SAMSUNG | SRCX *Root deep, reach high*

2019

Reducing Garbage Collection Overhead in SSD Based on Workload Prediction

Pan Yang, Ni Xue, Yuqi Zhang, Yangxu Zhou, Li Sun, Wenwen Chen, Zhonggang Chen, Wei Xia, Junke Li, Kihyoun Kwon

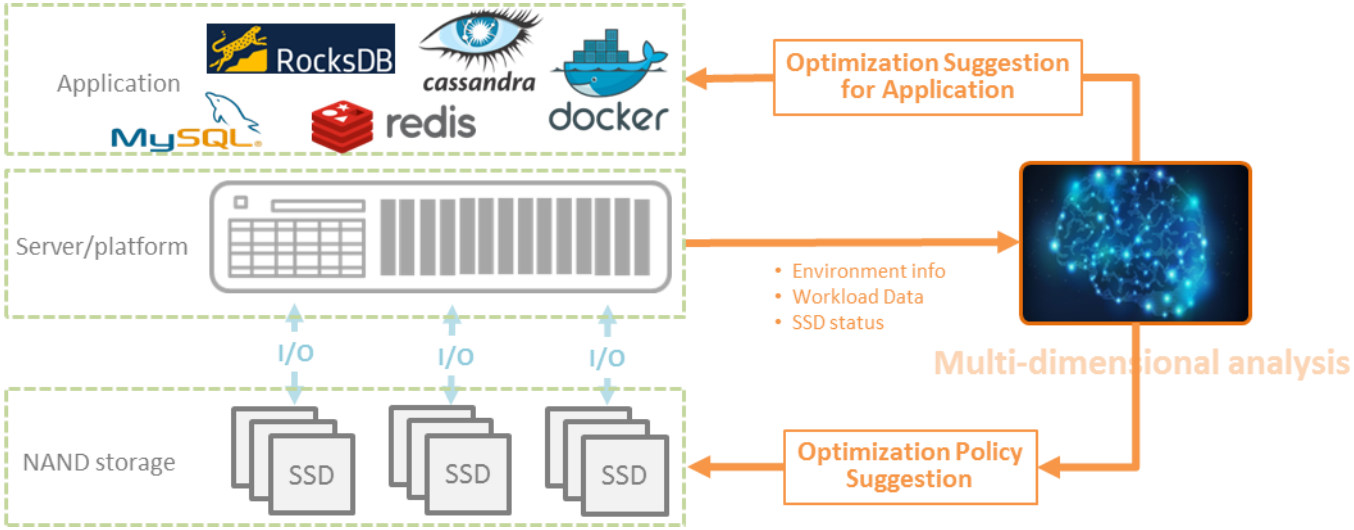
SAMSUNG R&D INSTITUTE CHINA XIAN

Background

- NAND flash-based solid-state drives (SSDs) have found wide use
  - 5 major issues in using SSD
    - Unexpected bad performance & QoS
    - Limited lifetime
    - Sudden failure
    - Increasing power consumption
    - Low capacity utilization
- Machine learning (ML) technology emerges to optimize SSD utilization

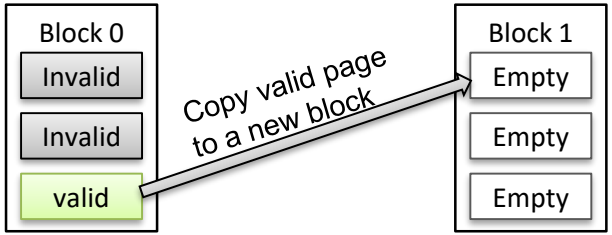
Vision

- Research goal:
  - Using ML to analyze workload pattern, and optimizing SSD/NAND storage in various scenarios with proactive learning and intelligent policy



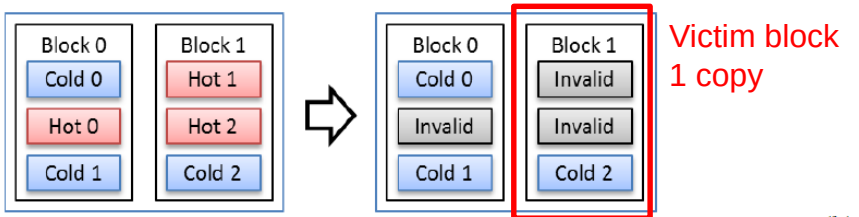
Introduction

- Garbage collection (GC) is an essential operation in SSD
- GC has a major impact on both performance and lifetime of SSD
  - Valid page copy
    - It consumes internal computing and bandwidth resources and decreases the I/O response performance
    - The more data are migrated, the quicker NAND flash will be worn out

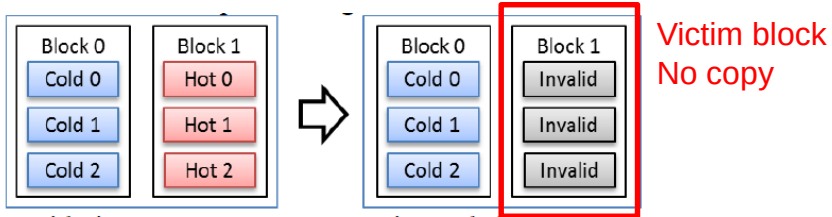


Introduction

- Reducing valid page copy to improve GC overhead
  - Choose victim block with least number of valid pages
  - Placing data according to different temperatures
    - Hot data : data are updated frequently
      - The write access count of LBA is high
    - Cold data: data are updated infrequently
      - The write access count of LBA is low



(a) Without considering temperature, the "Cold 2" page is valid and needed to be copied during GC.



(b) Considering temperature, no copy is needed for block 1.

Introduction

- Classifying data by temperature
  - The state-of-the-art
    - AutoStream (SYSTOR '17) / FStream (FAST'18) / PCStream (FAST'19)
  - Current solutions only focus on detecting current data temperature
    - Future temperature should be taken into account in data placement
    - **Predicting the future data temperature helps to guarantee accuracy**

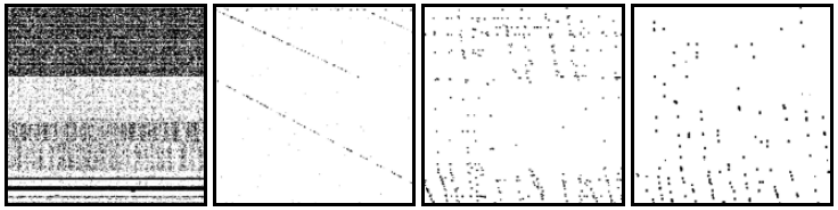
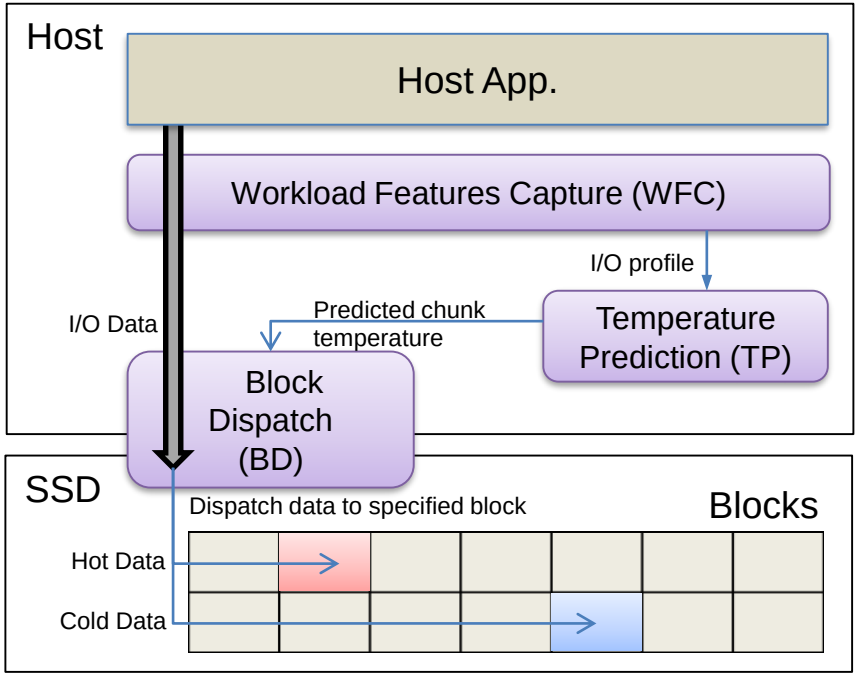


Fig. 2. Different gray levels in vertical means different temperature in different regions, while different gray levels in horizontal means temperature is changing over time.

Scheme Design

- Scheme architecture
  - Workload features capture (WFC)
    - To capture workload profiling data
  - Temperature prediction (TP)
    - To predict the future temperature
  - Block dispatch (BD)
    - To dispatch write requests to different blocks



Scheme Design

- Workload features capture (WFC)
  - Tool – StoneNeedle†
    - Capturing statistic workload features
      - » e.g., throughput, bandwidth, I/O size/count, and time interval.
    - Deployed in the device driver
    - It's open source
    - To construct a workload-profiling-data-standard together with all players

† **Git repo: https://github.com/Samsung/StoneNeedle**

Scheme Design

- Workload features capture (WFC)
  - Reducing data volume
    - Dividing entire LBA into different chunks
      - » All the I/O requests falling in the same chunk will be treated as the feature of this chunk.
    - Choosing key features
      - » Choosing features that are closely correlated with temperature
    - In each fixed period of time, the temperature and features are recorded as:

      $\boldsymbol{rec}_t(f_{1t}, f_{2t}, f_{3t}, ..., T_t)$,

      T_t — temperature of the t^{th} time period

      f_{it} — feature value of the t^{th} time period.

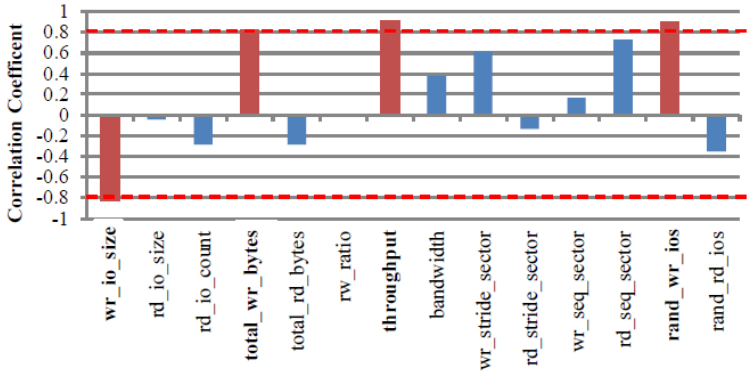


Fig. 4. Correlation coefficient in RocksDB

Scheme Design

- Temperature prediction (TP)
  - Problems
    - The temperature may change sharply in different periods
    - A chunk's temperature may be related to other chunks
    - The temperature may also be impacted by other features
  - Algorithm : Long Short Term Memory (LSTM)
    - LSTM can comprehensively consider multiple factors for prediction

Scheme Design

- Temperature prediction (TP)
  - LSTM can be seen as a function

$$T'_{t+1}=\mathrm{LSTM}(\boldsymbol{R}_{t-k+1,k}) \qquad (1)$$

    - Input: the records of *k* time periods
      - » the correlated features f
      - » the real temperature T
    - Output: the predicted temperature of the next time period T'_{t+1}

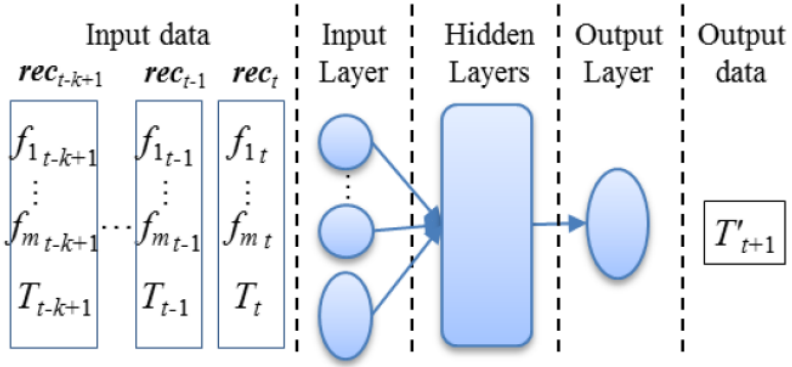
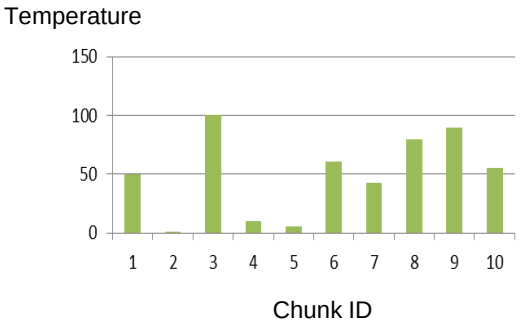
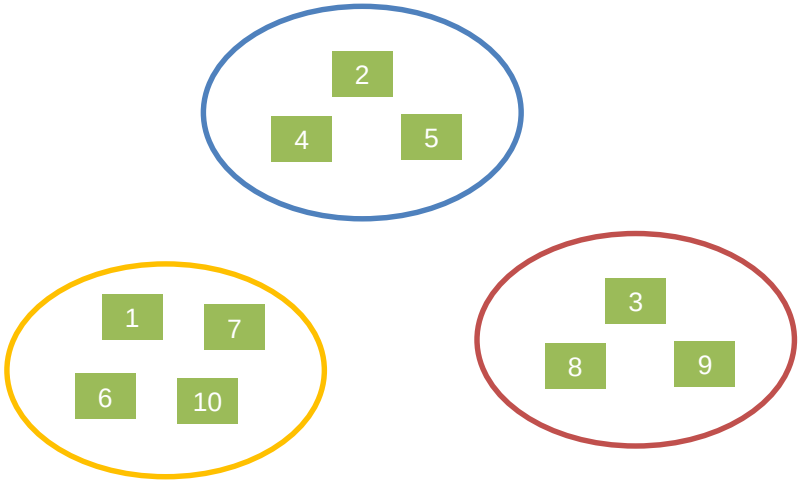


Fig. 5. Temperature prediction by using LSTM

Scheme Design

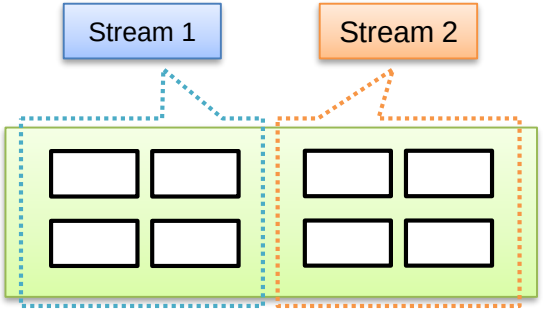
- Physical NAND block dispatch (BD)
  - How to place data with the predicted temperature?
    - Dividing chunks into different temperature ranges
    - Mapping the chunks with similar temperature to the same block
  - Algorithm : K-Means
    - High efficiency

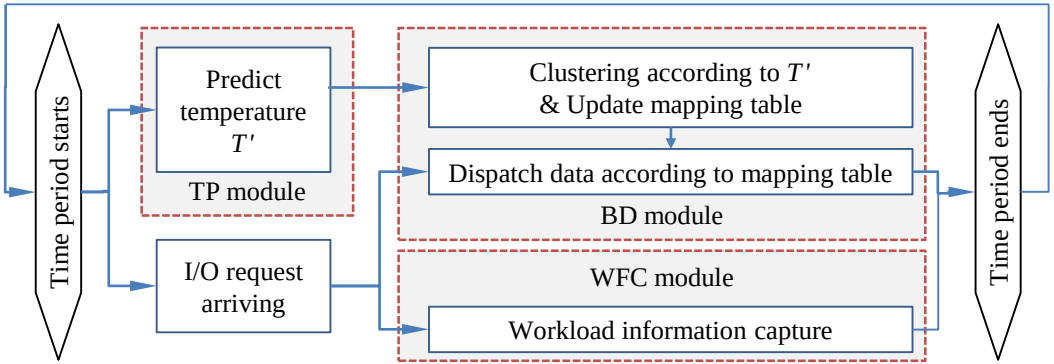




Scheme Design

- Implementation
  - Support of SSD
    - Directives and Streams (NVMe Spec. 1.3)
    - Samsung Multi-Stream SSD (HotStorage'14)
  - LSTM-training offline/online
  - Prediction and dispatch
    - WFC outputs $\boldsymbol{rec}_t$ data
    - TP predicts T'_{t+1}
    - BD dispatches block for I/O according to T'_{t+1}





Prediction and dispatch process

CID	SID
1	3
2	6
…	…

Evaluations

- Environment

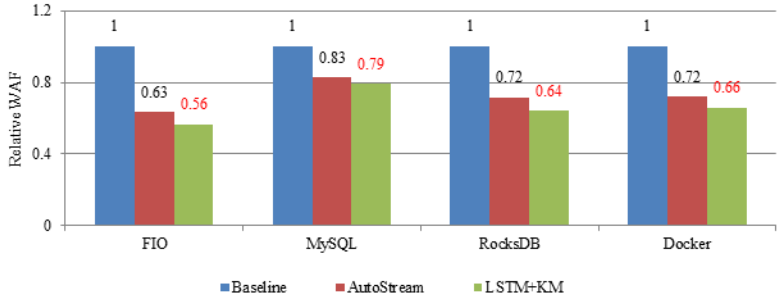
Table 1. Evaluation system configuration

Processor /Memory	**Processor Dual Socket:** Intel (R) Xeon (R) CPU E5-2620 v4 @ 2.10GHz/16 cores **Total Logical CPU:** 32 **Total Memory:** 64 GB **GPU:** NVIDIA GTX 1080, 8G
Operating System	**Distro:** CentOS Linux release 7.5.1804 (Core) **Kernel:** 4.4.2, patched for multi-stream support **Arch :** x86_64
SSD	**SSD:** Samsung NVMe PM963 2.5”, 960GB (Support both “normal” and “multi-stream” mode with 8 streams, with NAND write and host write values in additional S.M.A.R.T)

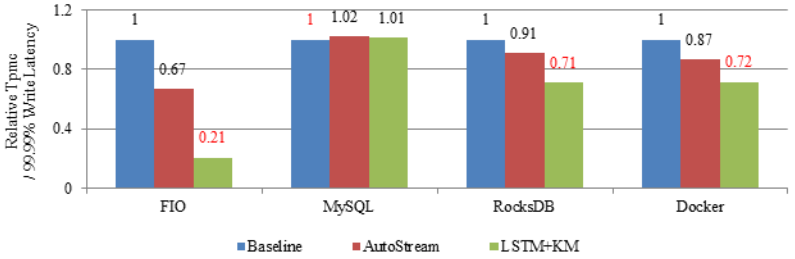
- Parameters
  - The number of the re-quired cluster is 8.
  - Each time period lasts for 5 seconds.
  - The time step k of LSTM is 5.
  - The entire address is divided into 10,000 chunks.

Evaluations

- Scheme effectiveness
  - Workload
    - FIO
    - MySQL
    - RocksDB
    - Docker
  - Comparisons
    - Baseline
    - AutoStream
    - Our scheme (LSTM+KM)
  - Measurement – WAF & latency



WAF in different applications



Performance in different applications

The performance and write amplification factor (WAF) are improved in various applications.

Evaluations

- Resource consumption
  - It takes less than 80ms to generate a new mapping table from the captured features
  - The GPU utilization of LSTM is less than 17%
  - Our scheme consumes roughly as many resources as legacy SSD with the difference within 5%.

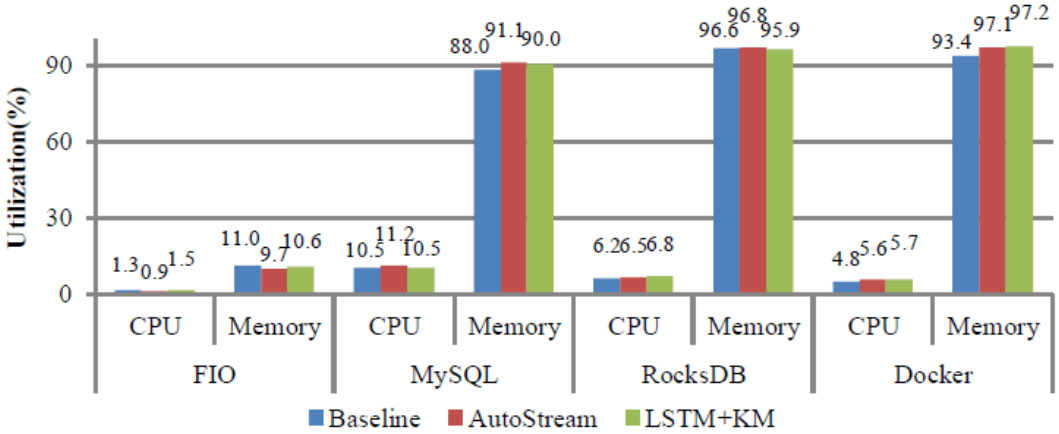


Fig. 9. Resource consumption of CPU & memory

Conclusion

- Achievements
  - We explored the use of machine learning to improve GC overhead
  - We developed a powerful workload information capture tool - StoneNeedle
  - The lifetime and performance of SSD are improved effectively

- In the future, we will focus on the following:
  - Improving the efficiency and accuracy of our proposed machine learning models
  - Optimizing the scheme on host FTL approach

THANK YOU

2019 Root deep, reach high

Email: pan87.yang@samsung.com