LESS : Logging Exploiting SnapShot

연세대학교 컴퓨터과학과 성한승

2019년 4월

SW STAR LAB
Software Technology Advanced Research

과제명: IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발

과제번호: 2017-0-00477

과학기술정보통신부
Ministry of Science and ICT

연세대학교
YONSEI UNIVERSITY

IITP 정보통신기술진흥센터
Institute for Information & communications Technology Promotion

Contents

01. Introduction

02. Background & Motivation

03. LESS

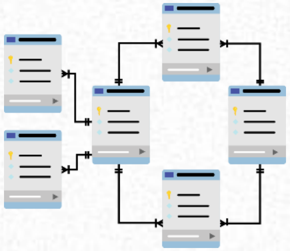
04. Evaluation

05. Conclusion

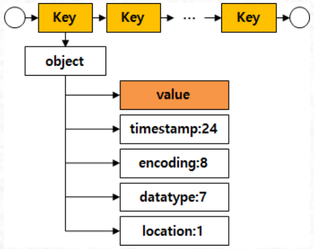
01 Introduction

Introduction

- Changes in Trends
  - The appearance of Key-Value Store
    - ✓ Unstructured data are produced (ex, blog data, sensor data...)
    - ✓ Difficult to process unstructured data with RDBMS
      - ➔ Store data as key-value pairs



Relational Model



Key-Value Model

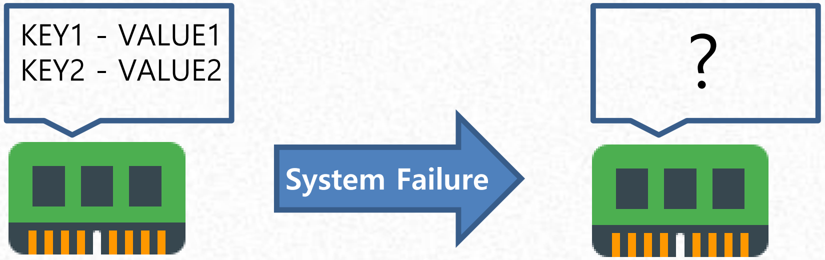
- Device
  - ✓ Demand for rapid processing of large volumes of data
    - ➔ Need high-performance device

Hard Disk Drive & Solid State Drive

Dynamic Random-Access Memory

Introduction

- In-Memory Key-Value Store
  - Use Dynamic Random-Access Memory as its primary storage
    - ✓ High performance and Efficient unstructured data management
    - ✓ Redis, Memcached, RAMCloud, MICA, etc.
    - ✓ Volatile….



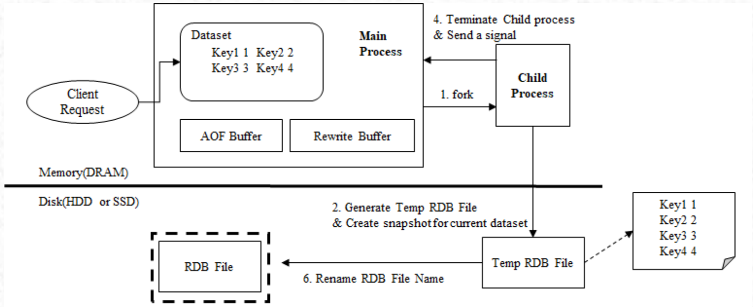
- To ensure data durability
  - ✓ Snapshot : Create snapshot file of current dataset
  - ✓ Append Log File : Write the log of performed command in the log file

02 | Background & Motivation

Background

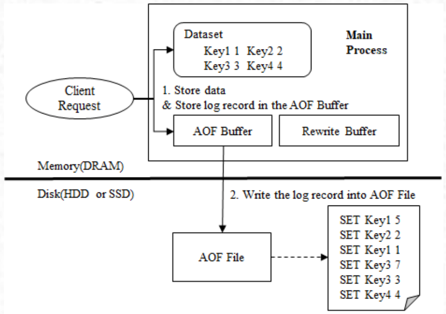
- Redis
  - The most popular in-memory key-value store
  - Store data in key-value pairs in memory
  - Provide various data structures
    - *string, list, set, sorted set, hash …*
  - Single thread-based process
  - Persistence method
    - RDB
    - Append-Only-File(AOF)

- RDB(Snapshot)



  - Generate point-in-time snapshots of a dataset at specific intervals
  - Advantage
    - Small file size
    - Higher performance than AOF
    - Fast recovery speed
  - Disadvantage
    - Can not guarantee data persistency

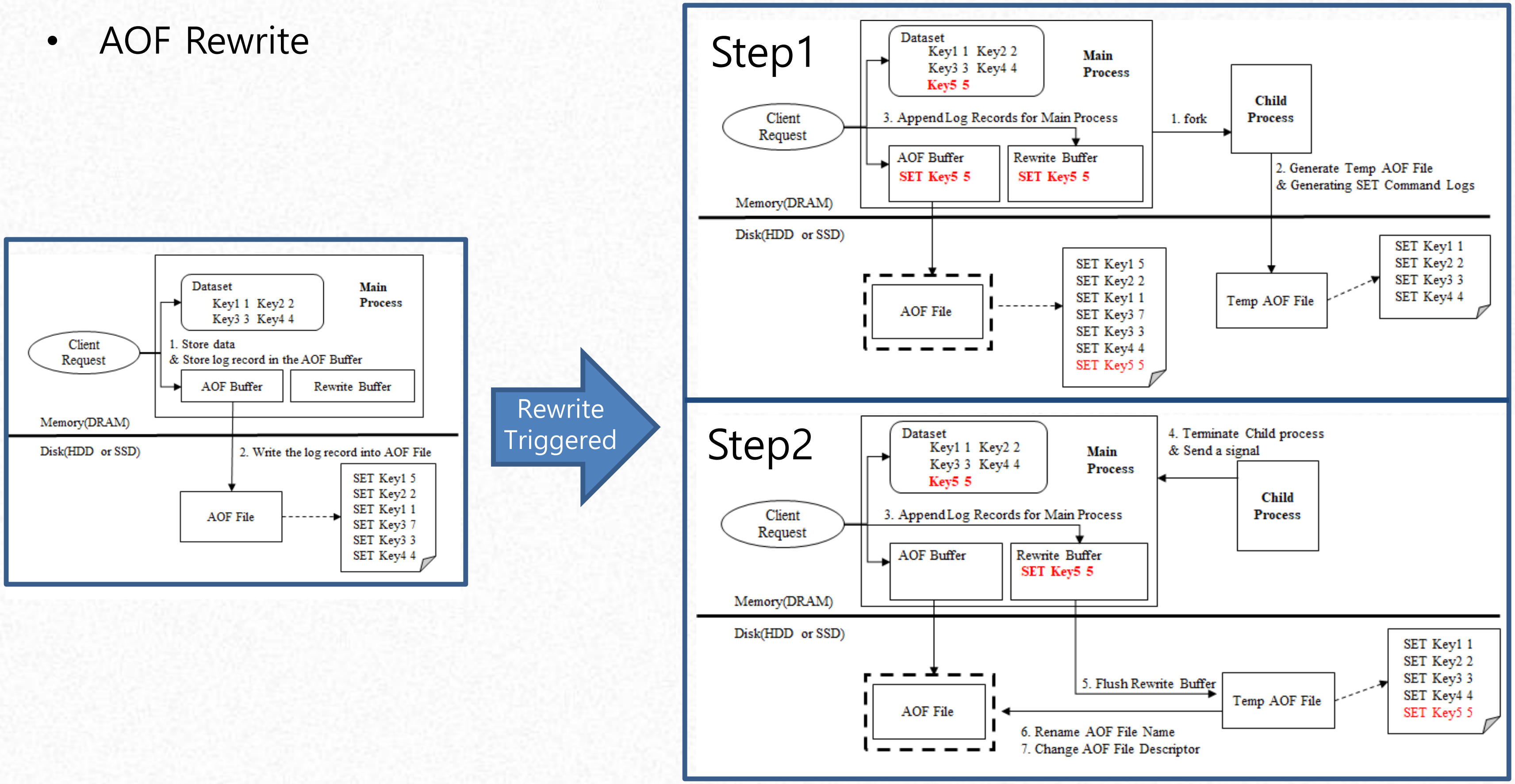
- AOF(Append-Only-File)



  - Append log records in an AOF file when command to change memory status
  - Advantage
    - Ensure data persistence higher than RDB
  - Disadvantage
    - Slower performance than RDB
    - Large file size (Data insertion ➔ Increase File size)
      - Slower recovery speed than RDB
      - Can cause system failure

Background

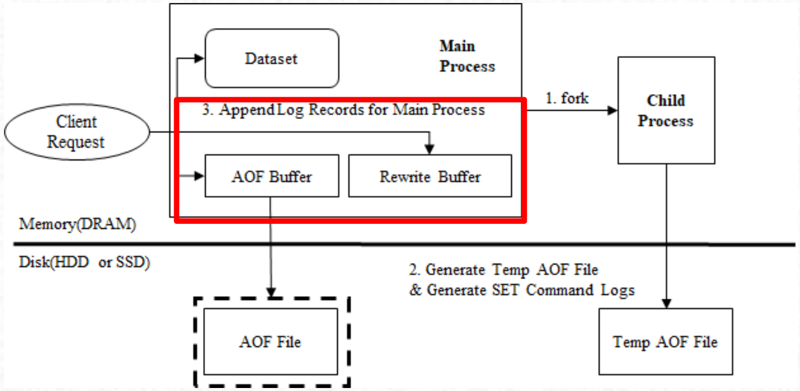
- AOF Rewrite



- Reconstruction method that reduces the size of an AOF File
- AOF Rewrite triggers when the file size is greater than the minimum size(Default : 64MB)
- Leave the command for the current data and remove the rest
- However, excessive memory usage and performance degradation occur during AOF Rewrite.

Motivation

- Memory Overhead

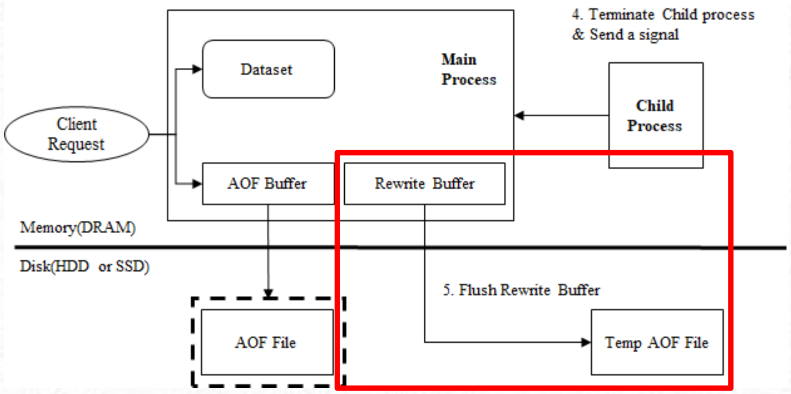


- The log record is stored in AOF Buffer and Rewrite Buffer during AOF Rewrite

➔ Increase memory usage

- AOF Rewrite causes an out-of-memory to occur with limited memory capacity

- Throughput degradation



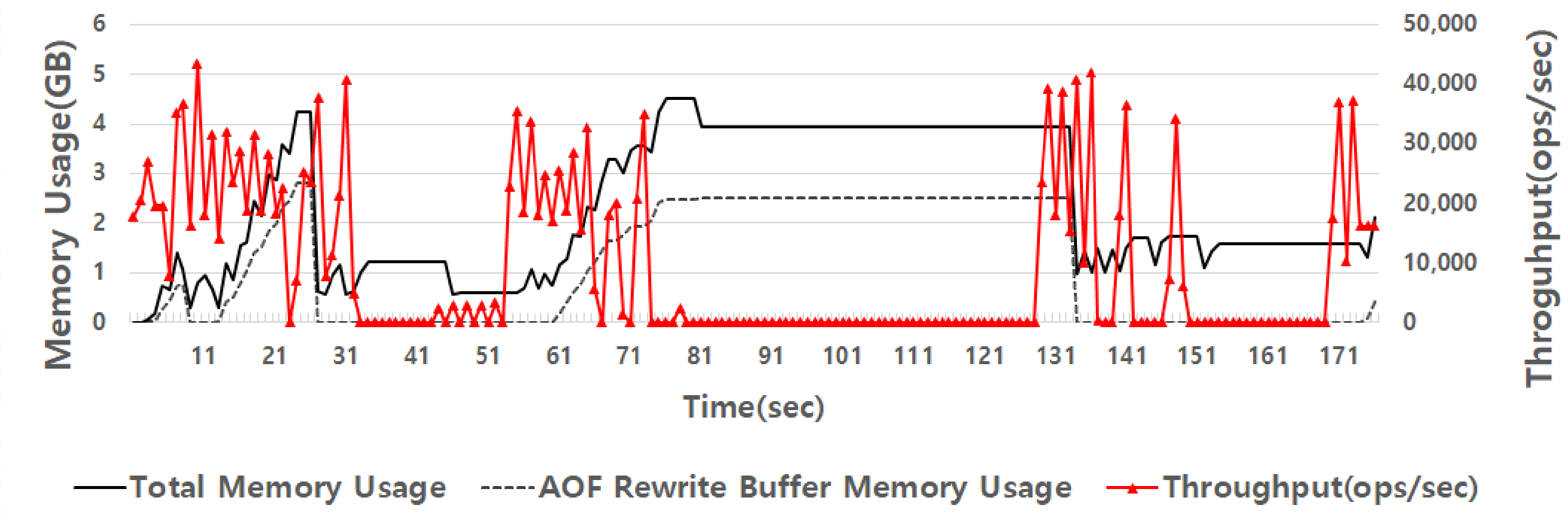
- Redis can process only one command at a particular time

- After creation of the temp AOF file, Flush the log records in Rewrite Buffer to temp AOF file

- The newly requested command is delayed until the flush operation is completed

➔ Decline of the overall throughput

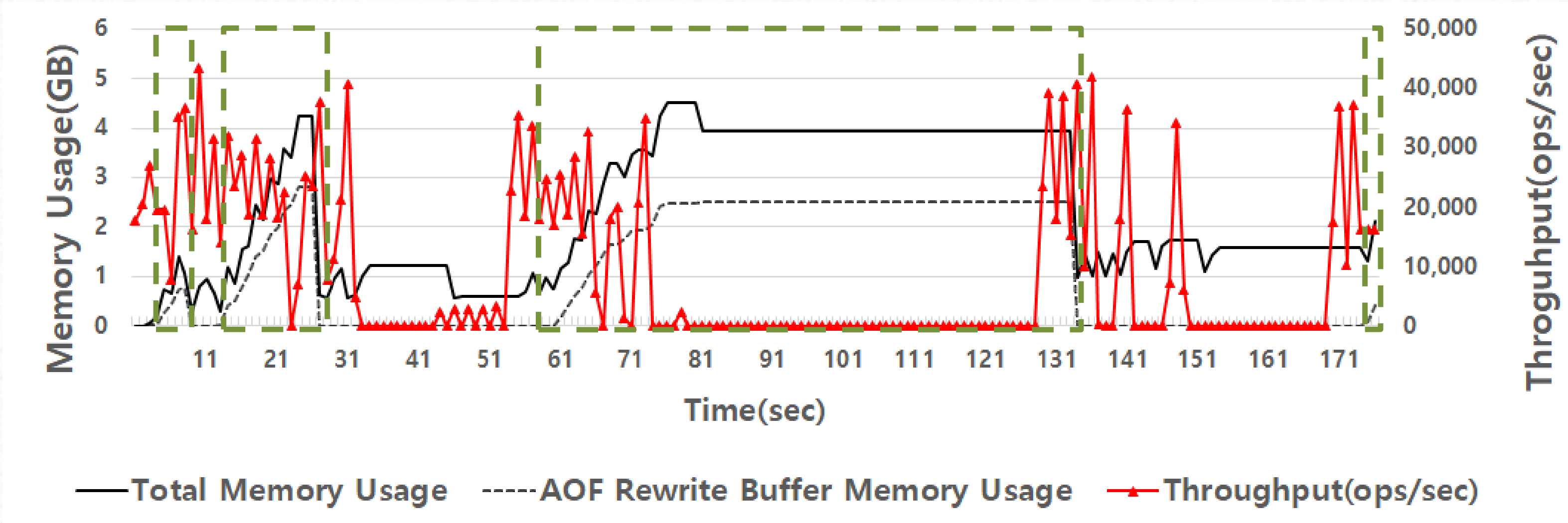
Motivation



- Simulate workloads of hot data that are frequently updated by users
  ➔ The effect of the AOF file can be identified

  - Key – 16B, Value – 10KB
  - 100,000 SET Commands, 900,000 Duplicate SET Commands

Motivation



- Simulate workloads of hot data that are frequently updated by users
  - Key – 16B, Value – 10KB
  - 100,000 SET Commands, 900,000 Duplicate SET Commands

- AOF Rewrite occurred four times
  - ✓ 3-8s, 13-26s, 60-133s, and 175-177s

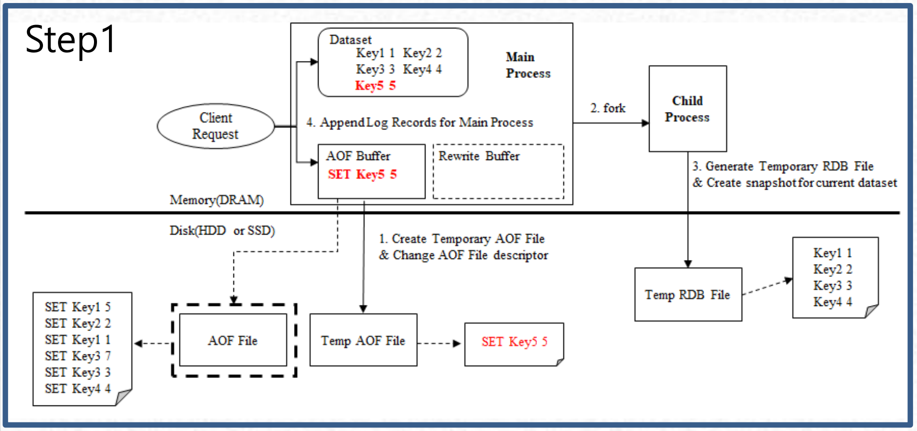
- Increase AOF Rewrite Buffer memory usage ➔ Increase Total memory usage

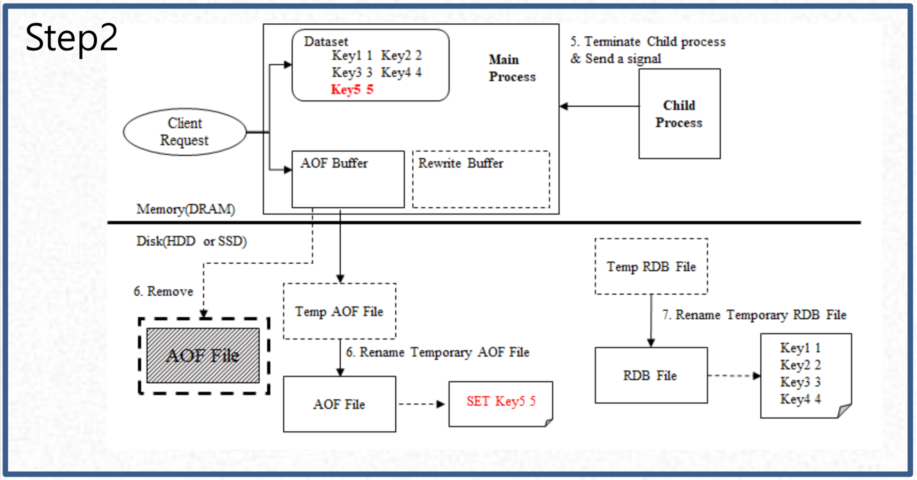
- Increased memory usage and performance degradation during AOF Rewrite

03 LESS

Logging Exploiting SnapShot

- LESS Operation





- Design to reduce memory usage and improve the performance

- The child process…
  - Create a snapshot file with a small file size and fast recovery speed

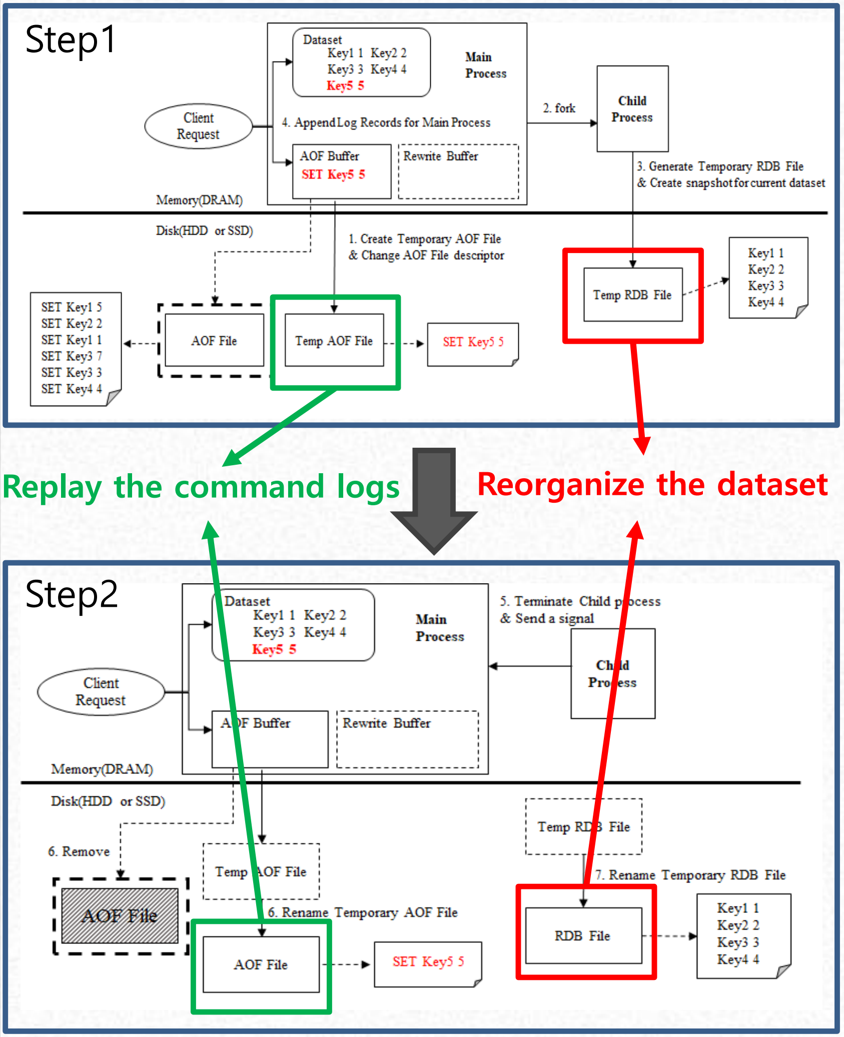
- The main process…
  - Perform AOF logging for the newly requested operation

- Difference(vs AOF Rewrite)
  - Child process generate **RDB** File
  - Do not use **Rewrite Buffer**
  - **Not merge** file created by child process and contents of Rewrite Buffer

- Advantage(vs AOF Rewrite)
  - Relatively **Light** disk I/O
  - **Small** memory usage
  - **Fast** performance

Logging Exploiting SnapShot

- LESS Recovery



- Use 4 types of files for complete data recovery
  - ✓ AOF File, Temp AOF File, RDB File, Temp RDB File

- System failure occurred before LESS was first run
  1. **AOF** File

- System failure occurred after LESS was first run
  1. **RDB** File
  2. **AOF** File

- System failure occurred between creation of Temp AOF File & Temp RDB File
  1. **RDB** File
  2. **AOF** File
  3. **Temp AOF** File

- System failure occurred during generation of RDB
  1. **RDB** File
  2. **AOF** File
  3. **Temp AOF** File

- System failure occurred after Temp AOF File renamed
  1. **Temp RDB** File
  2. **AOF** File

- System failure occurred after Temp RDB File renamed
  1. **RDB** File
  2. **AOF** File

04 Evaluation

Evaluation

- Configuration

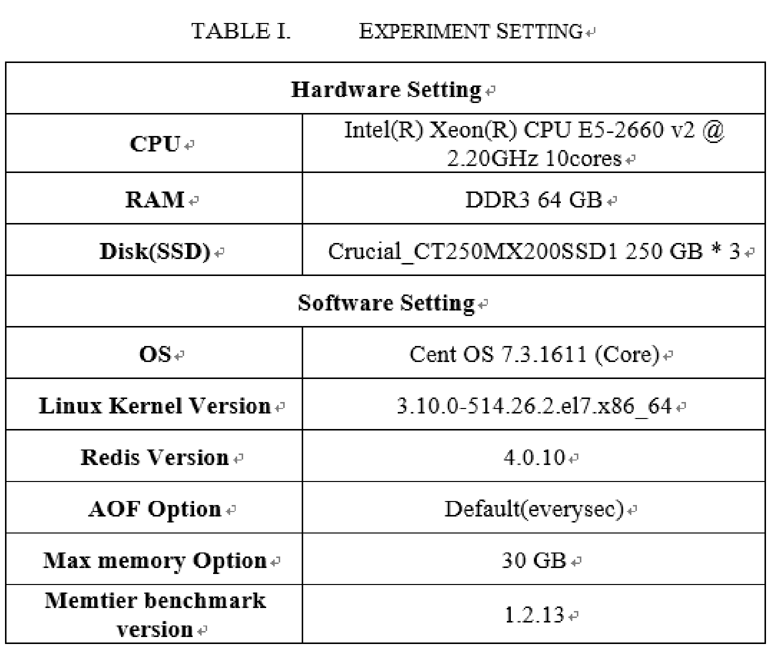


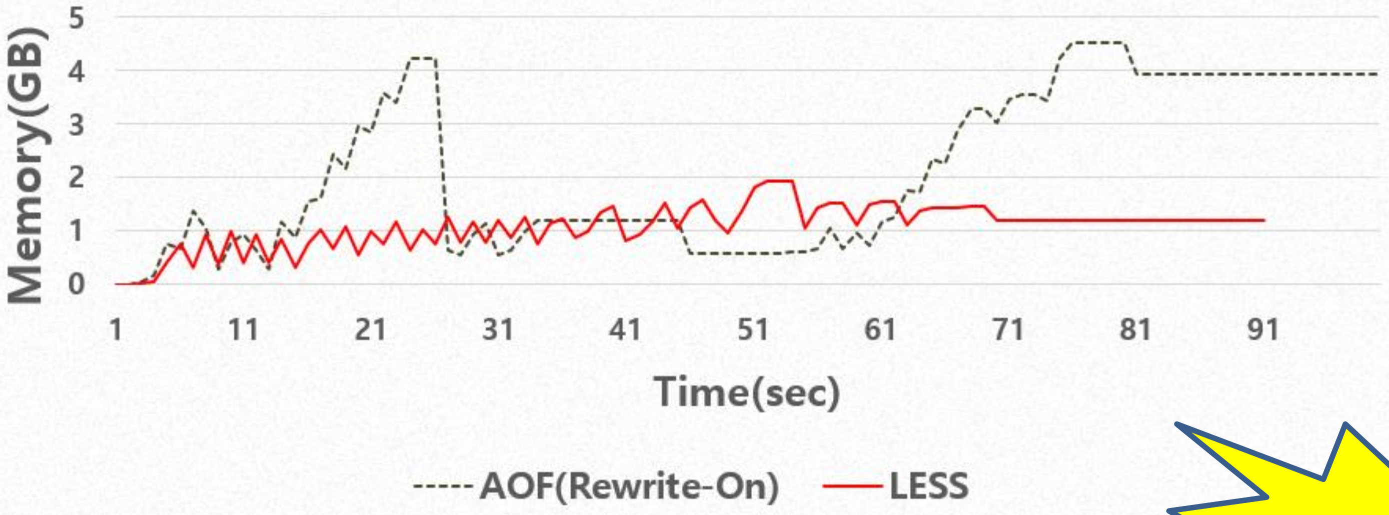
TABLE I. EXPERIMENT SETTING

Hardware Setting	
CPU	Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz 10cores
RAM	DDR3 64 GB
Disk(SSD)	Crucial_CT250MX200SSD1 250 GB * 3
Software Setting	
OS	Cent OS 7.3.1611 (Core)
Linux Kernel Version	3.10.0-514.26.2.el7.x86_64
Redis Version	4.0.10
AOF Option	Default(everysec)
Max memory Option	30 GB
Memtier benchmark version	1.2.13

- Benchmark
  - Memtier-Benchmark - v1.2.13
- Workloads
  - Simulate workloads where updates occur frequently
  - Key : 16B, Value : 10KB
  - SET : Duplicate SET = 100,000 : 900,000
- Redis configuration
  - Redis - v4.0.10
  - Maxmemory - 30GB
  - AOF *fsync* option - Everysec
  - Save - No

Evaluation

- Memory Usage



57% Reduction!!

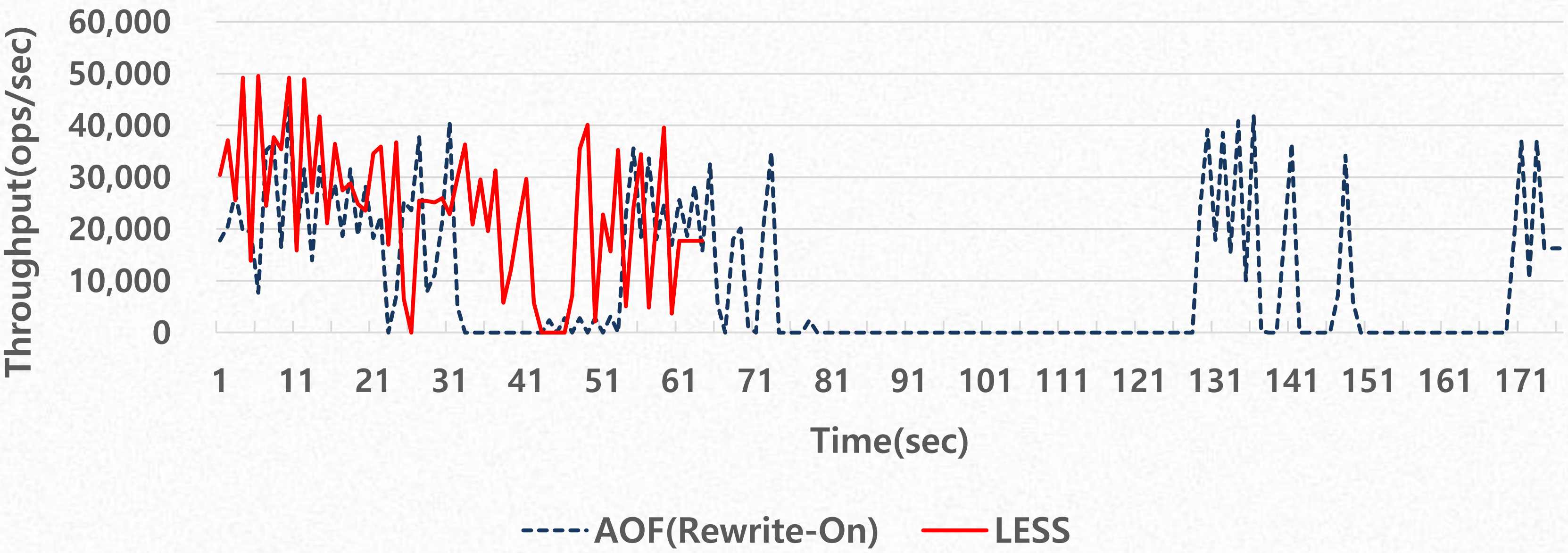
- Use the same workload as the workload used in motivation
- Average memory usage of LESS is less than AOF(Rewrite-on)
  - LESS write the log record in AOF directly

    ➔ LESS does not use AOF Rewrite Buffer
  - Can store more data than the original persistence method
- LESS has no time during which the use of the memory increases rapidly
- LESS is safe from out-of-memory

	Average Memory usage	Maximum Memory usage
AOF	2.6GB	4.5GB
LESS	1.1GB	1.9GB

Evaluation

- Throughput



- LESS completed workload in 64s, whereas AOF mode completed the workload in 177s
- The LESS method, the child process generates a RDB file as opposed to an AOF file
- Since the merge operation in AOF Rewrite is not performed, the throughput is improved

➔ Process more commands per second

05 | Conclusion

Conclusion

- Data persistence method is required due to volatile characteristics of Dram
  - RDB(Snapshot) : Generate point-in-time snapshots of a dataset at specific intervals
  - AOF : Append log records in an AOF file when command to change memory status
  - AOF Rewrite : Reconstruction method that reduces the size of an AOF File
    - ❖ **Issues**
      - *Memory overhead* : Increase memory usage due to AOF Rewrite Buffer
      - *Throughput degradation* : Newly requested command delayed due to *Flush* operation

- Logging Exploiting SnapShot : High performance & Low memory usage
  - The child process generates a RDB file as opposed to an AOF file
  - The main process write log record for newly requested command to AOF File without using Rewrite Buffer
  - Manages RDB file and AOF without MERGE, eliminating FLUSH operation

- Evaluation
  - 57% reduction in average memory usage & maximum memory usage
  - 2.7x performance improvement than an AOF Rewrite method
  - Outperform in-memory Key-Value stores and ensure data persistence

Thank You