최근 메모리 스왑 성능 최적화 관련 연구 동향

최문형, 손용석
중앙대학교
zhemfpdlf@cau.ac.kr, sysganda@cau.ac.kr

A Study on the Memory Swap Performance Optimization

Munhyung Choi, Yongseok Son
Chung-Ang University

요 약

본 논문은 최근 메모리 스왑 성능 최적화 관련 연구를 파악한다. 메모리 스왑은 메모리가 부족할 경우 자주 사용되지 않는 익명의 페이지를 스토리지로 저장해둠으로써 메모리 공간을 확보하고 관리한다. 최근 빅데이터 및 대용량 연산이 증가하면서 메모리 부족 현상이 자주 발생되며 이는 스왑 성능이 전체 시스템의 성능에 큰 영향을 미치고 있다. 이에 따라 본 논문에서는 스왑 성능 최적화 연구들을 파악한다. 예를들어, ZNSwap, VSWAPPER, NVM-Swap, DeepSwapper, 임베디드 환경에서의 스왑 기법을 분석하였다. ZNSwap은 ZNS SSD에 최적화된 스왑 서브 시스템으로 스왑 사용량 통계를 직접 액세스함으로써 성능 격리를 개선하였다. VSWAPPER는 가상화 환경에서의 uncooperative swapping 문제를 해결하여 성능을 향상시켰다. NVM-Swap은 비휘발성 메모리를 활용하여 스마트폰의 성능을 향상시켰으며, DeepSwapper는 딥러닝을 이용해 하이브리드 메모리 시스템에서의 페이지 스와핑을 최적화하였다. 마지막으로 임베디드 환경에서는 zswap의 문제점인 중복 페이지 식별을 통해 메모리 사용 효율을 극대화하였다.

이 연구들은 각각의 환경과 목적에 맞추어 설계되어 메모리 스왑 성능을 크게 향상시켰다. 앞으로의 연구에서는 이러한 기술들을 종합적으로 분석하고, 각 기술의 장점을 결합하여 보다 효율적인 메모리 스왑 최적화 방법을 모색하는 것이 필요하다. 새로운 메모리 기술의 발전과 함께 스왑 성능 최적화를 위한 연구는 더욱 활발히 이루어질 것으로 기대된다.

Ⅰ. 최근 메모리 스왑 성능 최적화 연구 동향

메모리 스왑은 시스템의 물리적 메모리가 부족할 때, 데이터를 디스크로 이동시키는 중요한 기법이다. 하지만 디스크의 접근 속도는 메모리보다 훨씬 느리기 때문에 스왑 작업은 일반적으로 성능 저하를 초래한다. 최근 다양한 연구들이 이러한 성능 문제를 해결하기 위한 새로운 접근법을 제시하고 있다. 본 논문에서는 최근 메모리 스왑 성능 최적화 관련 연구들을 분석하고, 각 기법이 어떠한 방식으로 성능 향상을 시키는지 살펴본다.

Ⅰ-1. ZNSwap

ZNSwap [1]은 ZNS (Zoned Namespace) SSD에 최적화된 스왑 서브 시스템이다. ZNSwap은 ZNS의 데이터 관리 제어를 활용하고, 효율적인 스왑 저장 공간을 위해 ZNGC (ZNS Garbage Collector)를 도입하였다. ZNSwap의 ZNGC는 linux kernel 내의 스왑 사용량 통계를 직접 액세스하여 세밀한 스왑 저장장치 관리와 GC 대역폭 사용을 정확히 계측함으로써 성능 격리를 개선하였다. 이를 통해 ZNS swap은 다양한 메모리 액세스 패턴에서도 안정적인 처리량을 유지하며, 실제 사용 시나리오에서 기존 SSD의 Linux 스왑에 비해 상당한 성능 이점을 보여주었다.

Ⅰ-2. VSWAPPER

VSWAPPER [2]는 가상화 환경에서 발생하는 uncooperative swapping 문제를 개선한 기법이다. Uncooperative swapping은 호스트 OS가 게스트 OS의 협력 없이 메모리 페이지를 디스크로 스왑하는 상황을 말한다. 이로 인해 게스트 OS가 이미 호스트에 의해 스왑된 페이지를 회수하려고 할 때, 스왑된 페이지가 호스트 스왑 영역에서 메모리로 다시 불러와진 후, 게스트 스왑 영역에 기록되는 불필요한 I/O가 발생하게 되어 성능 저하를 일으킨다. VSWAPPER는 Uncooperative swapping 문제를 해결하기 위해 Swap mapper와 False reads preventer를 도입했다. Swap mapper는 게스트의 디스크 I/O 작업을 모니터링하고, 게스트 메모리 페이지와 해당 디스크 블록 간의 매핑을 유지한다. 이를 통해 호스트 OS가 게스트의 페이지를 스왑 아웃시켜도 매핑 관계를 유지하기 때문에 불필요한 I/O가 발생하지 않는다. False reads preventer는 게스트가 스왑 아웃된 페이지로 쓰기 (write) 명령을 시도할 때, 이를 에뮬레이트하여 임시 버퍼에 데이터를 저장하고, 페이지 전체가 다시 쓰여질 때까지 디스크 읽기 작업을 지연시킨다. 이러한 방식으로 VSWAPPER는 기존 uncooperative swapping과 비교했을 때 최대 10배, 최소 1.035배의 성능 향상을 보여주었다.

Ⅰ-3. NVM-Swap

NVM-Swap [3]은 스마트폰의 성능을 향상시키기 위해 비휘발성 메모리 (NVM)를 스왑 영역으로 활용한다. 기존의 NAND 플래시 기반 스토리지는 느리기 때문에 스왑 기능을 활성화하면 성능 저하가 발생하게 된다. 그러나 NVM은 높은 성능과 낮은 에너지 소비를 특징으로 하기 때문에 성능 향상을 기대할 수 있다. NVM-Swap은 비활성 메모리를 효과적으로 활

용하기 위해 Copy-on-Write Swap-In (COWS), Heap-Wear 기법을 제공한다. COWS 기법은 메모리 복사 작업을 최소화하기 위해 도입된 기법으로 주로 읽기 전용 요청에 대해 불필요한 메모리 복사를 방지하여 성능을 향상시키는 데 목적을 둔다. 응용 프로그램이 스왑된 페이지에 읽기 (read) 요청할 때, 해당 페이지를 DRAM으로 복사하지 않고 NVM에서 직접 읽을 수 있도록 한다. 쓰기 (write) 요청 시에는 NVM이 DRAM에 비해 쓰기 속도가 느리다는 점을 고려하여 DRAM으로 메모리를 복사한 후에 쓰기 작업을 진행한다. Heap-Wear 기법은 NVM의 내구성을 향상시키기 위해 설계된 wear-leveling 알고리즘으로, 슬롯 간의 쓰기 분포를 균등하게 하여 특정 슬롯에 쓰기가 집중되는 문제를 해결하였다. 이러한 방식으로 NVM-Swap은 메모리 복사 작업을 최소 40%에서 최대 75%까지 감소시켰고, 애플리케이션 실행 시간은 평균 20% 이상 단축시켰다.

Ⅰ-4. DeepSwapper

DeepSwapper [4]는 하이브리드 메모리 시스템에서 DRAM과 NVM 간의 페이지 스와핑을 최적화하기 위해 설계된 딥러닝 기반 페이지 스와핑 관리 기법이다. DeepSwapper는 LSTM을 활용하여 미래의 메모리 접근 패턴을 예측하고, 이를 바탕으로 효율적인 페이지 스와핑을 진행한다. 또한, ReRAM과 같은 NVM의 온도에 민감한 특성을 고려하여 NVM의 온도를 실시간으로 모니터링한다. 만약 NVM의 특정 영역이 고온 상태일 경우, 해당 영역의 자주 쓰이는 페이지를 DRAM 영역으로 이동시킴으로써 온도 관리를 진행한다. 이를 통해 DeepSwapper는 PoM [5] 및 MemPod [6] 대비 최대 30.1% 성능 향상과 최대 1.87배 수명 연장을 보여주었다.

Ⅰ-5. 임베디드 환경에서의 swap

임베디드 환경에서 물리적 메모리는 한정된 중요한 자원이기 때문에 공간을 효율적으로 사용해야 한다. linux kernel의 zswap은 RAM을 더 효율적으로 사용할 수 있도록 페이지를 압축하여 스왑하는 메모리 압축 기반 스왑 모듈이다. 그러나 기존의 zswap은 압축한 페이지에 대해 중복 페이지를 식별하지 않아 불필요한 메모리 공간을 사용하게 된다. 해당 논문 [7]에서는 이 문제를 해결하기 위해 페이지의 체크섬을 계산하고, 동일한 내용을 가진 페이지가 발견되면 기존 압축된 페이지 데이터를 공유하는 방식을 제안했다. 이 방법을 통해 성능 페널티 없이 가용 메모리를 최소 10%에서 최대 12% 증가시켰다. 이는 메모리 사용량을 최적화하고, 메모리 압박으로 인한 스왑 횟수를 줄이며, 애플리케이션 실행 시간을 단축시켰다.

Ⅱ. 결론

본 논문에서는 최근 메모리 스왑 성능 최적화를 위한 다양한 연구들을 분석하여 각각의 기술이 어떠한 방식으로 성능 향상시키고 있는지 살펴보았다. ZNSwap은 ZNS SSD의 특성을 활용하여 성능 격리를 개선하였고, VSWAPPER는 가상화 환경에서의 uncooperative swapping 문제를 효과적으로 해결하였다. NVM-Swap은 스마트폰 환경에서 비휘발성 메모리를 활용하여 성능을 향상시켰으며, DeepSwapper는 딥러닝을 통해 하이브리드 메모리 시스템에서의 페이지 스와핑을 최적화하였다. 마지막으로 임베디드 환경에서는 zswap의 중복 페이지 식별을 통해 메모리 사용 효율을 극대화하였다.

이러한 다양한 접근법들은 각각의 환경과 목적에 맞추어 설계되었으며, 메모리 스왑 성능을 크게 향상시키는 데 기여하였다. 앞으로의 연구에서는 이러한 기술들을 종합적으로 분석하고, 각 기술의 장점을 결합하여 보다 효율적인 메모리 스왑 최적화 방법을 모색하는 것이 필요하다. 또한, 새로운 메모리 기술의 발전과 함께 스왑 성능 최적화를 위한 연구는 더욱 활발히 이루어질 것으로 기대된다.

ACKNOWLEDGMENT

이 성과는 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2022R1A4A5034130) (교신저자: 손용석).

참 고 문 헌

[1] Bergman, S., Cassel, N., Bjørling, M. and Silberstein, M., 2023. ZNSwap: Un-block your swap. ACM Transactions on Storage, 19(2), pp.1-25.

[2] Amit, N., Tsafrir, D., & Schuster, A. 2014. Vswapper: A memory swapper for virtualized environments. Acm Sigplan Notices, 49(4), 349-366.

[3] Zhong, K., Wang, T., Zhu, X., Long, L., Liu, D., Liu, W., ... & Sha, E. H. M. 2014. Building high-performance smartphones via non-volatile memory: The swap approach. In Proceedings of the 14th international conference on embedded software, pp.1-10.

[4] Valad Beigi, M., Pourshirazi, B., Memik, G., & Zhu, Z. 2020. DeepSwapper: A deep learning based page swap management scheme for hybrid memory systems. In Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques. pp.353-354.

[5] Sim, J., Alameldeen, A. R., Chishti, Z., Wilkerson, C., & Kim, H. 2014. Transparent hardware management of stacked dram as part of memory. In 2014 47th Annual IEEE/ACM International Symposium on Microarchitecture, pp.13-24.

[6] Prodromou, A., Meswani, M., Jayasena, N., Loh, G., & Tullsen, D. M. 2017. Mempod: A clustered architecture for efficient and scalable migration in flat address space multi-level memories. In 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp.433-444.

[7] Desireddy, S., & Pathireddy, D. R. 2016. Optimize In-kernel swap memory by avoiding duplicate swap out pages. In 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), pp.1-4.