ZNS SSD를 활용한 컨테이너 체크포인트 성능 간섭 최소화

한예진O, 오명훈, 최종무
단국대학교
{hyj0225, snt2426, choijm}@dankook.ac.kr

Minimizing Container Checkpointing Performance Interference using ZNS SSD

Yejin HanO, Myunghoon Oh, Jongmoo Choi
Dankook University

요 약

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 경합으로 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. 서 론

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].

컨테이너 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 읽어서 중단된 상태로부터 프로세스를 복원한다. 현재 Docker는 리눅스 소프트웨어인 CRIU (Checkpoint/Restore In Userspace) [3]를 사용하여 C/R를 수행한다.

컨테이너는 호스트 운영체제의 프로세스로 실행 중인 다른 컨테이너와 호스트의 물리적 자원을 나눠 가진다[4]. 따라서 각 컨테이너가 디스크의 파일에 동시에 접근하면 자원에 대한 경쟁으로 성능 간섭이 발생할 수 있다. 또한 컨테이너의 I/O가 수행되는 전통적인 SSD는 데이터를 쓸 때 여러 채널을 동시에 접근하여 파일을 쓰기 때문에, 서로 다른 I/O 작업들 간의 간섭이 발생할 수 있다. 결국 이러한 성능 간섭으로 인해 클라우드 서비스 사용자는 자신의 요청이 예상한 시간내로 처리되지 않아 QoS (Quality of Service)가 보장되지 못하는 문제점이 있다.

따라서 본 논문에서는 호스트가 수행하고 있는 I/O 작업에 컨테이너 체크포인트가 미치는 영향을 분석한다. 나아가 체크포인트로 인한 성능 간섭 문제를 해결하기 위해 차세대 스토리지인 ZNS(Zoned Namespace) SSD[5]를 새롭게 도입하여 컨테이너 성능을 고립하는 방법을 제안한다. ZNS SSD를 이용하여 제안한 기법의 성능을 평가한 결과 기존 SSD에 비해 체크포인트 성능 간섭 영향을 크게 줄였다.

본 논문은 5장으로 구성된다. 2장에서는 전통적인 SSD에서 체크포인트를 수행했을 때 관찰한 성능 간섭을 보이고 3장에서는 ZNS SSD를 활용하여 체크포인트를 고립하는 기법에 대한 설계 및 구현을 설명한다. 4장에서는 제안된 기법을 실험하여 측정한 체크포인트 성능 고립 결과에 대해서 논의하고, 5장에서는 결론을 맺는다.

2. 컨테이너 체크포인트 성능 간섭 문제

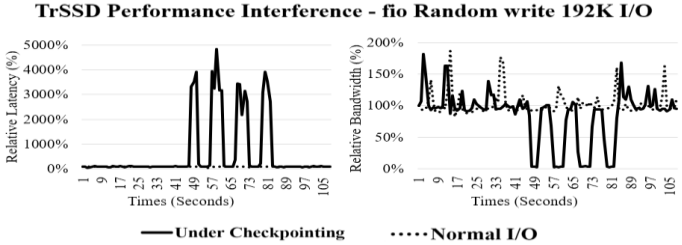

그림 1 컨테이너 체크포인트 성능 간섭 영향

* 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-01475, (SW 스타랩) 비정형 빅데이터를 위한 새로운 키-밸류 DB 개발)와 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 중견연구자지원사업의 지원을 받아 수행된 연구임(No.2022R1A2C1006050)

현재 물리 머신에서 동작하고 있는 Docker 컨테이너를 체크포인트하면 현 시점의 프로세스 상태를 Dump한다[6]. 이때 체크포인트 작업으로 인해 기존에 I/O를 수행하던 작업에 간섭이 발생할 수 있다. 그림 1은 fio benchmark를 이용하여 SATA 인터페이스를 사용하는 전통적인 SSD (Traditional SSD, 이하 TrSSD)에 I/O 작업을 계속해서 수행하고 있을 때 Cassandra 애플리케이션이 수행되는 컨테이너를 체크포인트 하였을 때 발생한 성능 간섭을 보여준다. Normal I/O는 fio 단일 랜덤 쓰기를 나타내고 Under Checkpointing은 fio가 랜덤 쓰기로 I/O를 진행중일 때 컨테이너 체크포인트가 수행된 경우이다. Fio가 TrSSD에 쓰기를 진행하다가 컨테이너 체크포인트가 수행되는 구간에는 순간적으로 Latency가 50배까지 증가하고 Throughput은 40배 정도 크게 감소한다. 결국 컨테이너 체크포인트로 인해 사용자의 디스크 작업이 간섭을 크게 받아 QoS가 일정하게 보장되지 못한다는 것을 알 수 있다.

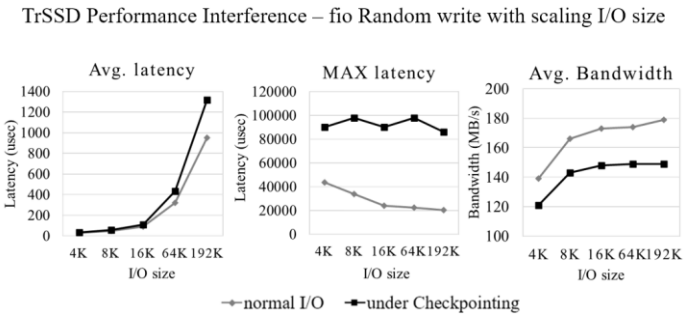

그림 2 I/O 크기 변화에 따른 체크포인트 성능 간섭

이어서 컨테이너 체크포인트로 인한 성능 간섭 영향을 자세히 관찰하기 위해 여러 변수를 조절해가면서 실험을 진행하였다. 그림 2는 I/O 크기를 4K에서 192K까지 증가시키면서 그림 1과 같이 쓰기를 수행하였다. 기존의 체크포인트가 수행되지 않을 때 normal I/O 성능과 비교하면 평균 Latency와 최대 Latency 모두 증가하였고, 평균 Bandwidth는 감소하였다.

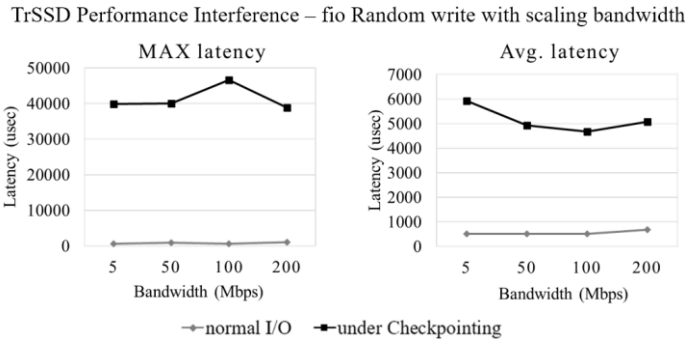

그림 3 bandwidth 변화에 따른 체크포인트 성능 간섭

그림 3은 Bandwidth를 5Mbps, 50Mbps, 100Mbps, 200Mbps로 증가시키면서 동일하게 체크포인트 성능 간섭 실험을 수행한 결과이다. 최대 Latency와 시간과 평균 Latency 모두 체크포인트를 수행하지 않았을 경우에 비해 큰 폭으로 증가한 것을 관찰할 수 있다.

그림 4는 동시에 체크포인트가 수행되는 컨테이너 수를 0개(체크포인트가 수행되지 않는 normal I/O)부터 하나씩 차례로 3개까지 증가시키면서 쓰기 성능 변화를 관찰하였다. 실험 결과를 통해 동시에 체크포인트를 많이 수행할수록 컨테이너 체크포인트로 인한 성능 간섭을 더욱 크게 받는 다는 것을 확인하였다.

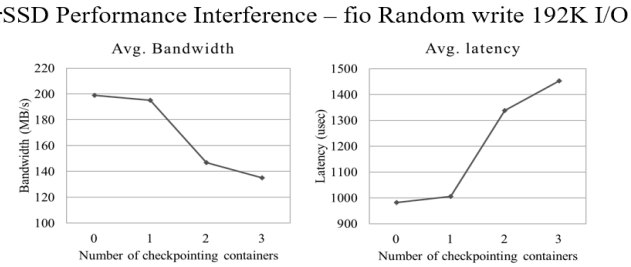

그림 4 동시에 수행되는 컨테이너 개수에 따른 체크포인트 성능 간섭

컨테이너 체크포인트로 인한 성능 간섭 문제는 하나의 물리 머신에서 동작하고 있는 컨테이너들이 호스트 커널 OS를 공유하고 있어 자원의 경쟁이 발생하기 때문이다[4]. 또한 쓰기가 진행되는 전통적인 SSD는 여러 애플리케이션의 데이터를 저장할 때 SSD 내에서 영역을 구분하지 않고 같은 채널에 동시에 접근하여 저장하기 때문에 쓰기 시 간섭의 영향을 크게 받는다.

3. ZNS SSD를 활용한 컨테이너 체크포인트 고립 제안

컨테이너 체크포인트로 인한 성능 간섭 문제를 개선하기 위한 방안으로 기존 SSD를 대체하는 차세대 스토리지인 ZNS SSD[5]를 도입한다.

3.1 ZNS (Zoned Namespace) SSD

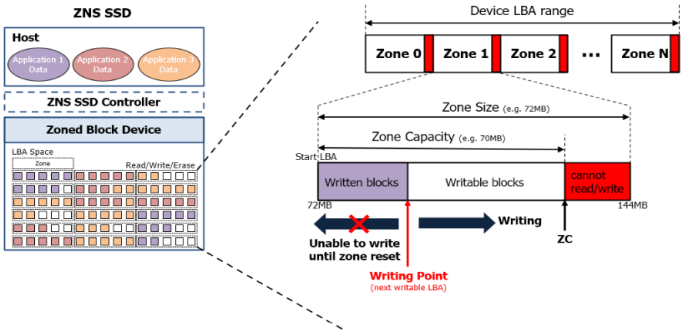

그림 5 ZNS SSD의 구조

ZNS SSD는 전통적인 SSD와 달리, 디바이스의 전체 LBA (Logical block Address)를 독립적인 Zone으로 나누어서 관리한다. Zone 내에서는 순차적으로 써야 하는 특징이 있으며, 쓰기가 진행되면 Zone의 WP(Write Pointer)가 증가하면서 다음 LBA를 가리킨다.

ZNS SSD는 호스트 단에서 각 애플리케이션 데이터의 Hot/Cold 특성과 용도에 따라 디바이스의 Zone을 지정하여 데이터를 배치할 수 있다. 즉 기존 SSD과 달리 워크로드 별 고립이 가능하다. 따라서 ZNS SSD의 Zone과 Channel간의 연결 관계

를 고려하여 컨테이너 체크포인트 성능의 고립이 가능하다.

3.2 체크포인트 고립 설계 및 구현

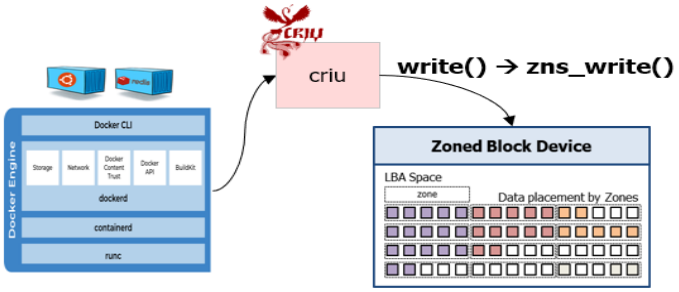

그림 6 ZNS SSD 특성을 이용한 도커 체크포인트 고립

Docker는 컨테이너 체크포인트 시 CRIU에서 POSIX API를 통해 디바이스에 이미지 파일을 Dump한다. 이미지 파일을 기존 SSD에서 ZNS SSD에 쓰기 위해서는 그림 6과 같이 POSIX write()를 ZNS controller를 통해 Zone에 접근할 수 있도록 수정한다.

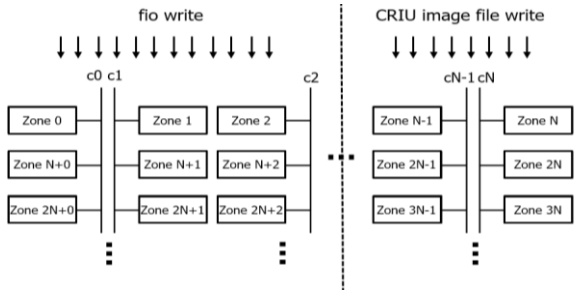

그림 7 이미지 파일과 normal 작업의 채널 분리

그림 7에서 SSD의 내부 구조를 보면 채널 별로 Zone이 그룹화 되어있다. 전통적인 SSD는 디바이스 내의 채널을 지정하여 데이터를 저장할 수 없는데 반해, ZNS SSD는 호스트단에서 Zone을 인지하여 데이터를 구분하여 저장할 수 있다. 이 특성을 이용하여 데이터 쓰기 시 채널 간 간섭을 최소화하도록 하기 위해 각 워크로드별로 채널을 분리한다. 즉, 체크포인트로 인한 Dump 이미지 파일은 Zone N, Zone 2N, Zone 3N, … 에 쓰도록 하고 일반적인 디스크 I/O는 체크포인트가 수행되지 않는 다른 채널의 Zone에 나누어 쓰도록 하여 간섭을 최소화하게 구현하였다.

4. ZNS SSD 체크포인트 성능 간섭 실험 결과

표 1 실험 환경

CPU	Intel(R) Core(TM) i5-4440, 3.10GHz
Memory	32GB
Storage	ZNS SSD prototype 2TB
Kernel	Linux 5.7.7
Tool	CRIU v3.16.1

본 논문은 표 1의 실험 환경을 바탕으로 ZNS SSD에서 수행한 컨테이너 체크포인트 성능 간섭을 평가하였다. 백업 이미지가 저장되는 ZNS SSD는 프로토타입 버전을 사용하였다.

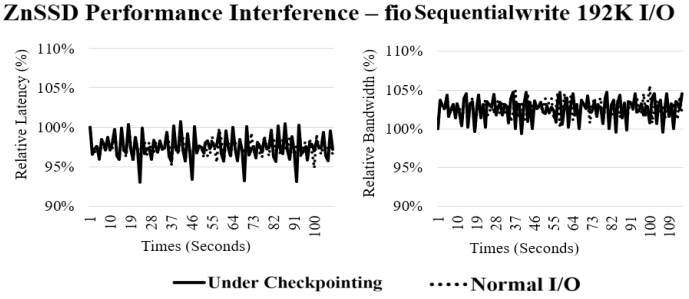

그림 8 ZNS SSD에서 컨테이너 체크포인트 성능 간섭 최소화

그림 8은 ZNS SSD에서 fio 순차 쓰기를 수행하면서 컨테이너 체크포인트에 따른 영향을 평가한 그래프이다. Normal I/O는 fio 순차 쓰기를 나타내고 Under Checkpointing은 fio의 순차 쓰기 도중 컨테이너 체크포인트가 수행된 경우이다. 그림 1의 기존 SSD에서의 실험 결과와는 달리 ZNS SSD에서 쓰기 Latency는 Normal I/O와 Under Checkpointing의 양상이 비슷하고 크게 증가하지 않는다. 또한 Bandwidth 역시 normal I/O와 유사하게 크게 감소하는 부분이 존재하지 않는다. 따라서 ZNS SSD는 컨테이너 체크포인트 시 기존의 SSD에서 발생하던 체크포인트 성능 간섭을 최소화하고 Bandwidth를 일정하게 유지하여 사용자의 QoS (Quality of Service)를 제공할 수 있었다.

5. 결 론

본 논문은 컨테이너 체크포인트로 인한 성능 간섭을 분석하고 그에 대한 해결방안으로 차세대 스토리지인 ZNS SSD를 활용하였다. 전통적인 SSD에서는 데이터의 특성을 고려하지 않고 애플리케이션 데이터를 임의로 저장하기 때문에 컨테이너 체크포인트 성능 간섭으로 인한 영향을 크게 받았다. 그러나 ZNS SSD를 도입하여 워크로드 별 데이터를 채널 별로 고립하여 성능 간섭을 최소화할 수 있었다.

참고 문헌

[1] C. Anderson, “Docker.” IEEE Software, vol. 32, no. 3, 2015.

[2] Google: ‘EVERYTHING at Google runs in a container’, [Online]. Available: https://www.theregister.co.uk/2014/05/23/google_containerization_two_billion/

[3] CRIU. Checkpoint/Restore In Userspace, [Online]. Available: https://criu.org/Main_Page.

[4] Giorgos Kappes, Stergios V. Anastasiadis, “Experience Paper: Danaus: Isolation and Efficiency of Container I/O at the Client Side of Network Storage”, ACM Middleware, 132-145, 2021

[5] NVMe Zoned Namespaces (ZNS) SSDs, [Online]. Available: https://Zonedstorage.io/docs/introduction/zns

[6] 한예진, 최종무, “CRIU를 이용한 Docker Container 체크포인트 성능 평가”, KSC, 2021