

ZNS SSD를 활용한 컨테이너 체크포인트 성능 간섭 최소화

한예진, 오명훈, 최종무

KCC 2022

Dankook University

2022. 06. 29

Presented by Myunghoon Oh

snt2426@dankook.ac.kr

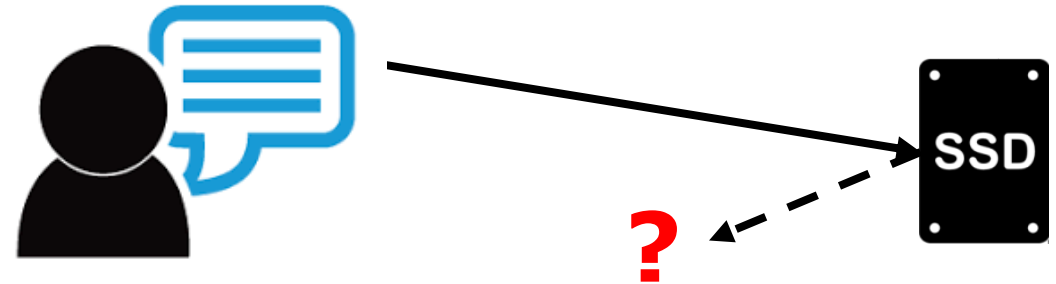
Introduction

- Predictable performance is an important requirement for computing system
 - Users expect the response within predicted latency
 - For clients' satisfaction, interactive web services or cloud environment need to achieve stable performance



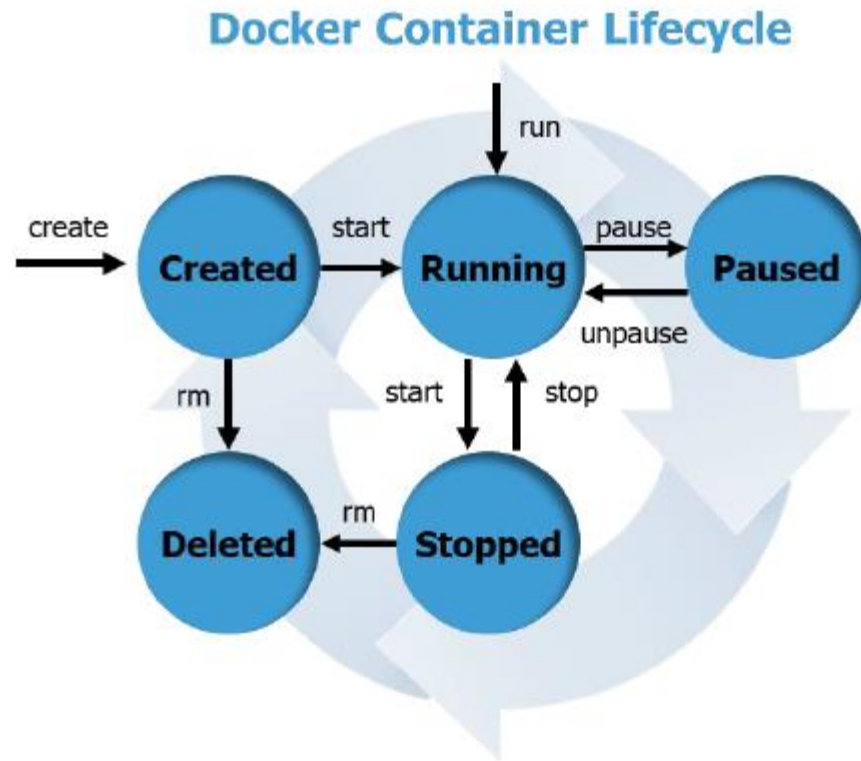
Introduction

- Predictable performance is an important requirement for computing system
 - Users expect the response within predicted latency



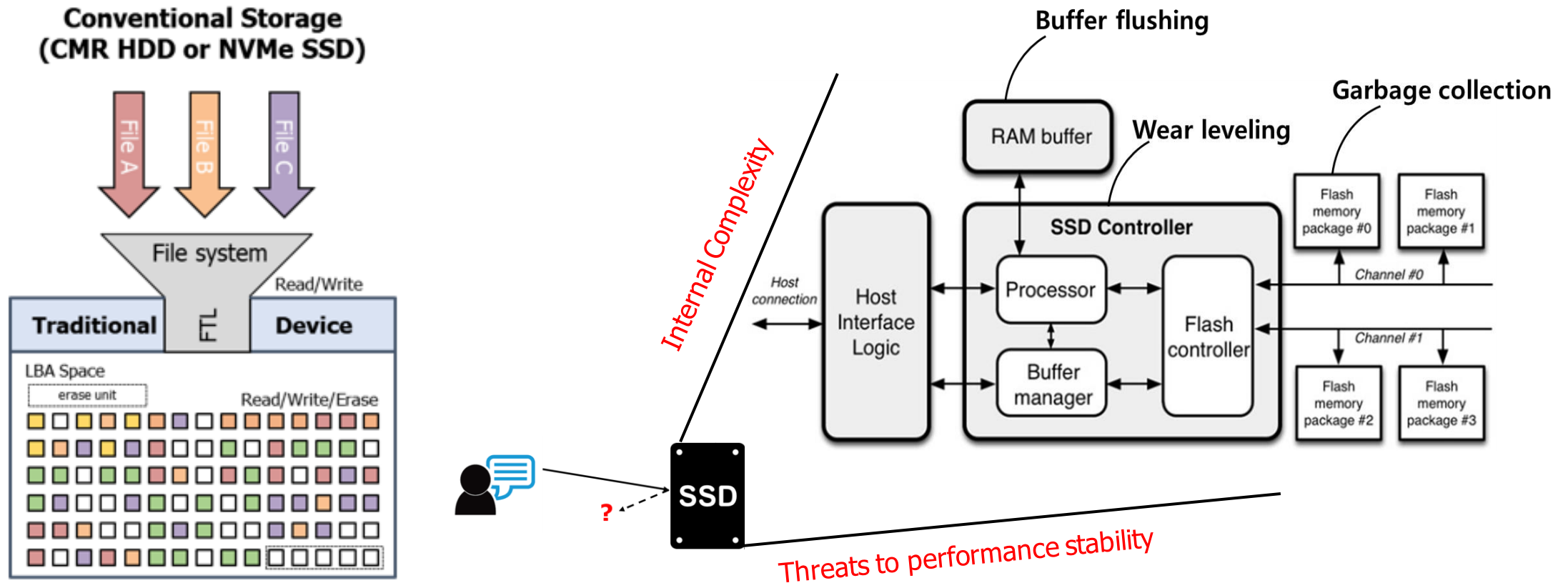
Introduction

- For performance and reliability, C/R is needed on cloud environment
 - Container lifecycle management is essential where containers are running on the machine
 - Large & Sequential writes occur during the checkpoint process



Motivation

- Problem: Unpredictable Latency due to performance interference
 - Traditional SSDs cannot guarantee QoS between different jobs

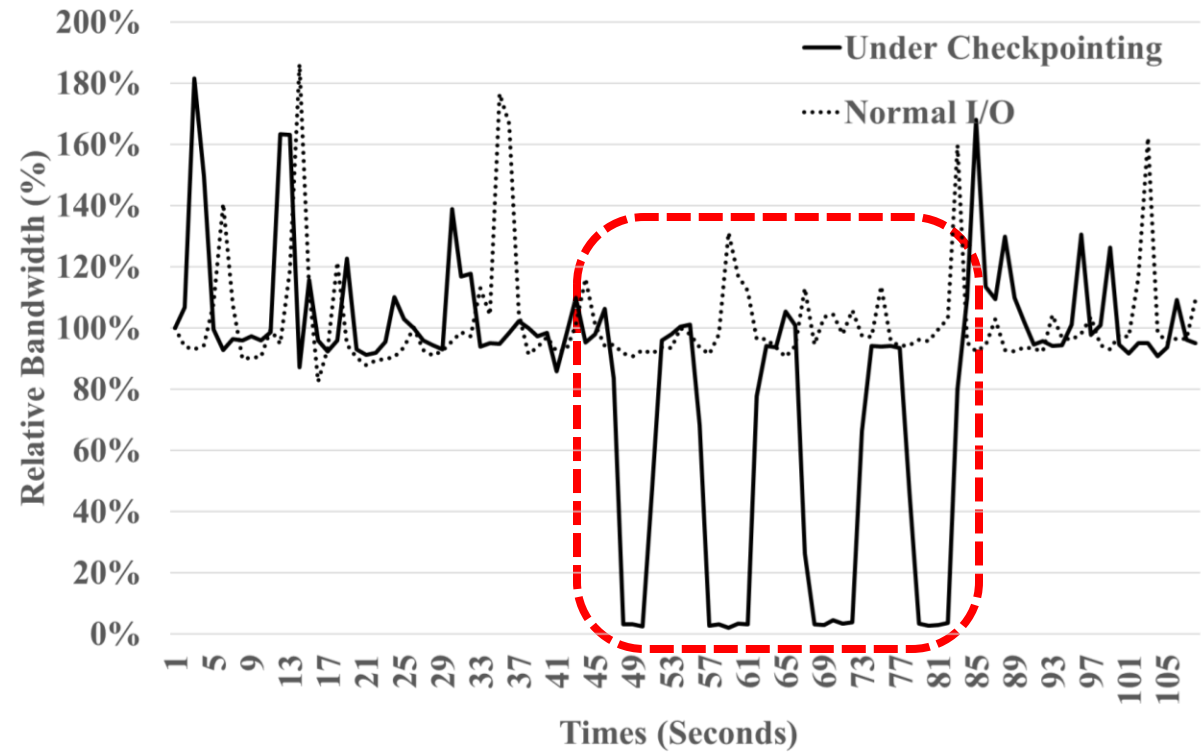
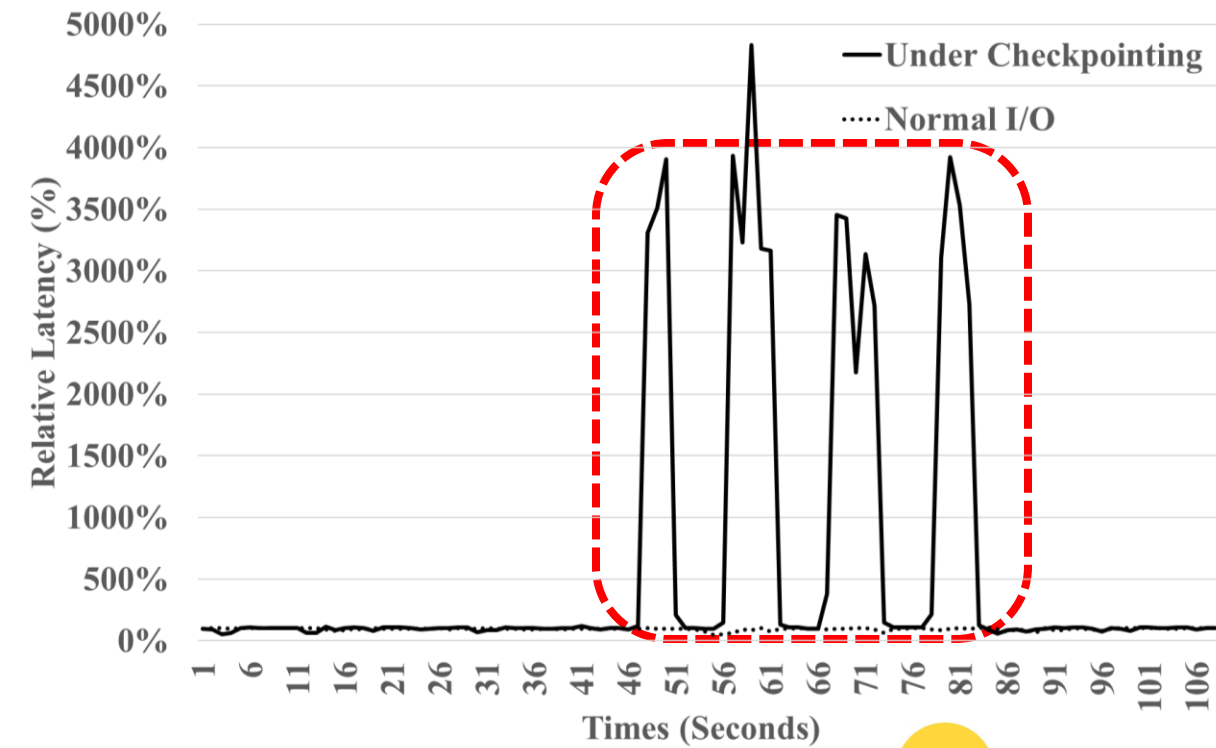


M. Hao et al., "LinnOS: Predictability on Unpredictable Flash Storage with a Light Neural Network", OSDI, 2020

H. Li et al., "IODA: A Host/Device Co-Design for Strong Predictability Contract on Modern Flash Storage", SOSP, 2021

Motivation

- Problem: Unpredictable Latency due to performance interference
 - Traditional SSDs cannot guarantee QoS between different jobs

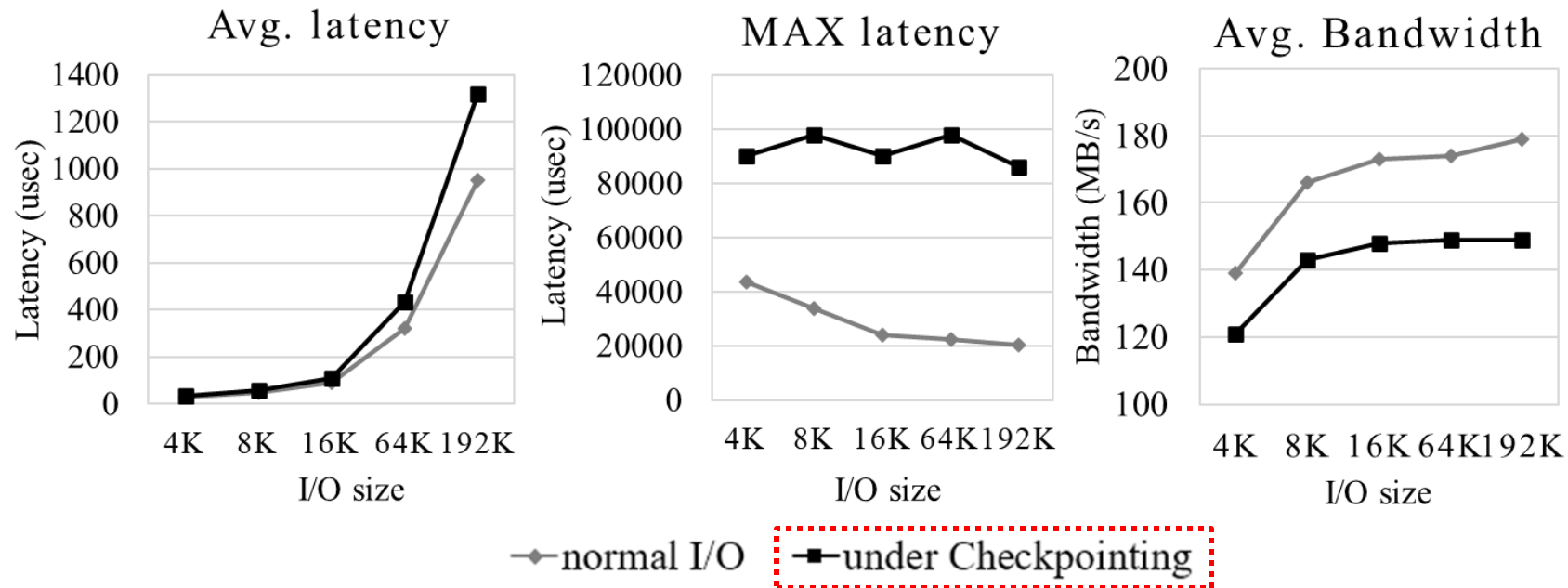


Unpredictable performance

Motivation

- Problem: performance interference for disk-intensive workloads during checkpointing job
 - Variable1. I/O size: degree to which I/O is being performed

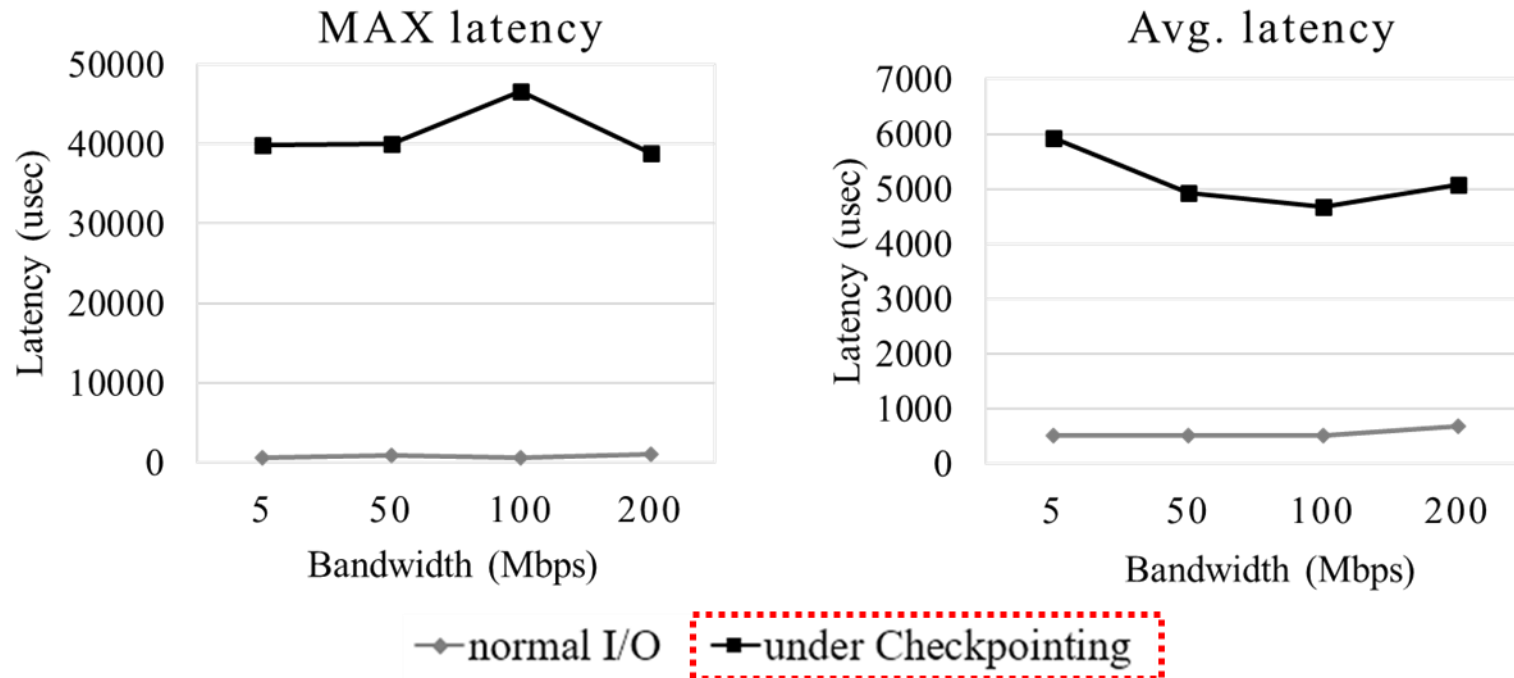
TrSSD Performance Interference – fio Random write with scaling I/O size



Motivation

- Problem: performance interference for disk-intensive workloads during checkpointing job
 - Variable2. bandwidth: degree to which I/O is being performed

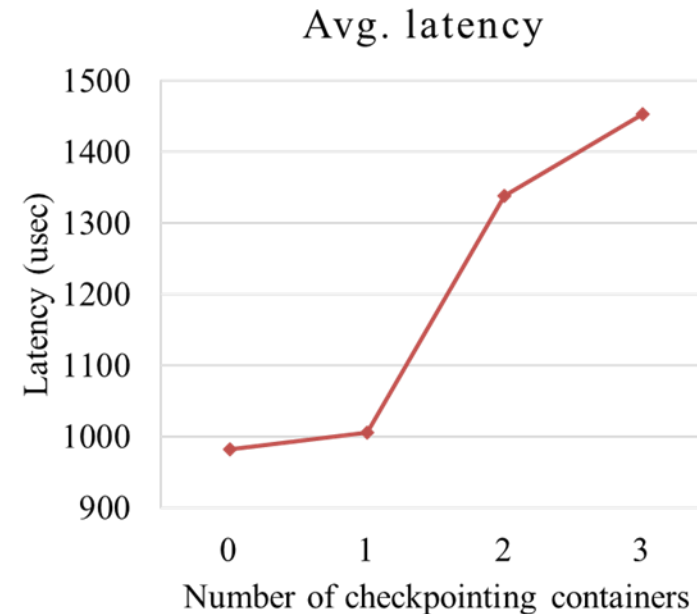
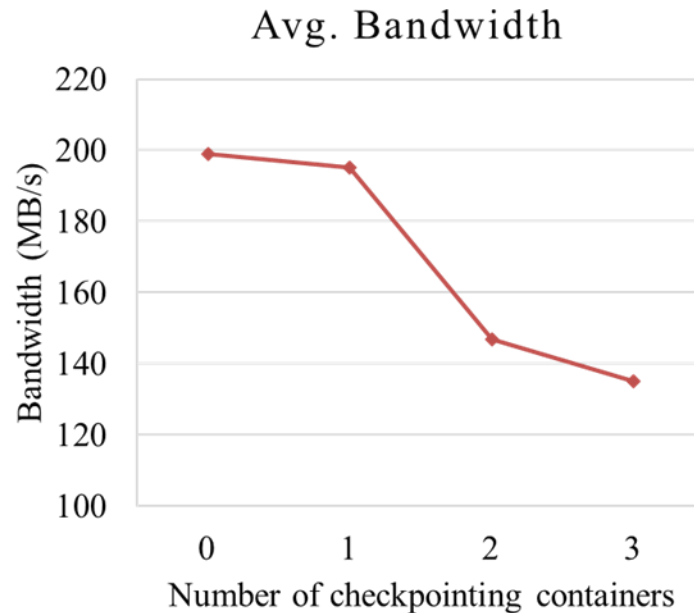
TrSSD Performance Interference – fio Random write with scaling bandwidth



Motivation

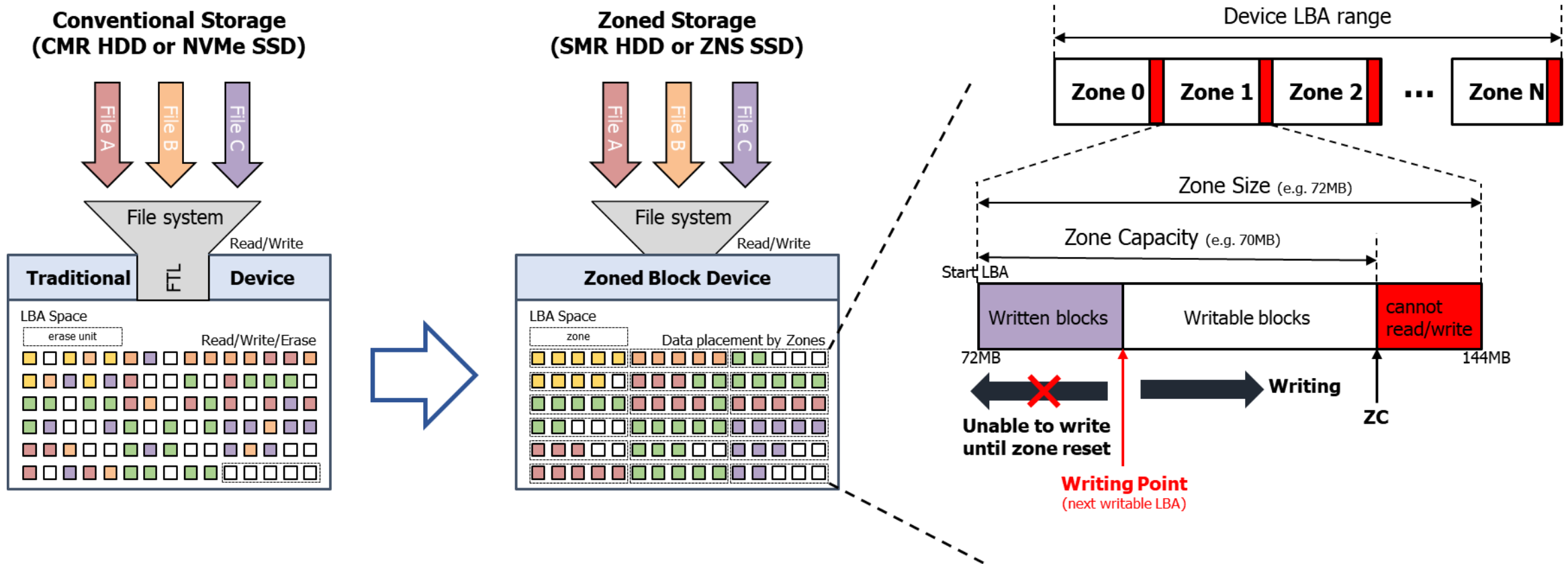
- Problem: performance interference for disk-intensive workloads during checkpointing job
 - Variable3. number of checkpointing containers (memory size fixing)

TrSSD Performance Interference – fio Random write 192K I/O



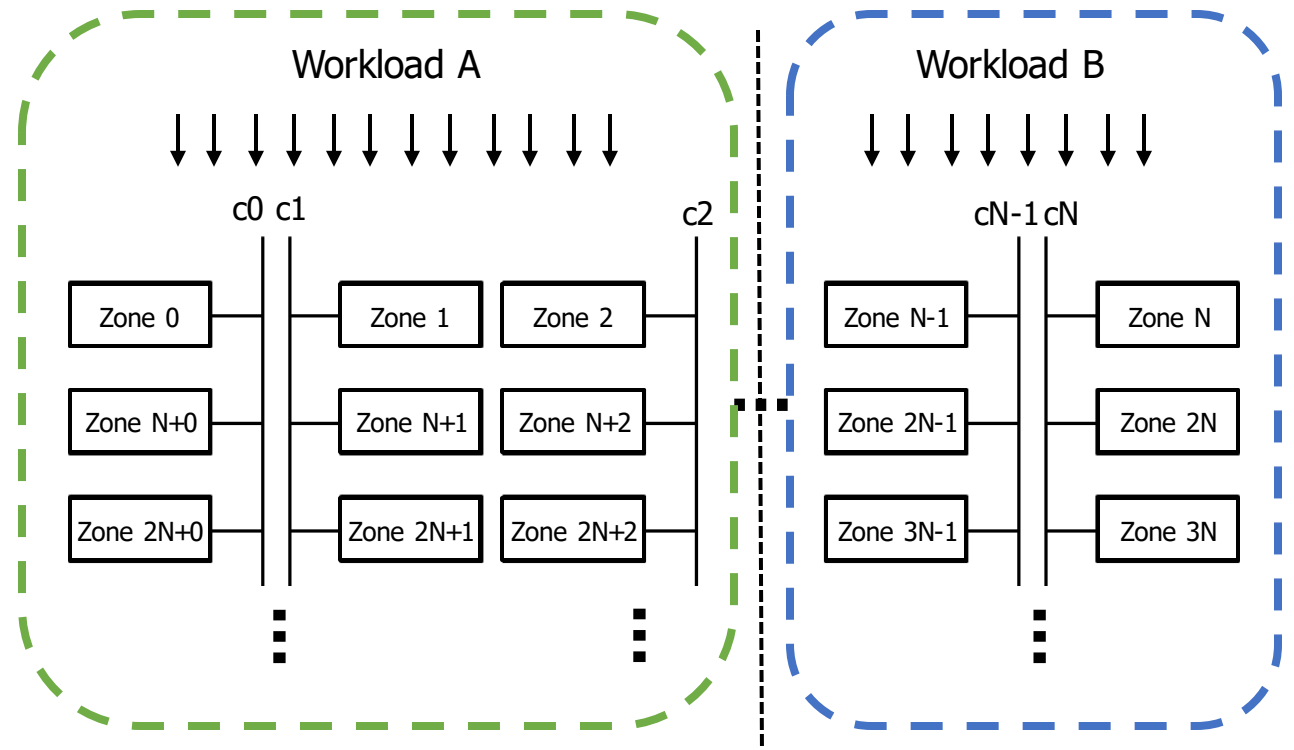
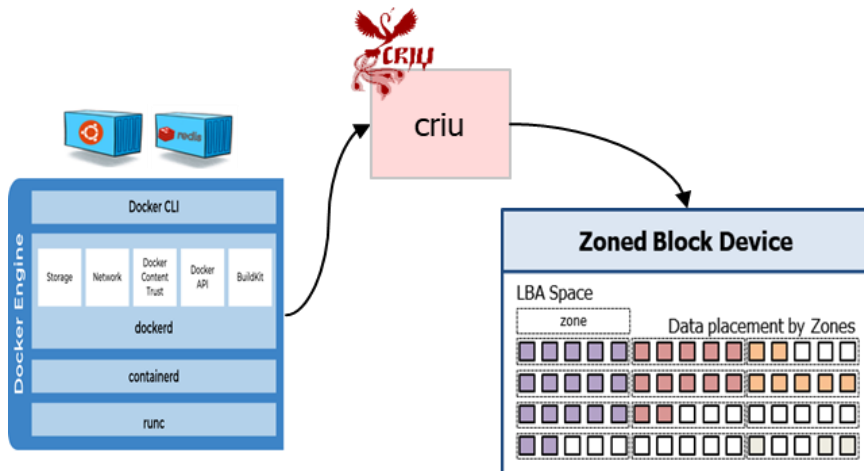
Motivation

- Solution: ZnSSD can guarantee QoS because host can collaborate on data placement



Design

- To improve isolation, separate checkpoint jobs and other write jobs on dedicated channels
 - Metadata image file, page image file, fio job



Evaluation

- Evaluation Environment

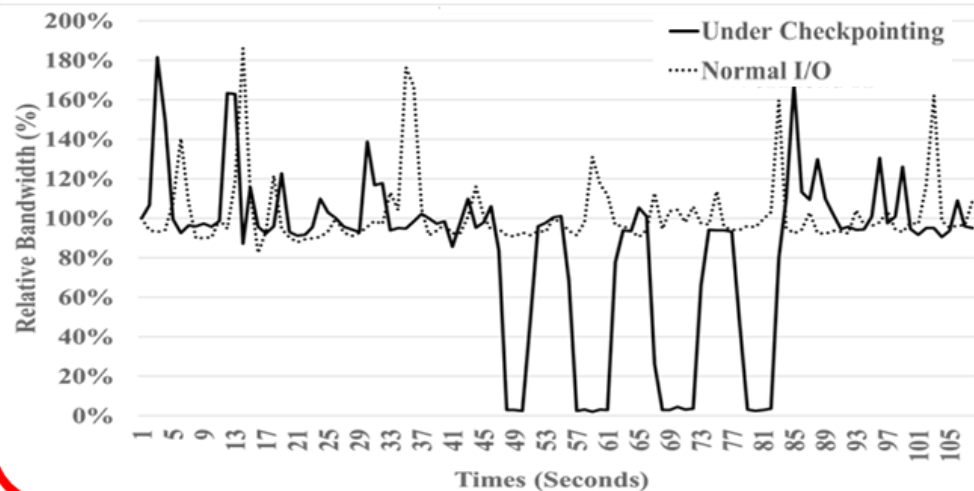
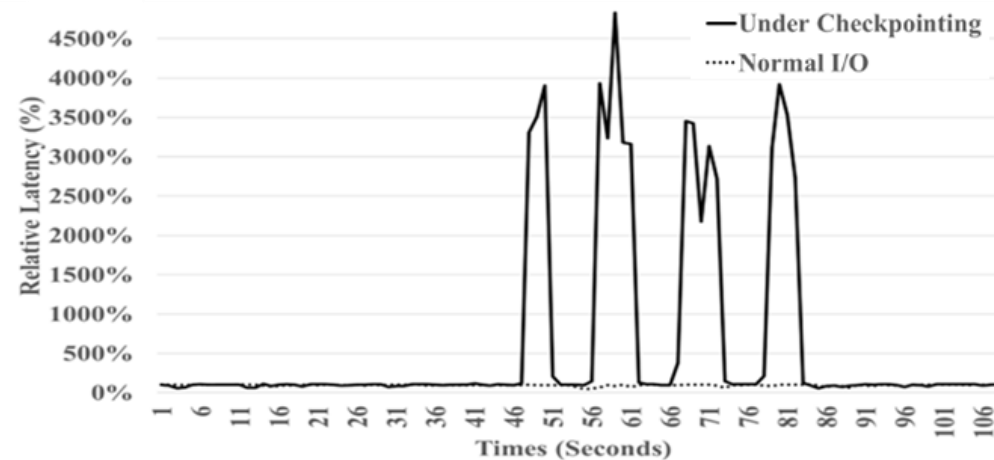
CPU	Intel ® Core™ i5-4440, 3.10Hz
Memory	32 GB RAM
Kernel	Linux 5.7.7
Storage	TrSSD: Samsung 850 Pro 256GB / ZnSSD: Prototype 2TB
Tool	CRIU v3.16.1

Evaluation

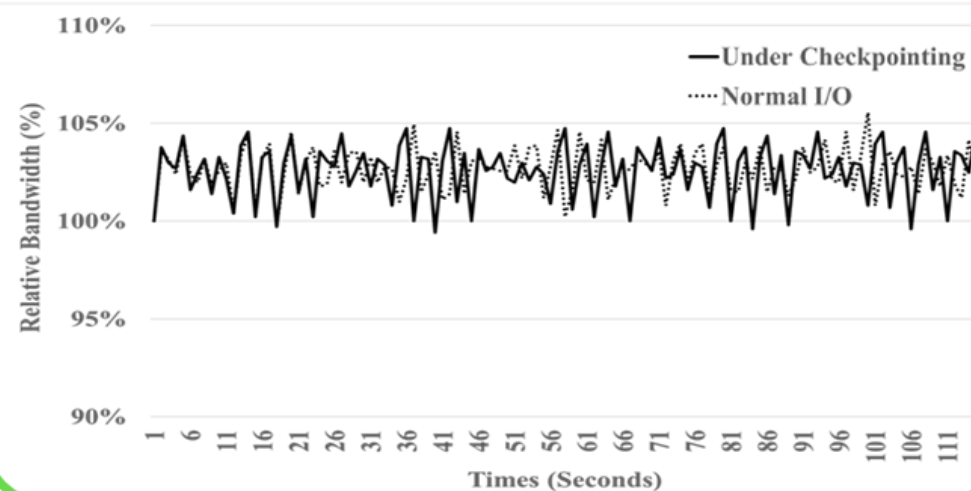
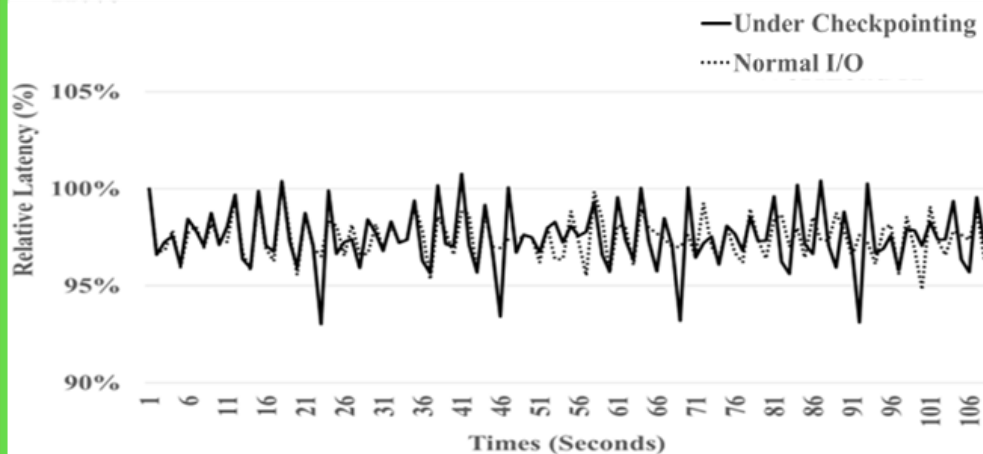
Latency

Bandwidth

Traditional SSD



ZNS SSD



Conclusion

- Predictable performance is essential in many modern computer environment
- Because of the intrinsic NAND characteristic, container checkpointing job can interfere with other write jobs
- By allocating dedicated channels for different write jobs, we can guarantee QoS

2022 KCC paper

ZNS SSD를 활용한 컨테이너 체크포인트 성능 간섭 최소화

한예진*, 오명훈, 최종무

단국대학교

{hyj0225, snt2426, choim}@dankook.ac.kr

Minimizing Container Checkpointing Performance Interference using ZNS SSD

Yejin Han*, Myunghoon Oh, Jongmoo Choi

Dankook University

요약

Docker는 수많은 컨테이너가 실행되고 있는 클라우드 환경에서 컨테이너의 라이프 사이클을 관리하기 위해 체크포인트를 수행한다. 그러나 물리적인 호스트 머신에서 수행 중인 컨테이너는 호스트 자원을 공유하고 있기 때문에 사용자가 수행 중인 I/O 작업이 체크포인트로 인해 간섭을 받아 성능이 저하될 수 있다. 또한 체크포인트 시 이미지 파일이 저장되는데, 이때 전통적인 SSD는 쓰기 작업을 수행할 때 채널 간 경쟁으로 기존 I/O 작업의 Latency는 50배 증가, Bandwidth는 40배 감소한다. 본 논문에서는 컨테이너 체크포인트로 인한 성능 저하를 분석하고 이를 해결하기 위해 ZNS SSD를 활용하여 체크포인트를 고립하는 방법을 제안한다. 제안하는 기법의 성능평가 결과 기존 SSD에 비해 ZNS SSD에서 체크포인트로 인한 성능 간섭을 크게 최소화하였다.

1. 서론

클라우드 환경에서 컨테이너 기반 가상화는 낮은 가상화 오버헤드와 높은 효율성으로 인해 하이퍼바이저 기반 가상화의 대안으로 사용된다. 이러한 컨테이너 기반 가상화 도구의 대표적인 예시로 Docker [1]는 적은 자원을 가지고 빠르고 유연하게 확장할 수 있어 주요 클라우드 플랫폼에서 널리 채택된다. 이렇게 가상화를 사용하는 데이터센터에서는 수많은 클라이언트의 요청을 서버에 배치해서 처리하게 되는데, 이를 위한 컨테이너의 라이프 사이클 관리가 필수적이다 [2].

컨테이너는 라이프 사이클 관리의 핵심 메커니즘으로 C/R (Checkpoint/Restore)가 있다. C/R은 실행 중인 프로세스를 중단하여 현재 상태를 디스크에 이미지 파일로 쓰고, 이후 저장된 파일을 위에서 중단된 상태로부터 프로세스를 복원한다. 현재 Docker는 리눅스 소프트웨어인 CRUI (Checkpoint/Restore In Userspace) [3]를 사용하여 C/R을 수행한다.

컨테이너는 호스트 운영체제의 프로세스로 실행 중인 다른 컨테이너와 호스트의 물리적 자원을 나눠 가진다 [4]. 따라서 각 컨테이너가 디스크의 자원에 동시에 접근하면 자원에 대한 경쟁으로 성능 간섭이 발생할 수 있다. 또한 컨테이너의 I/O가 수행되는 전통적인 SSD는 데이터들을 쓸 때 여러 채널을 동시에 접근하

* 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획재정부의 지원을 받아 수행된 연구임 (No.2021-0-01475, (SW-스타랩) 비경쟁 빅데이터를 위한 새로운 기-클라우드 DB 개발)와 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 중견연구자지원사업의 지원을 받아 수행된 연구임 (No.2022R1A2C1A01006050)

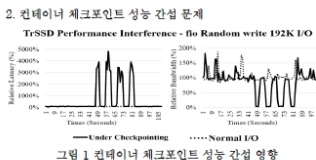


그림 1 컨테이너 체크포인트 성능 간섭 영향

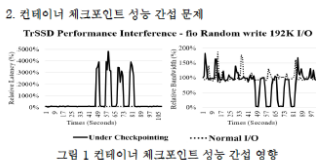


그림 1 컨테이너 체크포인트 성능 간섭 영향

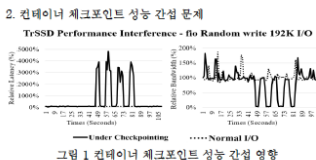


그림 1 컨테이너 체크포인트 성능 간섭 영향

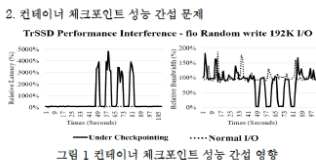


그림 1 컨테이너 체크포인트 성능 간섭 영향

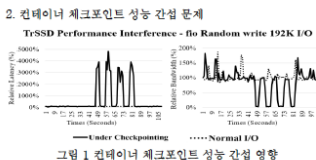


그림 1 컨테이너 체크포인트 성능 간섭 영향

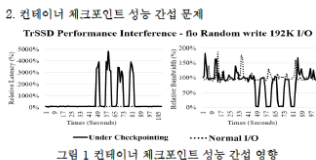


그림 1 컨테이너 체크포인트 성능 간섭 영향

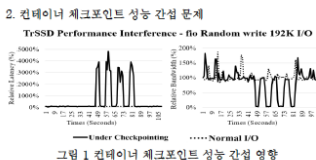


그림 1 컨테이너 체크포인트 성능 간섭 영향

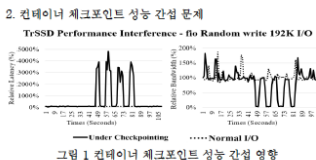


그림 1 컨테이너 체크포인트 성능 간섭 영향

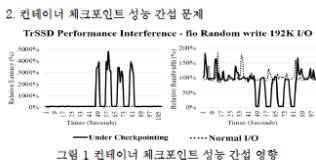


그림 1 컨테이너 체크포인트 성능 간섭 영향

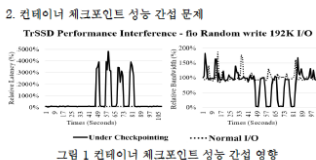


그림 1 컨테이너 체크포인트 성능 간섭 영향

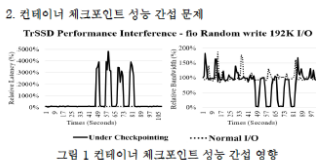


그림 1 컨테이너 체크포인트 성능 간섭 영향

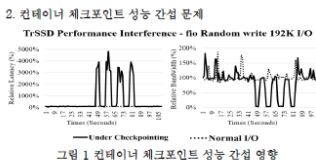


그림 1 컨테이너 체크포인트 성능 간섭 영향

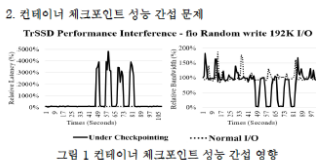


그림 1 컨테이너 체크포인트 성능 간섭 영향

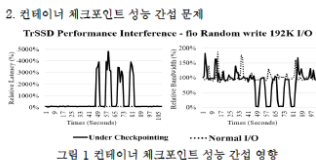


그림 1 컨테이너 체크포인트 성능 간섭 영향

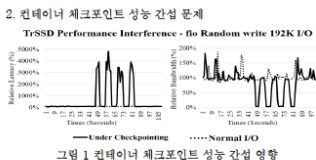


그림 1 컨테이너 체크포인트 성능 간섭 영향

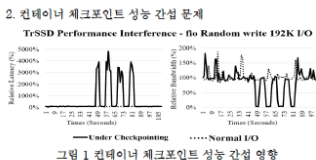


그림 1 컨테이너 체크포인트 성능 간섭 영향

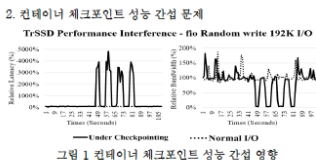


그림 1 컨테이너 체크포인트 성능 간섭 영향

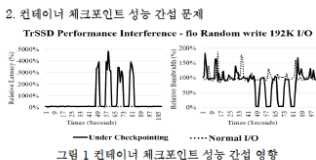


그림 1 컨테이너 체크포인트 성능 간섭 영향

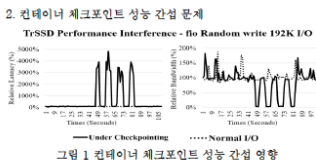


그림 1 컨테이너 체크포인트 성능 간섭 영향

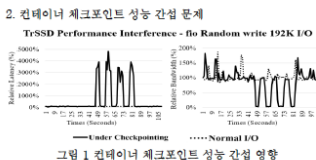


그림 1 컨테이너 체크포인트 성능 간섭 영향

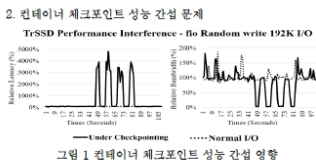


그림 1 컨테이너 체크포인트 성능 간섭 영향

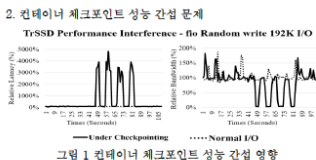


그림 1 컨테이너 체크포인트 성능 간섭 영향

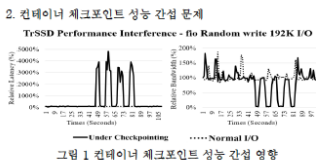


그림 1 컨테이너 체크포인트 성능 간섭 영향

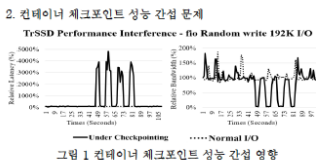


그림 1 컨테이너 체크포인트 성능 간섭 영향

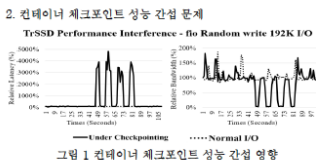


그림 1 컨테이너 체크포인트 성능 간섭 영향

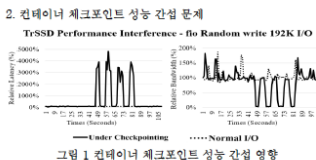


그림 1 컨테이너 체크포인트 성능 간섭 영향

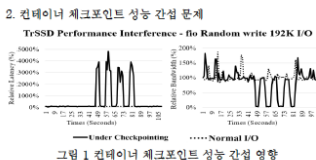


그림 1 컨테이너 체크포인트 성능 간섭 영향

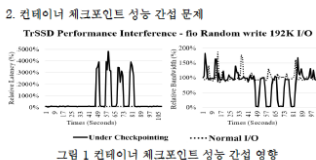


그림 1 컨테이너 체크포인트 성능 간섭 영향

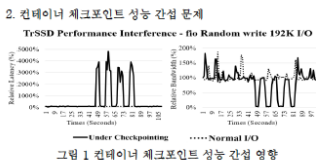


그림 1 컨테이너 체크포인트 성능 간섭 영향

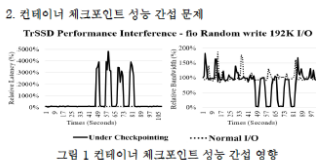


그림 1 컨테이너 체크포인트 성능 간섭 영향

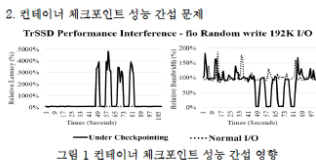


그림 1 컨테이너 체크포인트 성능 간섭 영향

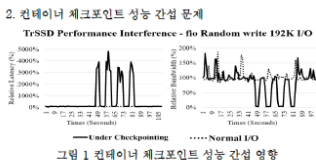


그림 1 컨테이너 체크포인트 성능 간섭 영향

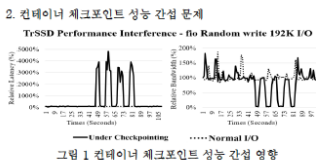


그림 1 컨테이너 체크포인트 성능 간섭 영향

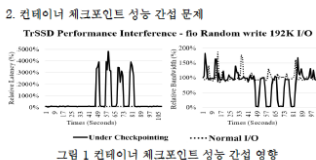


그림 1 컨테이너 체크포인트 성능 간섭 영향

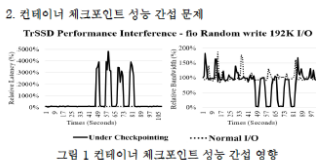


그림 1 컨테이너 체크포인트 성능 간섭 영향

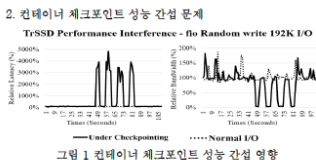


그림 1 컨테이너 체크포인트 성능 간섭 영향

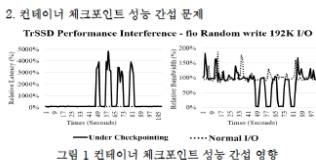


그림 1 컨테이너 체크포인트 성능 간섭 영향

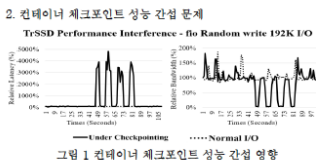


그림 1 컨테이너 체크포인트 성능 간섭 영향

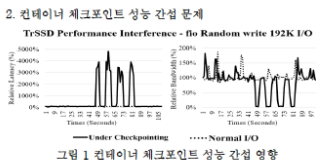


그림 1 컨테이너 체크포인트 성능 간섭 영향

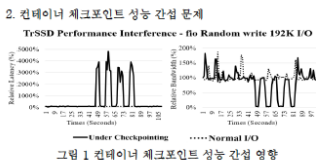


그림 1 컨테이너 체크포인트 성능 간섭 영향

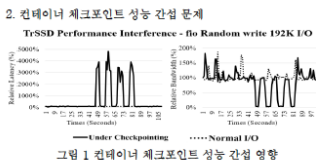


그림 1 컨테이너 체크포인트 성능 간섭 영향

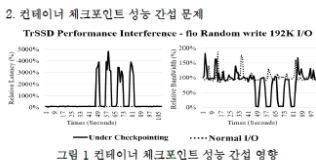


그림 1 컨테이너 체크포인트 성능 간섭 영향

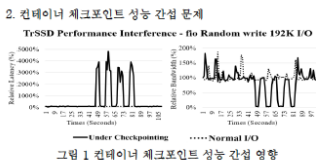


그림 1 컨테이너 체크포인트 성능 간섭 영향

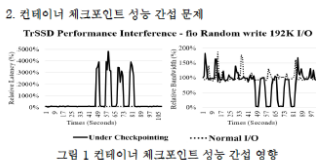


그림 1 컨테이너 체크포인트 성능 간섭 영향

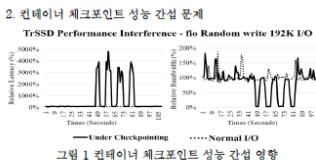


그림 1 컨테이너 체크포인트 성능 간섭 영향

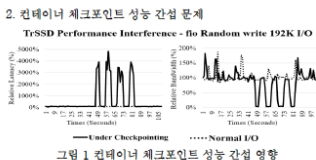


그림 1 컨테이너 체크포인트 성능 간섭 영향

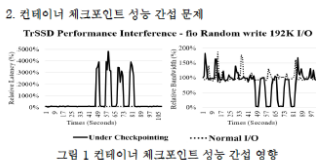


그림 1 컨테이너 체크포인트 성능 간섭 영향

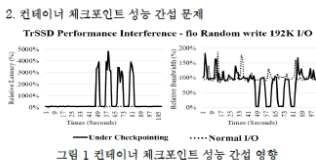


그림 1 컨테이너 체크포인트 성능 간섭 영향

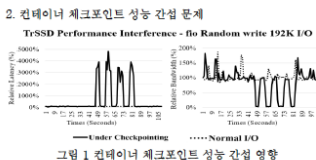


그림 1 컨테이너 체크포인트 성능 간섭 영향

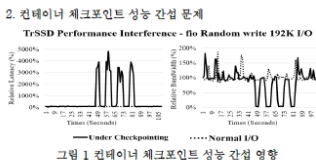


그림 1 컨테이너 체크포인트 성능 간섭 영향

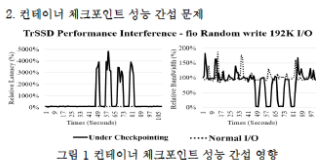


그림 1 컨테이너 체크포인트 성능 간섭 영향

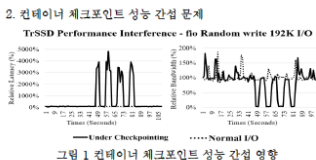
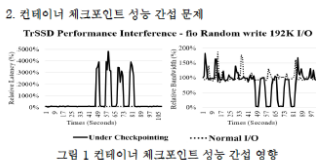


그림 1 컨테이너 체크포인트 성능 간섭 영향



Thank You !

2022. 06. 29

Presented by Myunghoon Oh

snt2426@dankook.ac.kr