Platform-Agnostic Lightweight Deep Learning for Garbage Collection Scheduling in SSDs

Junhyeok Jang, Donghyun Gouk, Jinwoo Shin, Myoungsoo Jung

Computer Architecture and Memory systems Laboratory
**Motivation**

How to hide GC latency?
- Let’s perform GCs at user idle times!

*How long will be the user idle times?*
Hiding GC latency: Background GC

Request

Time

SSD

I/O

Wait

Garbage Collection

Common Assumption:
Storage won’t be touched in the near future!
Hiding GC latency: Background GC

*Real workload from MS Production Server (https://trace.camelab.org/)

Request

SSD

I/O

Garbage Collection

Wait

Time

Assumption

Real workload*

*Real workload from MS Production Server (https://trace.camelab.org/)
**GC-Tutor**

**DNN-based GC scheduler**

- Precisely predict future request arrivals
- Schedules GC in user-invisible time
- Consistently accurate regardless of workload with lightweight online learning mechanism
DNN-based GC Scheduling

I/O Pattern

DNN Model

DNN-based Idle Time Prediction

Background GC

Problem:
A fixed DNN model fails to predict unseen workloads
DNN-based GC Scheduling

Problem:
A fixed DNN model fails to predict unseen workloads

Online Learning!
Naïve

Takes more than a few hours

Infeasible!

Meta Learning*

Offline

Online

I/O traces

Meta Learning

Deeplearn

GCTutor

Online Learning

CAMELab

*Chelsea Finn, et al., Model Agnostic Meta Learning for Fast Adaptation of Deep Networks, ICML 2017
**Evaluation**

**GC-Tutor can accurately predict idle time**
- Consistently higher accuracy on trained workloads
- Significantly higher accuracy on unseen workloads
  - prxy, stg:
    Very different idle time distribution compared to trained workloads

**GC-Tutor can reduce GC-induced delays by 82.4%,**
on average, compared to rule-based GC scheduler
Conclusion: GC-Tutor

DNN-based GC scheduler
- Accurate request arrival prediction using DNN model
- Meta learning-based light-weight online learning mechanism

Making GC overhead invisible to users!
Thank you!

Junhyeok Jang
Electrical Engineering, KAIST