aliusM: 누마환경에서 영구메모리를 활용한 대규모 시스템 메모리로 확장

aliusM: Extension to large-scale system memory using persistent memory in a NUMA system

요 약

인텔의 옵테인 데이터센터 영구 메모리 모듈(data-center persistent memory module, DCPMM)을 메모리 모드로 설정할 경우 시스템 메모리로 사용할 수 있다. 그러나 DRAM보다 낮은 성능 특성으로 인하여 메모리 모드보다는 앱 다이렉트 모드로 더 많이 사용되고 있다. 누마(non-uniform memory access, NUMA) 구조에서는 DCPMM에 접근하는 방식에 따라 다양한 성능 특성을 보인다. 본 연구에서는 이러한 특성을 기반으로 DCPMM을 시스템 메모리로 사용할 수 있는 aliusM을 설계하고 구현하였다. aliusM에 기반하여 DCPMM을 사용할 경우 원격 DRAM에 접근할 때와 유사한 성능을 얻을 수 있음을 보여준다.

1. 서 론

시스템 메모리는 고성능 컴퓨팅 환경에서 성능에 영향을 주는 주요 요인이다. 시스템 메모리로 DRAM을 많이 사용하고 있다. 인텔의 옵테인 DC 영구 메모리 모듈(data-center persistent memory module, DCPMM)도 시스템 메모리로 사용할 수 있으며 메모리 모드로 사용하는 방법[1]과 앱 다이렉트 모드에서 KMEM DAX 드라이버를 사용하는 방법[2] 등이 지원된다. DCPMM은 표 1에서와 같이 DRAM보다 낮은 성능 특성을 갖고 있다.

표 1 DIMM당 DCPMM과 DRAM과의 대역폭 [1]

DIMM당 대역폭	DCPMM	DRAM
순차읽기	~7.6GB/s	~15GB/s
무작위읽기	~2.4GB/s	~15GB/s
순차쓰기	~2.3GB/s	~15GB/s
무작위쓰기	~0.5GB/s	~15GB/s

메모리 모드에서 DRAM은 DCPMM의 캐시(memory-side cache)로 사용되며 커널에서 인식되지 않는다. DRAM은 그림 1에서와 같이 DCPMM에 대하여 직접사상 방식의 캐시(direct-mapped cache)로 사용되며 전체 시스템 메모리의 크기는 DCPMM의 크기가 된다. 직접사상방식의 캐시는 구현이 쉽고 빠른 처리가 가능하다. 메모리 모드에서 가상머신을 이용한 연구[3]에서 2개씩 가상 머신을 짝을 지어 동시에 동작하였을 때 성능이 떨어지는 가상 머신 짝이 있음이 관찰되었다. 동일한 캐시공간(메모리쪽 캐시의 녹색부분)을 사용하는 가상머신(시스템 메모리쪽의 녹색부분)이 동시에 동작하면서 캐시 미스가 증가하면서 성능 저하가 발생하였으며 직접사상 방식 캐시를 사용할 때 존재하는 단점이다.

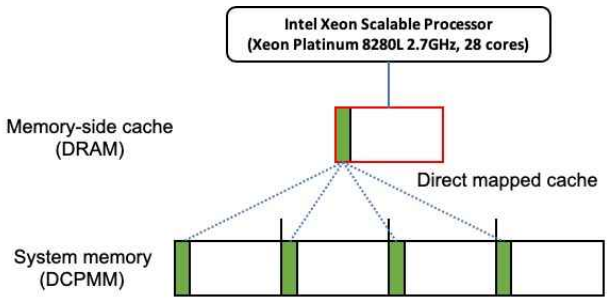

그림 1 DCPMM을 시스템 메모리로 사용할 경우 DRAM과의 관계. DRAM은 DCPMM의 캐시로 사용되고 시스템 메모리의 크기는 DCPMM의 크기와 같다.

KMEM DAX 드라이버를 사용할 경우 DCPMM은 메모리만 있는 누마(non-uniform memory access, NUMA) 노드로 인식된다. CPU에서 접근할 수 있는 DCPMM 정보가 없기 때문에 시스템 메모리로 사용하기 위해서는 사용자의 개입이 필요한 문제가 있다.

많은 시스템 메모리를 사용할수록 높은 성능을 얻을 수 있으나 시스템 메모리의 증설은 하드웨어 구조에 제한을 받는다. 하나의 슬롯에 DRAM은 256GB 까지 설치할 수 있지만 DCPMM은 512GB까지 설치하여 사용할 수 있어 시스템 메모리 증설에 유리하다. DCPMM을 시스템 메모리로 사용할 때 성능 특성 분석 연구[4]에 따르면 특정 슬롯에 설치된 DCPMM들을 인터리빙(interleaving) 방식

으로 사용하면 원격의 DRAM을 사용할 때와 비슷한 성능을 얻을 수 있다. 본 연구에서는 DCPMM 크기만을 시스템 메모리로 사용할 수 있는 메모리 모드의 단점과, 사용자의 개입이 필요한 KMEM DAX의 단점을 보완하기 위하여 새로운 시스템 메모리 사용 기법인 aliusM을 설계하였다. aliusM은 DRAM과 DCPMM이 설치된 누마 시스템에서 DRAM과 DCPMM을 모두 시스템 메모리로 사용할 수 있다. 그리고 사용자의 개입 없이 DCPMM을 시스템 메모리로 사용할 수 있으며 원격의 DRAM을 사용할 때와 비슷한 성능으로 사용할 수 있도록 설계하였다. vmalloc을 사용하여 메모리 할당 소요시간으로 성능 측정을 수행하였을 때 할당 크기가 클수록 aliusM이 메모리모드 보다 높은 성능을 보여준다.

2. 시스템 메모리와 aliusM

2.1 리눅스의 시스템 메모리

누마(non-uniform memory access, NUMA) 구조에서 리눅스는 높은 성능을 얻기 위해 CPU로 부터 접근 지연(access latency)이 작은 위치의 메모리를 먼저 사용한다. 인텔의 옵테인 DC 영구 메모리 모듈(data-center persistent memory module, DCPMM)의 성능 특성 연구들[4,5]에 따르면 DCPMM은 DRAM과 다른 성능 특성을 갖고 있으며 인터리빙(interleaving) 구성에 따라 다른 성능을 보여준다. DRAM을 시스템 메모리로 사용해온 리눅스는 DCPMM의 특성을 고려하지 않고 사용하도록 구현되어 있어 DCPMM을 시스템 메모리로 사용할 경우 높은 성능을 얻기 어렵다. 메모리 모드를 통해 DCPMM을 시스템 메모리로 사용할 경우 커널과 응용 등 소프트웨어를 수정할 필요가 없는 장점이 있다. 그러나 DRAM이 DCPMM의 캐시로 사용되기 때문에 DCPMM 크기만큼 시스템 메모리로 사용할 수 있고, DRAM이 직접사상 방식의 캐시(direct-mapped cache)로 동작하기 때문에 DRAM의 동일한 영역에 캐싱되는 메모리 영역의 사용이 빈번할 경우 성능이 저하되는 문제가 있다.

2.2 aliusM의 설계와 구현

aliusM은 DRAM과 DCPMM을 모두 시스템 메모리로 사용하면서 DCPMM을 사용자의 개입 없이 사용할 수 있도록 설계하였고 리눅스 커널 5.4 환경에서 구현하였다. DRAM을 시스템 메모리로 사용하기 위해서는 앱 다이렉트 모드로 부팅 하여야 한다. 부팅과정에서 전달받는 하드웨어 정보로 DRAM과 DCPMM을 구분하였으며 DCPMM 관리를 위해 memblock 구조체와 memblock_add_region 함수를 사용하였다. DCPMM영역에서 페이지를 할당할 수 있도록 alloc_pages_current 함수를 수정하였다. 그림 2는 aliusM의 시스템 메모리 개략도이다.

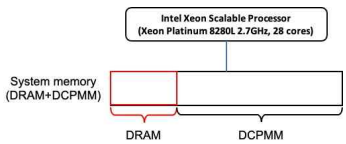

그림 2 aliusM 환경에서의 시스템 메모리

2.3 성능 평가

실험에 사용한 시스템은 그림 3과 같이 메모리가 구분되어있다. 하나의 CPU 소켓에는 2개의 메모리 컨트롤러(integrated memory controller, iMC)가 있는 CPU가 설치되어 있으며 2개의 누마 클러스터로 구분된다. 하나의 메모리 컨트롤러에 96GB의 DRAM과 768GB의 DCPMM이 연결되어 있으며 지역 노드가 된다.

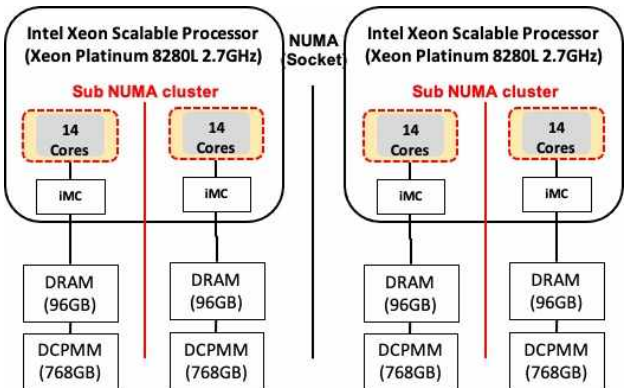

그림 3 테스트 시스템의 메모리 위치와 크기

aliusM의 성능을 평가하기 위하여 vmalloc을 사용하였다. vmalloc는 커널의 가상주소공간에서 연속적인 주소를 갖는 메모리를 할당 받는다. 사용 가능한 물리 메모리에 대해 할당을 요청할 수 있으며 파일 입출력을 필요로 하지 않는다. 실험은 노드별 메모리 크기를 고려하여 하나의 코어에서 할당 요청을 수행하였으며 실험 결과는 표 2와 같다.

표 2 메모리모드와 aliusM vmalloc 수행시간 결과(초)

할당 크기	메모리 모드	aliusM
80GB	64.74	19.12
700GB	581.29	258.19
1000GB	847.93	383.88
2000GB	1805.33	911.55
3000GB	실행불가	2005.21

메모리 모드에서 DRAM은 DCPMM의 캐시로 사용되고 캐시 적중률이 낮을수록 성능이 떨어지게 진다. vmalloc은 메모리 할당만을 하기 때문에 메모리 모드에서 DRAM보다 작은 크기의 메모리 할당을 요청하여도 캐시 사용으로 인한 이익은 없다. aliusM은 DRAM과 DCPMM을 모두 시스템 메모리로 사용하기 때문에 DRAM보다 작은 크기의 메모리를 사용할 경우에는 DRAM만을 시스템 메모리로 갖고 있는 시스템과 같은 성능을 보인다. 그리고 DCPMM을 사용하게 될 때에는 최고의 성능을 얻을 수 있도록 인터리빙하기 때문에 메모리 모드 보다 높은 성능을 얻을 수 있다. 3000GB 할당 요청의 경우 메모리 모드에서는 커널이 점유하고 있는 공간을 제외하고 남는 크기가 3000GB가 되지 않기 때문에 할당이 불가능하다. aliusM은 DRAM과 DCPMM을 합한 크기가 전체 시스템 메모리 크기가 되기 때문에 3000GB 할당이 가능하다. 2000GB에 비해 긴 시간이 소요된 이유는 원격 소켓에 있는 DCPMM 할당이 증가했기 때문이다.

3. 결론 및 향후 연구

고성능 컴퓨팅 시스템에서 대용량의 시스템 메모리 사용은 성능 개선에 유리하다. 인텔의 옵테인 DC 영구 메모리 모듈(data-center persistent memory module, DCPMM)는 DRAM보다 저렴하여 시스템 메모리로 사용할 경우 대용량의 시스템 메모리 사용이 용이한 장점이 있다. 그러나 DRAM보다 낮은 성능으로 인하여 사용이 활발하지 않다. 본 연구는 DRAM과 DCPMM을 시스템 메모리로 사용할 수 있는 환경에서 DCPMM의 성능을 최대한 활용하기 위한 기법 탐구를 위해 수행하였다. aliusM은 DRAM과 DCPMM을 모두 시스템 메모리로 사용하도록 지원하여, DRAM보다 작은 크기의 메모리 사용이 필요할 경우 DRAM만 설치된 시스템과 동일한 성능을 얻을 수 있으며 많은 메모리 사용이 필요할 경우 메모리 모드보다 높은 성능을 얻을 수 있음을 보였다. 빈번하게 사용되는 데이터와 사용 빈도가 많지 않은 데이터를 구분하여 할당하는 연구가 앞으로 수행되어야 할 과제이다.

참고문헌

[1] Intel Corporation, Intel® 64 and IA-32 Architectures Optimization Reference Manual, pp.345-349, May, 2020
[2] Memkind support for KMEM DAX option, https://pmem.io/2020/01/20/memkind-dax-kmem.html
[3] VMware, Inc, Intel Optane DC Persistent Memory “Memory Mode” Virtualized Performance Study, 2019, https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/IntelOptaneDC-PMEM-memory-mode-perf.pdf
[4] 이용섭, 정성인, 누마시스템에서 시스템 메모리로 사용하는 영구메모리모듈의 성능 특성 분석, 2020 한국소프트웨어종합학술대회 논문집(ISSN 2586-4599), pp.1-3, 2020,
[5] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson. 2019 b. An Empirical Guide to the Behavior and Use of Scalable Persistent Memory. arXiv preprint arXiv:1908.03583.