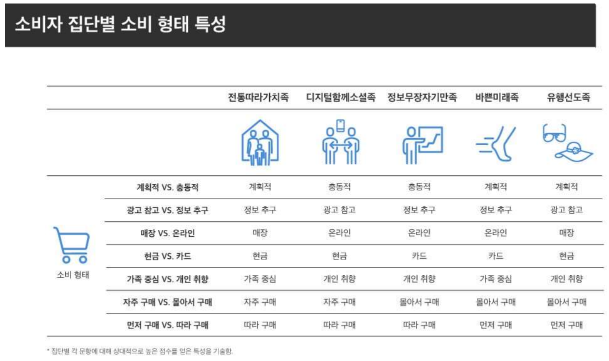


XML



은행 계정 정보

은행 계정 정보 유형	0 급여계좌
예금주	김신압
은행 국가	KR 한국
은행 키	01 한국은행
은행 계정	1018146925
지급 방법	F 펌뱅킹
목적	
지급 통화	KRW



소비자 집단별 소비 형태 특성

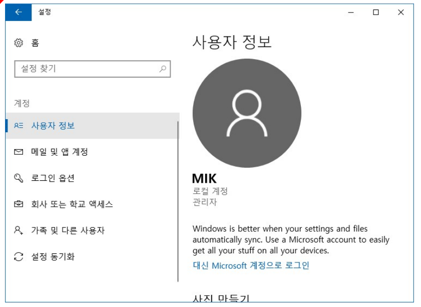
		전통따라가치족	디지털함께소셜족	정보무장자기만족	바쁜미래족	유행선도족
소비 형태	계획적 VS. 충동적	계획적	충동적	충동적	계획적	계획적
	광고 참고 VS. 정보 추구	정보 추구	광고 참고	정보 추구	정보 추구	광고 참고
	매장 VS. 온라인	매장	온라인	온라인	온라인	매장
	현금 VS. 카드	현금	현금	카드	카드	현금
	가족 중심 VS. 개인 취향	가족 중심	개인 취향	개인 취향	가족 중심	개인 취향
	자주 구매 VS. 몰아서 구매	자주 구매	자주 구매	몰아서 구매	몰아서 구매	몰아서 구매
	먼저 구매 VS. 따라 구매	따라 구매	따라 구매	따라 구매	먼저 구매	먼저 구매

* 집단별 각 문항에 대해 상대적으로 높은 점수를 얻은 특성을 기술함.

TALK
blog

RocksDB Festival: How to manage Data

- Various Data



설정

홈

설정 찾기

계정

- 사용자 정보
- 메일 및 앱 계정
- 로그인 옵션
- 회사 또는 학교 액세스
- 가족 및 다른 사용자
- 설정 동기화

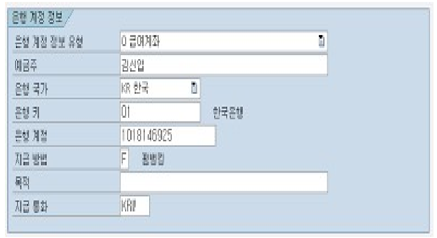
사용자 정보

MIK
로컬 계정
관리자

Windows is better when your settings and files automatically sync. Use a Microsoft account to easily get all your stuff on all your devices.

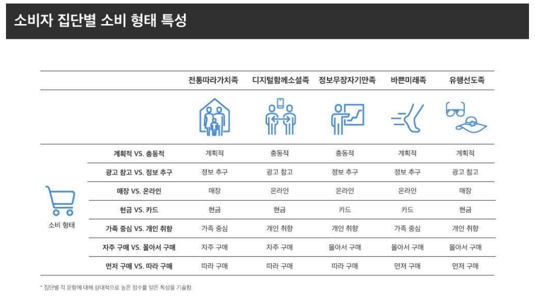
대신 Microsoft 계정으로 로그인

사진 만들기



은행 계정 정보

은행 계정 정보 유형	0 급여계좌	
예금주	김신입	
은행 국가	KR 한국	
은행 키	01	한국은행
은행 계정	1018146925	
지급 방법	F	펌뱅킹
목적		
지급 통화	KRW	



소비자 집단별 소비 형태 특성

		전통따라가치족	디지털함께소셜족	정보무장자기만족	바쁜미래족	유행선도족
소비 형태	계획적 VS. 충동적	계획적	충동적	충동적	계획적	계획적
	광고 참고 VS. 정보 추구	정보 추구	광고 참고	정보 추구	정보 추구	광고 참고
	매장 VS. 온라인	매장	온라인	온라인	온라인	매장
	현금 VS. 카드	현금	현금	카드	카드	현금
	가족 중심 VS. 개인 취향	가족 중심	개인 취향	개인 취향	가족 중심	개인 취향
	자주 구매 VS. 몰아서 구매	자주 구매	자주 구매	몰아서 구매	몰아서 구매	몰아서 구매
	먼저 구매 VS. 따라 구매	따라 구매	따라 구매	따라 구매	먼저 구매	먼저 구매

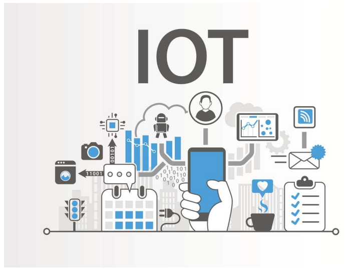
* 집단별 각 문항에 대해 상대적으로 높은 점수를 얻은 특성을 기술함.

Structured Data

{JSON}

XML

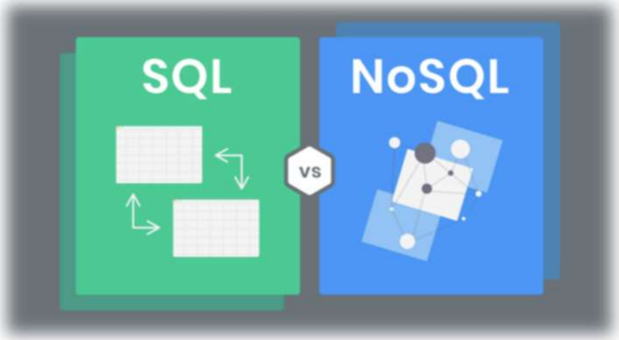
Semi-structured Data



Unstructured Data

RocksDB Festival: How to manage Data

- Data management
  - ✓ SQL (Structured Query Language)
    - Interact with a particular type of database
    - Can store, modify, delete and retrieve data from RDBMS
    - Features
      - Strict schema
      - Relation
  - ✓ NoSQL (Not only SQL)
    - Adjust the stored data at any time and add new "fields“
    - Key-value, document, wide-column, graph
    - Features
      - No schema
      - No Relation



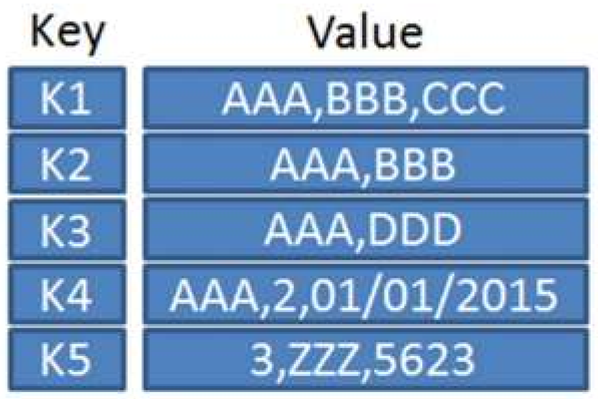
RocksDB Festival: How to manage Data

- Data management cont'
  - ✓ SQL
    - RDBMS(Relational DataBase Management System)
    - Easy to perform transactions by minimizing data redundancy through normalization
    - Data integrity – Accuracy, consistence
    - MSSQL, MySQL, Oracle
  - ✓ NoSQL
    - Does not define relationships between data
    - Store large amounts of data
    - LevelDB, RocksDB, Cassandra, MongoDB, Memcached

SQL **NoSQL**

RocksDB Festival: This is Key-Value Store

- Key-Value Store
  - ✓ A type of non-relational database that uses key-value methods to store data
  - ✓ Key is a unique identifier and cannot overlapping value
  - ✓ Value can be anything – integer, string, JSON, image …
  - ✓ Hash function is used to process the key

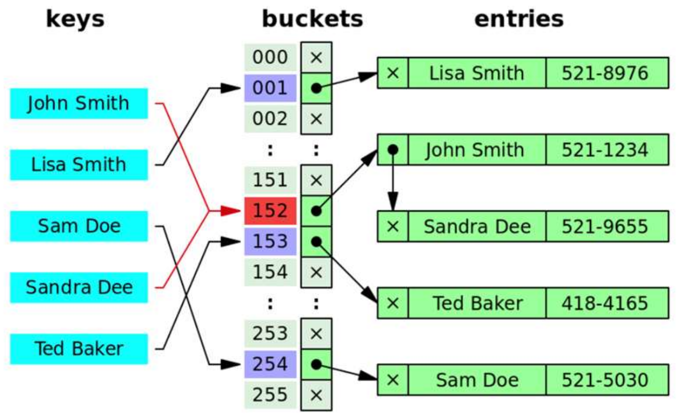


Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Key-Value Overview

RocksDB Festival: This is Key-Value Store

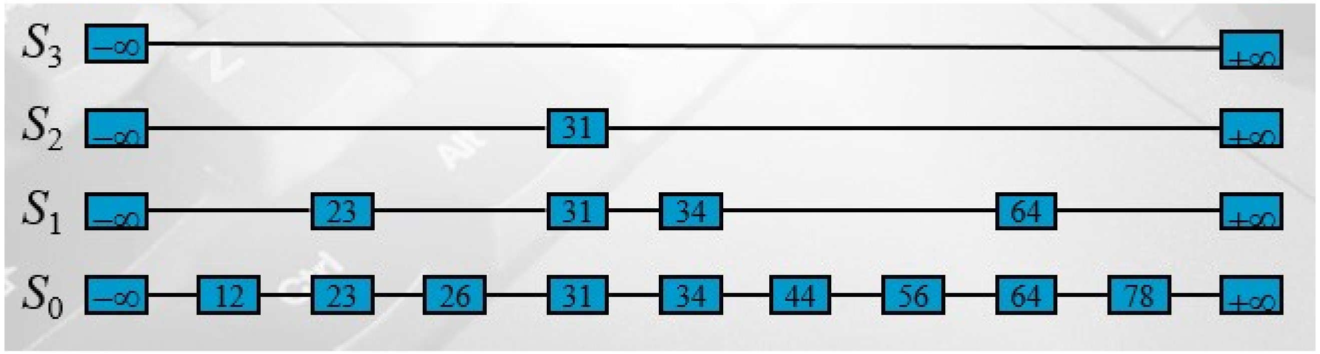
- Hash-Table
  - ✓ Data structure that stores value in key
  - ✓ When searching for data about a key, if you execute hash function only once ➔ store and delete data is fast
  - ✓ Mapping ➔ Called Hashing
  - ✓ Data access: insert, delete, retrieve O(1)
  - ✓ Problem
    - Hash collision ➔ Can solve using “Chaining”



Hash Table Overview

RocksDB Festival: This is Key-Value Store

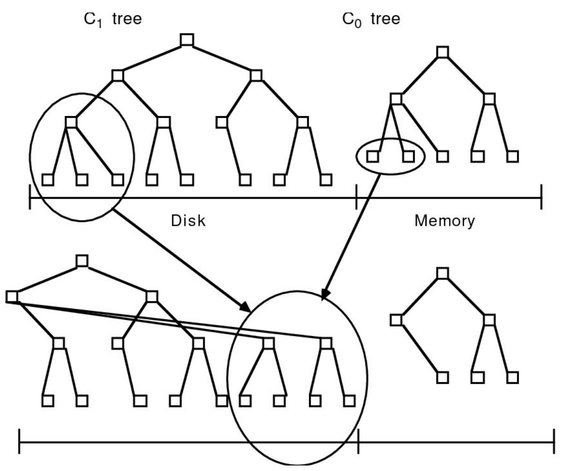
- Skiplist
  - ✓ A data structure that enables fast search, insert, and delete with an algorithm applied to a sorted linked list
  - ✓ Useful in multithreaded system architectures
  - ✓ Complexity : O(logn)



Skiplist Overview

RocksDB Festival: This is Key-Value Store

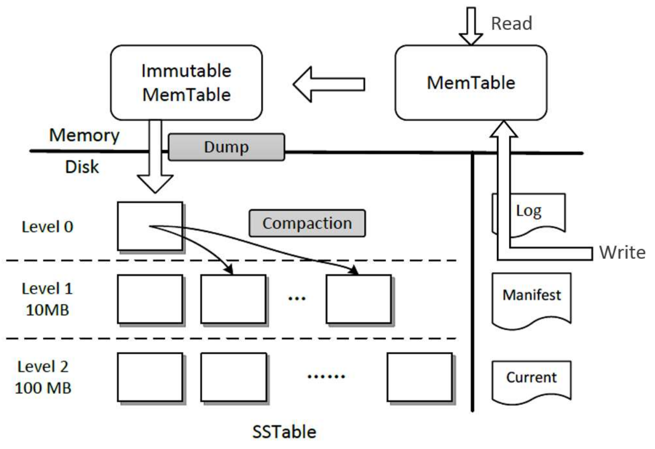
- LSM(Log Structured Merge)-Tree
  - ✓ Patrick O'Neil, The Log-Structured Merge Tree, 1996
  - ✓ Has 0~L levels, L0 is in memory others are in disk(storage)
  - ✓ Buffer located in L0 stores data and when buffer full, flushed to lower level one by one
  - ✓ Not In-place-update, using Append
  - ✓ Write
    - Memory buffer construct skiplist, maintain input data's order
  - ✓ Read
    - When reading, all files are checked
    - Memory ➔ Immutable memory ➔ Disk



LSM Tree Overview

RocksDB Festival: LevelDB

- LevelDB
  - ✓ Google's opensource project
  - ✓ Developed in the programming language C++
  - ✓ Data is stored after sorting by key
  - ✓ Operation : Put(K, V), Get(K), Delete(K)
  - ✓ Multiple operations can be created and processed in one batch
  - ✓ Limitation
    - Single processing: only one process can access DB
    - Not support SQL query



LevelDB Overview

RocksDB Festival: RocksDB

- Overview
  - ✓ RocksDB is a storage engine for server workloads (Developed by Facebook)
  - ✓ Data stored by key and value
  - ✓ Flexibility: Support various production env. (Memory, Flash, HDD)

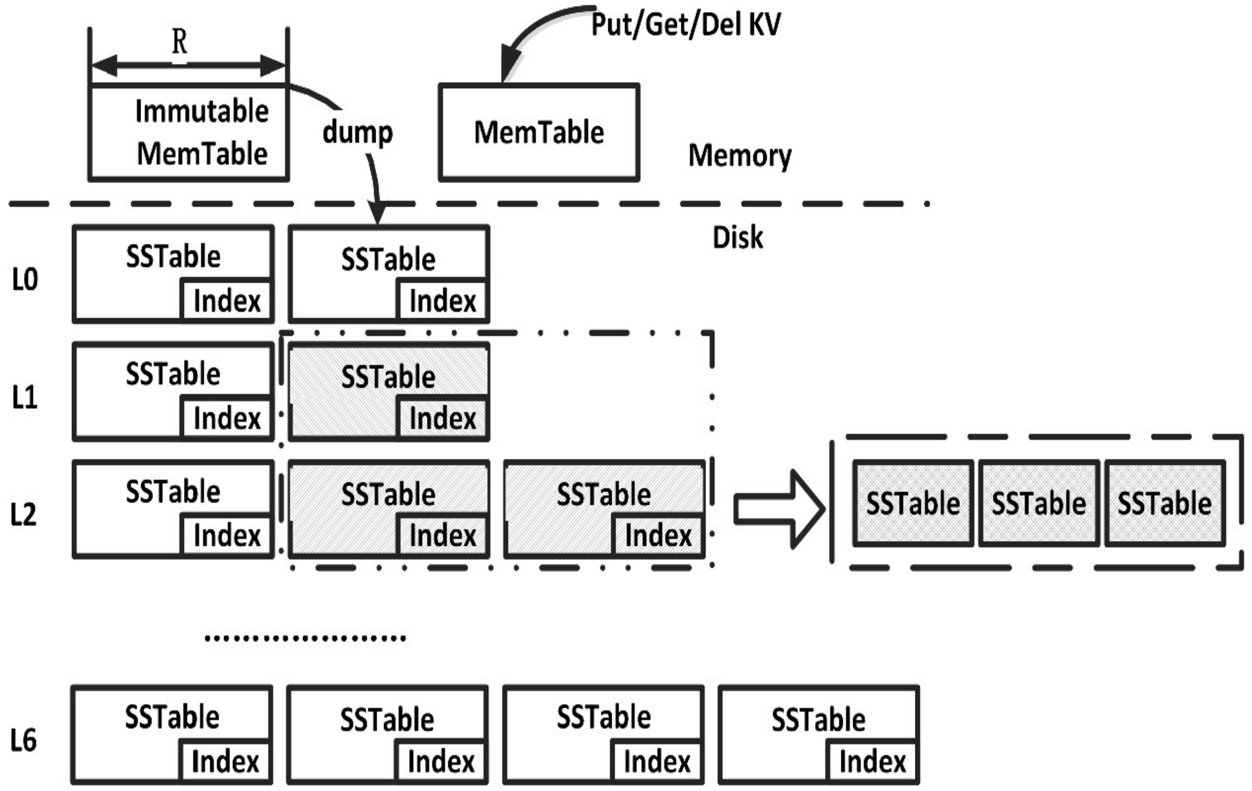
  - ✓ Operation: Get(K), NewIterator(), Put(K,V), Delete(K) …
  - ✓ Memtable: In-memory data structure
  - ✓ Log: Sequential write into storage
  - ✓ SSTable: Sorted data (L1 >) in storage
  - ✓ WAL(Write-Ahead-Log): Before flush memtable, do flush log

RocksDB Festival: RocksDB

- Overview cont’
  - ✓ Column Family
    - Supports partitioning database into multiple column families
  - ✓ Update
    - Put API inserts a single key-value into the database
  - ✓ Get, Iterator
    - Get a single key-value from the database using the Get API
    - Iterator API allows applications to perform range scans on the database
  - ✓ Compaction
    - Removes multiple copies of the same key that may occur when overwriting an existing key
    - Write throughput depends on compactions speed

RocksDB Festival: RocksDB

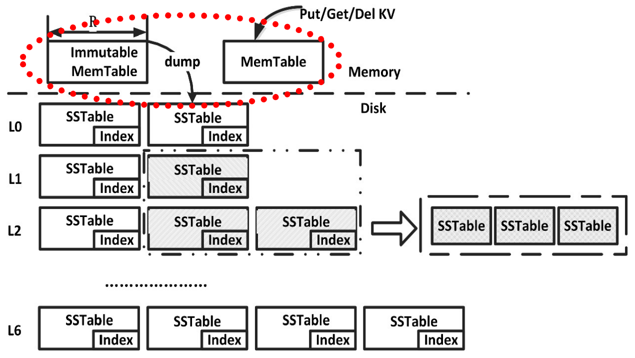
- RocksDB Architecture



RocksDB Architecture

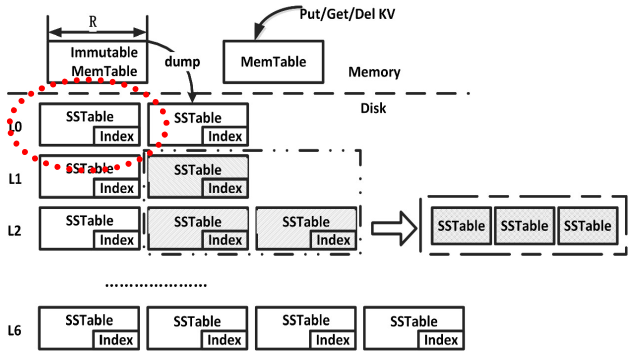
RocksDB Festival: RocksDB

- Main Terminology for RocksDB: MemTable
  - ✓ Memtable is an in-memory data-structure that holds data before being flushed to the SST file
  - ✓ When memtable full, it becomes immutable memtable
  - ✓ The background thread flushes the contents of the memtable to the SST file
  - ✓ Memtable has a skiplist structure
  - ✓ Insert and check first when performing Put() and Get()
  - ✓ Skiplist, HashSkiplist, HashLinklist, Vector



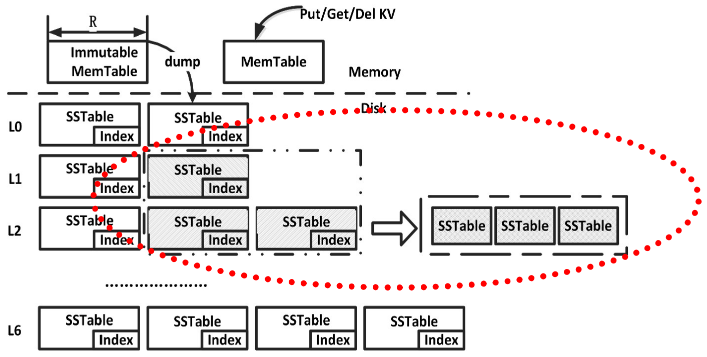
RocksDB Festival: RocksDB

- Main Terminology for RocksDB: SSTable
  - ✓ SSTable (Sorted String Table)
  - ✓ SSTable that exist in levels other than L0 have an ordered state
  - ✓ Inside, there are data blocks, index blocks, bloom filter blocks, etc
  - ✓ Index block
    - The data block containing the range containing the lookup key is used for lookup
    - Has a binary search data structure
  - ✓ Bloom filter
    - Given an arbitrary key, this bit array may be used to determine if the key may exist or does not exist in the key set



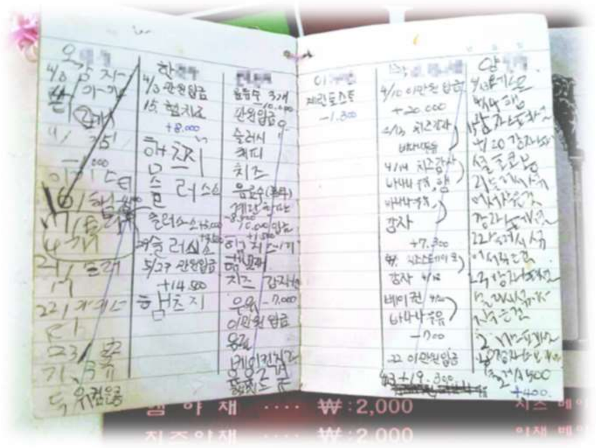
RocksDB Festival: RocksDB

- Main Terminology for RocksDB: Compaction
  - ✓ Compaction is for update
  - ✓ Prevents overlapping key-value pair from accumulating for existing key
  - ✓ Data in L0 is the hot data, and the lower level is the cold data
  - ✓ Compaction uses a background thread
  - ✓ Various Type
    - Leveled compaction, Tiered(Universal) compaction, FIFO compaction



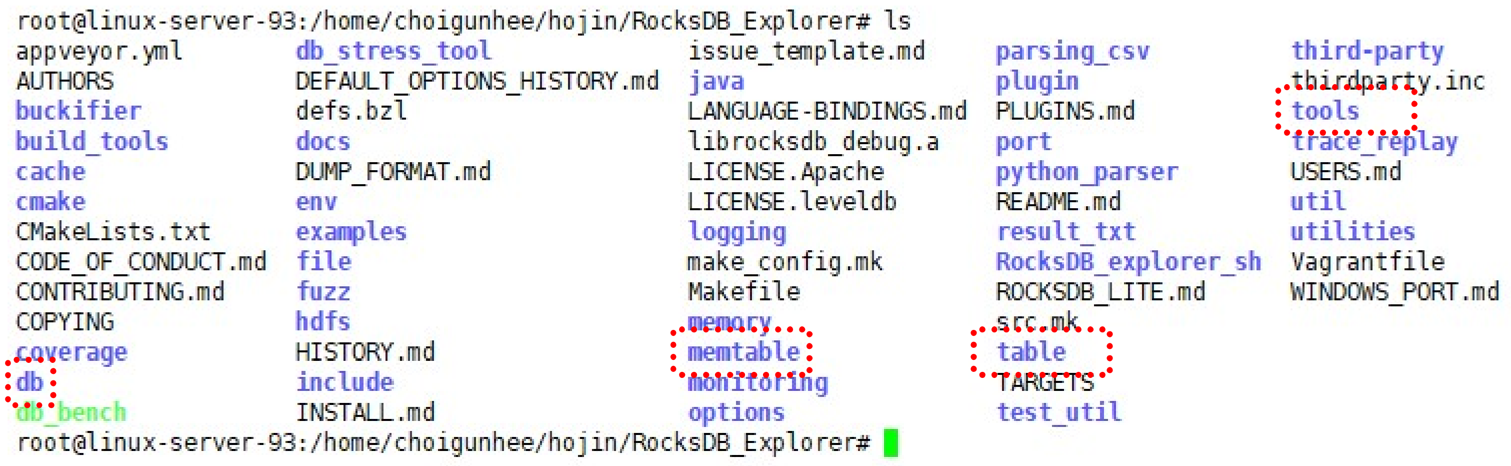
RocksDB Festival: RocksDB

- Main Terminology for RocksDB: WAL(Write-Ahead-Log)
  - ✓ Record the WAL of memtable and disk for every update
  - ✓ Used for recovery in the event of an unexpected shutdown or error
  - ✓ Memtable securely as storage(disk) when flushed, the WAL is deleted.

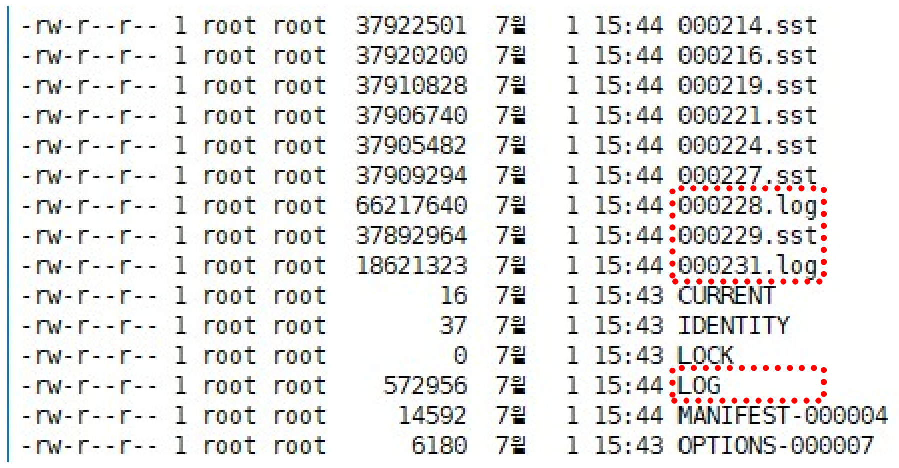


RocksDB Festival: RocksDB

■ RocksDB preview



```
root@linux-server-93:/home/choigunhee/hojin/RocksDB_Explorer# ls
appveyor.yml        db_stress_tool              issue_template.md     parsing_csv          third-party
AUTHORS             DEFAULT_OPTIONS_HISTORY.md  java                  plugin               thirdparty.inc
buckifier           defs.bzl                    LANGUAGE-BINDINGS.md  PLUGINS.md           tools
build_tools         docs                        librocksdb_debug.a    port                 trace_replay
cache               DUMP_FORMAT.md              LICENSE.Apache        python_parser        USERS.md
cmake               env                         LICENSE.leveldb       README.md            util
CMakeLists.txt      examples                    logging               result_txt           utilities
CODE_OF_CONDUCT.md  file                        make_config.mk        RocksDB_explorer_sh  Vagrantfile
CONTRIBUTING.md     fuzz                        Makefile              ROCKSDB_LITE.md      WINDOWS_PORT.md
COPYING             hdfs                        memory                src.mk
coverage            HISTORY.md                  memtable              table
db                  include                     monitoring            TARGETS
db_bench            INSTALL.md                  options               test_util
root@linux-server-93:/home/choigunhee/hojin/RocksDB_Explorer# 
```



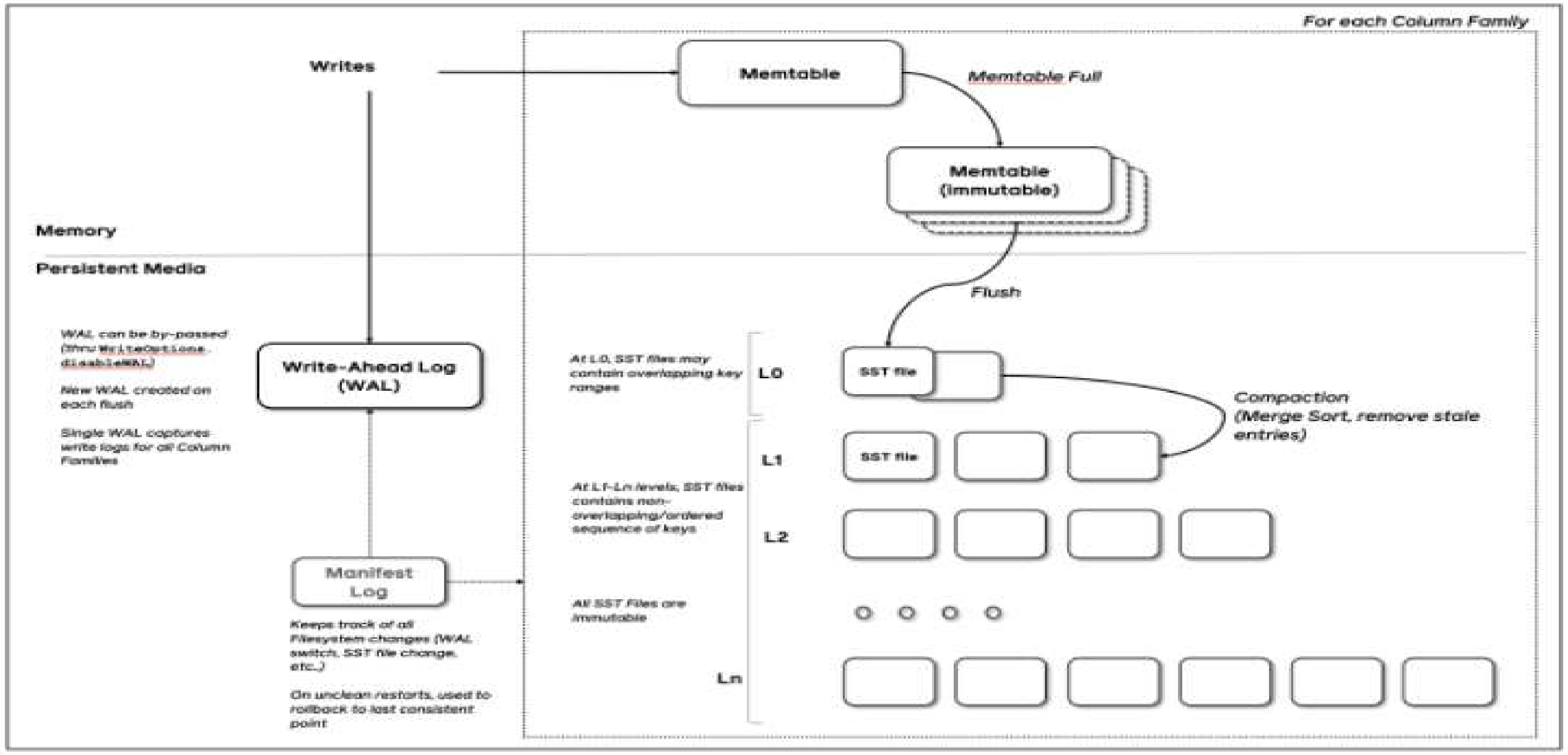
```
-rw-r--r-- 1 root root  37922501  7월   1 15:44 000214.sst
-rw-r--r-- 1 root root  37920200  7월   1 15:44 000216.sst
-rw-r--r-- 1 root root  37910828  7월   1 15:44 000219.sst
-rw-r--r-- 1 root root  37906740  7월   1 15:44 000221.sst
-rw-r--r-- 1 root root  37905482  7월   1 15:44 000224.sst
-rw-r--r-- 1 root root  37909294  7월   1 15:44 000227.sst
-rw-r--r-- 1 root root  66217640  7월   1 15:44 000228.log
-rw-r--r-- 1 root root  37892964  7월   1 15:44 000229.sst
-rw-r--r-- 1 root root  18621323  7월   1 15:44 000231.log
-rw-r--r-- 1 root root        16  7월   1 15:43 CURRENT
-rw-r--r-- 1 root root        37  7월   1 15:43 IDENTITY
-rw-r--r-- 1 root root         0  7월   1 15:43 LOCK
-rw-r--r-- 1 root root    572956  7월   1 15:44 LOG
-rw-r--r-- 1 root root     14592  7월   1 15:44 MANIFEST-000004
-rw-r--r-- 1 root root      6180  7월   1 15:43 OPTIONS-000007
```

SW스타랩

Dankook University Embedded System

RocksDB Festival: RocksDB

- Preview next week
  - ✓ RocksDB Architecture
  - ✓ RocksDB Operation (Compaction, WAL, Read, Write …)
  - ✓ RocksDB Benchmark


For each Column Family
Writes
Memtable
Memtable Full
Memtable (immutable)
Memory
Persistent Media
Flush
WAL can be by-passed (thru WriteOptions.disableWAL)
New WAL created on each flush
Single WAL captures write logs for all Column Families
Write-Ahead Log (WAL)
Manifest Log
Keeps track of all Filesystem changes (WAL switch, SST file change, etc.)
On unclean restarts, used to rollback to last consistent point
At L0, SST files may contain overlapping key ranges
At L1-Ln levels, SST files contains non-overlapping/ordered sequence of keys
All SST Files are immutable
L0
L1
L2
Ln
SST file
SST file
Compaction (Merge Sort, remove stale entries)

Discussion