대용량 다중 코어 시스템을 위한 요구 병합 메모리 관리자

박희권○ 최종무
단국대학교 컴퓨터학
parkhk81@dankook.ac.kr, choijm@dankook.ac.kr

On-demand Coalescing Memory Manager for Huge Memory Multi-core Systems

Heekwon Park○ Jongmoo Choi
School of Computer Science and Engineering, Dankook University

최근 메모리 집적도의 향상과 높은 가격 경쟁력을 기반으로 시스템 메모리의 대용량화가 급속하게 진행되고 있으며 프로세서도 여러 개의 코어가 하나의 칩에 집적된 멀티 코어 프로세서의 사용이 일반화되어 가고 있다. 이러한 변화는 고성능 서버에서부터 개인용 컴퓨터, 노트북에 이르기까지 대부분의 컴퓨터 시스템에 영향을 미치고 있다. 한편 하드웨어의 발전과 더불어 이를 효과적으로 지원하기 위한 운영체제의 변화도 거듭되고 있다. 예를 들어 64비트 프로세서와 대용량 메모리를 지원하기 위한 가상 메모리 구조의 변화, CPU에서 프로세스들의 공평성을 지원하기 위한 CFS(Completely Fair Scheduler)의 도입 그리고 NUMA 구조를 효율적으로 지원하기 위한 부하균등 정책 등이 있다[1][2][3].

반면 대부분의 운영체제에서는 효과적인 메모리 관리를 위해 버디 메모리 할당자를 도입하고 있다. 버디 메모리 할당자는 다양한 크기의 요청을 효과적으로 처리하기 위하여 연속된 페이지들을 버디 그룹[1]으로 관리하는 정책을 사용한다[4]. 이러한 관리 정책은 특정 크기의 요청이 빈번한 시스템에서는 비효율적으로 동작하는 경향이 있다. 이를 위해 리눅스에서는 지연 버디 시스템을 도입하여 빈번히 발생하는 할당 크기에 대하여 효율적으로 처리하도록 설계되었다[3]. 구체적으로 지연 버디 시스템은 특정 개수의 페이지들을 지연 계층에서 처리함으로써 버디 그룹을 생성하기 위한 페이지들의 병합/분할 연산을 감소시킨다[5].

하지만 대용량 메모리를 사용하는 메모리 집중적인 응용의 경우 지연 버디 시스템에서 사용하는 기법은 비효율적으로 동작할 수 있다. 구체적으로 지연 버디 시스템의 최대값을 넘는 할당과 반납이 반복된다면 결국 대부분의 처리는 버디 계층에서 이루어지며 지연 계층에서의 관리는 오버헤드로 작용한다.

본 논문에서는 대용량 메모리를 사용하는 멀티 코어 프로세서에서 메모리 관리를 효율적으로 지원하기 위한 요구 병합 메모리 관리자를 제안한다. 제안된 메모리 관리자는 요청이 빈번한 1개 페이지 할당에 대하여 효율적으로 동작하면서 동시에 간헐적으로 발생되는 다중 페이지 할당 요청에 대해서도 가용성을 보장한다. 이는 크게 두 가지 기법으로 구성되어 있다. 첫째, 다중 페이지 요청 시 병합되는 비용을 최소화하기 위해 메모리 영역을 최대 병합 가능 크기(최대 할당 요청 크기)로 분할하여 관리하는 버디 블록 관리자(Buddy Block Manager)를 설계하였다. 이 기법은 작은 단위의 잠금 정책이 적용 가능하므로 멀티 코어 시스템에서 메모리 자원에 대한 락(Lock) 경쟁을 회피하는데도 효과적이다. 둘째, 병합/분할의 연산 비용을 최소화 하기 위하여 1개 페이지 할당 시 O(1)의 시간 복잡도를 갖는 확장 가능한 단일 페이지 할당자(Extensible single page Allocator)를 설계하였다. 이는 할당/반납이 빈번한 1개 페이지 요청에 대하여 연산 비용을 최소화 할 수 있으며 다중 페이지 할당 요청에 대해서도 효과적으로 처리할 수 있다.

1) 버디 시스템은 연속된 페이지를 하나의 그룹으로 구성하여 관리 하며 본 논문에서는 이를 버디 그룹으로 표현하였음

요구 병합 메모리 관리자는 인텔사의 Xeon 시스템상에서 리눅스 커널 2.6.32 버전에 구현되었다. 실험에 사용된 Xeon 시스템은 16개의 CPU와 32GB의 메인 메모리를 탑재하고 있다. 한편 벤치마크 프로그램으로는 Kernel Complie, Lmbench, Sysbench-OLTP, Unixbench를 사용하였으며 각각의 벤치마크에 대하여 1~32개의 쓰래드를 생성하여 실험 하였다. 실험 결과 기존 지연 버디 시스템에 비하여 최대 69% 평균 56% 성능이 향상되었음을 확인하였다.

참고 문헌

[1] Intel, Extended Memory 64 Technology Software Developer's Guide, 2004

[2] Samsung Electronics,
http://www.samsung.com/Products/Semiconductor/Products.htm

[3] Wolfgang Mauerer, "Professional Linux Kernel Architecture", Wiley Publishing Inc., 2008.

[4] James L. Peterson , Theodore A. Norman, Buddy systems, Communications of the ACM, v.20 n.6, p.421-431, June 1977

[5] R. E Barkley and T. Paul Lee, "A Lazy Buddy System Bounded by Two Coalescing Delays per Class", In Proceeding of the ACM Symposium on Operating System Principles (SOSP), pp. 167-176, 1989