토폴로지 인지 기반 공여 메모리 관리 메커니즘 연구

김영호*, 안신영*, 임은지*, 차규일*
*한국전자통신연구원 클라우드컴퓨팅연구부
e-mail : {kyh05,syahn,ejlim,gicha}@etri.re.kr

A Study on Mechanism for Topology-aware based Granted Memory Management

Young-Ho Kim*, Shin-Young Ahn*, Eun-Ji Lim*, Gyu-Il Cha*
*Dept. of Cloud Computing Research, ETRI

요 약

본 논문에서는 고속 저지연 네트워크로 연결된 다수의 분산 메모리 공여 노드를 통해 분산 통합 메모리 서비스를 제공하는 메모리 가상화 시스템에서, 대용량 메모리와 다수의 호스트 채널 어댑터(HCA)를 장착한 공여 노드의 프로세서, 물리 메모리, 그리고 HCA 의 연결구조와 정보로부터 토폴로지 구조를 추출하고, 프로세서 중심으로 자원 연관성 정보를 나타내는 토폴로지 맵을 생성한다. 토폴로지 맵을 기반으로 공여 메모리의 초기화, 등록, 할당 및 메모리 데이터 전송 등을 수행하는 공여 메모리 관리 메커니즘을 제안한다. 이를 통해 대용량 분산 통합 메모리를 이용하는 빅데이터 처리 환경에서 참조 데이터 대한 메모리의 응답 시간 및 접근 지연 시간을 최소화시킬 수 있다.

1. 서론

고성능 컴퓨팅 분야에서는 빅데이터 응용 같은 대용량의 데이터 분석 및 처리를 위해 고속의 스토리지 저장 장치와 대용량 고속 메모리에 대한 요구가 증대되고 있다. 기술의 발전 측면에서 보면 컴퓨팅 능력의 발전 속도와 처리 데이터 크기에 비해 요구되는 메모리 용량 및 성능이 상대적으로 부족한 측면이 있다. 스토리지와 저장 장치간의 저장 용량 및 접근 속도 등의 차이로 인해 고속 대용량 데이터 처리의 병목이 발생한다. 대용량 고속 메모리에 대한 요구를 해결하기 위해 분산 노드의 메모리의 일부를 활용하는 분산 메모리의 메모리 가상화 시스템에 대한 연구가 활발히 진행되었고, 분산 메모리를 직접 접근 할 수 있는 저지연 고속 네트워크나 연결망 기술의 발전으로 분산 메모리 통합 서비스가 대규모의 시스템에서 가능하게 되었다.

다수의 분산 노드의 공여 메모리(granted memory)를 이용한 소프트웨어 기반의 원격 메모리 가상화 기술은 다양한 방식으로 연구되고 발전되었다. 초기에는 순수 소프트웨어 형태의 메모리 가상화 기술에 대한 연구가 이루어졌지만, 성능상의 이유로 프로토타입 수준으로 개발되었다. RDMA[1] 등의 원격 메모리를 직접 접근할 수 있는 고속 저지연 네트워크와 연결망 기술의 발전으로 실제 상용화된 형태의 메모리 가상화 제품 및 서비스가 출시되고 있다. 메모리 가상화 시스템은 구현되는 수준에 따라 사용자 동작 수준, 커널 수준, 그리고 하이퍼바이저 수준으로 구분된다. 또한 구현 방식에 의해 순수 소프트웨어 방식과 하드웨어 지원 방식으로 구분이 된다. 이 중에서 순수하게 사용자 동작 수준에서 구현된 Rice 대학의 TreadMarks[2]와 커널 수준에서 구현된 UCLA 대학의 Mirage[3]가 개발되어 일부 상용시스템에서 사용되었다. 하이퍼바이저 수준에서는 Virtual Iron 사의 VFe[4], New South Wales 대학의 vNUMA[5], ScaleMP 사의 vSMP Foundation[6]등이 대표적인 결과물이다. VFe, vNUMA 와 vSMP 기술은 공통적으로 다수의 노드를 소프트웨어 계층으로 추상화한 후 단일 시스템 이미지(SSI: Single System Image)가 운영될 수 있는 SMM(Shared-Memory Multi-processor) 시스템으로 가상화하는 것을 목적으로 한다. 특히, 최근 빅데이터 기술을 처리하기 위해 고성능을 제공하는 단일 서버 개발이 다양한 물리적 요인으로 제한 받는 상황에서 다수의 서버를 통합하여 가상화한 후 고성능의 Scale-up 시스템을 구축하는 것이 기술의 궁극적 목표이다.

최근에는 하드웨어 기술의 발전으로 메모리 가상화 시스템에서 사용되는 노드 자원들의 규모와 성능이 증가하고 있다. 프로세서는 멀티소켓 구조로 발전하고, 메모리 용량 또한 노드 당 수백 GB 까지 확장이 가능하다. 프로세서 소켓 별로 메모리를 장착할 수 있는 메모리 모듈이 별도로 존재하며 동일 노드의 메모리 간에도 장착된 소켓의 프로세서 위치에 따라 접근 비용의 차이가 발생하는 NUMA(Non-uniform Memory Access) 형상으로 동작한다. 프로세서와 메모

본 연구는 미래창조과학부 '범부처 Giga KOREA 사업'의 일환으로 수행하였음.
[GK14P0100, Giga Media 기반 Tele-experience 서비스 SW 플랫폼 기술 개발]

리 및 HCA(Host Channel Apapter)의 연결 구조는 다수의 HCA 가 연결된 경우 각 프로세서 소켓과 PCI 를 통해 연결된다. 분산 노드의 대용량 메모리를 공여 메모리로 제공하기 위해서는 RDMA 를 통해 물리 메모리 접근 서비스를 지원하는 다수 HCA 의 사용이 요구된다. RDMA 통신에서는 공여 노드의 프로세서 사용 부하를 줄이고, 하드웨어 영역에 대한 접근 지연 시간을 최소화하고, 클라이언트에서 분산 공여 노드의 물리 메모리 영역에 직접 접근하기 위해서 HCA 에 가상 주소와 물리 주소 공간을 매핑하고 등록하는 메모리 영역 등록 작업과 Memory Semantic 으로 동작이 요구된다. 메모리 영역 등록 작업은 내부적으로 시스템의 물리 페이지 단위로 이루어지며 등록 메모리 영역의 크기가 증가하면 등록 비용이 선형이 아니라 로그 스케일로 증가한다[7].

기존의 메모리 가상화 시스템은 다중 HCA 와 NUMA 형상의 대용량 메모리가 장착된 메모리 공여 노드의 토폴로지 구성과 연관성에 대한 고려가 이루어지지 않아, HCA 에 인접하지 않은 메모리가 할당이나 등록되어 메모리 서비스의 응답 지연 시간이 증가하고 이로 인한 성능 저하가 발생할 수 있다.

2. 분산 메모리 통합 프레임워크 구조

본 논문의 기반이 되는 메모리 가상화 시스템인 메모리 클라우드는 메모리를 공여하는 분산 노드를 통합하여 대용량 논리 메모리 그룹을 제공하는 메모리 가상 프레임워크이다. 그림 1 은 분산 메모리 통합 프레임워크 구조를 나타낸다. 분산 메모리 통합 프레임워크의 기본 구조는 서버-클라이언트 모델을 따른다. 서버 부분(DMI server)은 분산 메모리 통합 관리자(DMI manager)와 분산 메모리 공여 에이전트(gagent)로 구성되고, 클라이언트 부분은 메모리 응용(Consumer)에게 메모리 확장 서비스 기능을 제공하는 분산 메모리 통합 클라이언트(DMI client)로 구성된다.

DMI server 는 분산 메모리 풀 관리(DM pool mgmt.), 분산 메모리 공여(DM grant), 그리고 메모리 확장 서비스(Memory extension service)의 서버측 기본 기능 제공 등의 역할을 담당한다. DMI server 는 DMI manager 와 gagent 로 구성되어 있다. gagent 는 실제로 gnode 에 존재하는 특정 지역 메모리를 확보하고 Consumer 에게 메모리 확장 서비스를 제공하기 위해 DMI manager 에게 이 메모리를 등록(Registration)하는 분산 메모리 공여(Grant) 과정을 수행한다. DMI manager 는 다수 gagent 의 등록 요청을 받아들여 분산 메모리 풀 관리를 수행한다. DMI manager 가 수행하는 분산 메모리 풀 관리는 다수 DMI client 의 메모리 요구를 받아 분산 메모리를 할당하거나 해지하고, 사용된 분산 메모리의 사용 상황을 추적하는 기능 등으로 이루어 진다. DMI server 의 또 다른 기능은 메모리 확장 서비스의 서버측 기능 제공이다. 이 기능은 확장 메모리 할당/해제 처리와 분산 메모리의 실접근 서비스를 포함한다. 확장 메모리 할당, 해제 처리는 DMI manager 에 의해 수행되며, 분산 메모리의 실접근 서비스는 해당 분산 메모리를 공여한 gagent 에 의해 수행된다.

DMI server 서버 기능을 수행하는 DMI manager 와 gagent 는 분산 메모리 풀 관리에 있어 내부적으로 서버와 클라이언트 관계에 있다. 그러나, 분산 메모리 풀 관리 중에서 분산 메모리 할당, 해제와 분산 메모리의 실접근 서비스를 대상으로 하는 서버와 클라이언트 역할은 DMI server 와 DMI client 에 의해 수행된다. DMI client 는 Consumer 와 DMI server 의 중간에서 메모리 확장 서비스를 Consumer 에게 제공한다. DMI client 는 Consumer 의 확장 메모리 할당, 해제 요청을 받아 DMI manager 에게 메모리 할당, 해제 요청을 중계하고 할당 받은 분산 메모리를 가상 메모리 영역에 사상(Mapping)한다.

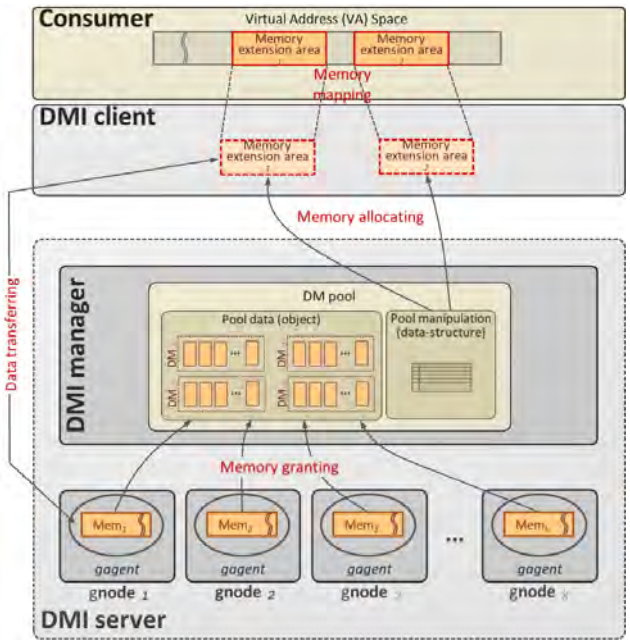

그림 1 분산 메모리 통합 프레임워크 구조

3. 토폴로지 인지 기반 공여 메모리 관리 개념

대용량 메모리를 장착한 공여 노드에서는 메모리 서비스 대역폭 확대와 사용자 지원을 위해 하나 이상의 HCA 의 지원이 요구된다. 기존 메모리 가상화 기술에서는 다중 HCA 와 수백 GB 이상의 대용량 메모리를 장착한 환경에 대한 고려가 거의 이루어지지 않고 있다. 공여 노드의 공여 메모리 초기화 및 등록 단계에서 HCA, 프로세서 및 메모리의 연결 구조를 고려하지 않아 최적의 등록과 할당이 어렵고, 이로 인해 공여 메모리 응답 시간이 증가할 수 있다.

3.1. 토폴로지 인지 토폴로지 맵 구성

본 논문에서는 시스템을 구성하는 장치들의 토폴로지 구조를 인지하여 HCA 와 메모리간의 최적 할당 및 접근 관리를 제공하기 위해 동일 소켓에 연결된 메모리와 HCA 를 매핑시켜 공여 메모리 초기화 및 할당 작업에 반영한다. 공여 메모리 등록 작업을 수행하기 전에, 시스템 및 자원의 연결 구조 정보를 추출하여 토폴로지 맵(topology map)을 구성한다. 토폴로

지 맵은 HCA 와 프로세서 그리고 물리 메모리간의 연결 구성에 대한 매핑 정보이다. 토폴로지 맵을 이용하여 최적의 공여 메모리 등록 수행이 가능하다. 먼저, 가상 메모리 영역 중 공여 메모리 영역으로 사용 될 가상 메모리를 할당한다. HCA0 를 통해서는 DRAM0 의 공여 메모리 영역에 대한 초기화 및 메모리 영역 등록 작업을 수행하고, HCA1 을 통해서는 DRAM1 의 공여 메모리영역에 대한 초기화 및 메모리 영역 등록 작업을 처리한다.

그림 2 는 토폴로지 기반의 공여 메모리 맵의 자원 간 연관 정보를 표현한 도식도이다. 프로세서를 중심으로 생성되는 토폴로지 맵 테이블의 정보는 다음과 같다.

- 토폴로지 자원(장치) 목록
- 프로세서 연관 자원(RAM, HCA) 정보
  - 자원 별 개수
  - 자원 별 용량 정보
- HCA 최대 등록 가능 메모리 영역 정보

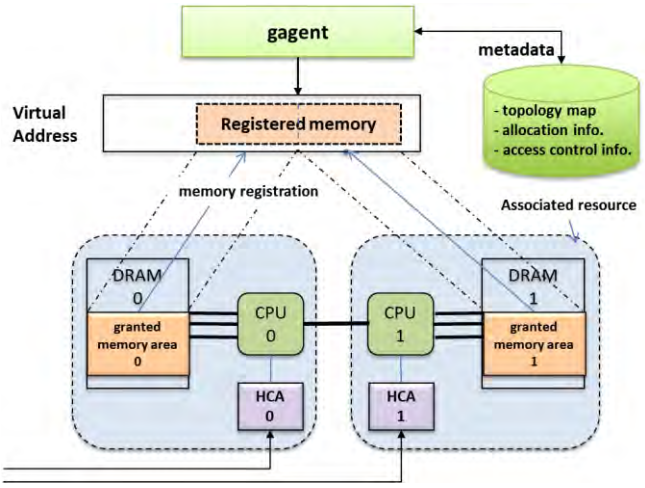

그림 2 토폴로지 기반 공여 메모리 맵 도식도

gagent 는 HCA 의 개수를 확인하고 개수대로 공여 메모리 블록(DM block)을 할당하고 특정 HCA 에 할당된 DM block 을 HCA 에 등록하는 초기화 작업을 수행한다. 각 HCA 의 DM block 에 대해 초기화가 완료되면, 분산 메모리 통합 관리자(DMI manager)의 분산공유 메모리 풀(DM pool)에 DM block 을 등록하고 DMI client 로 부터의 연결 요청에 대기한다.

3.2. 토폴로지 인지 기반 분산 메모리 관리

토폴로지 인지 기반의 분산 메모리 관리 메커니즘은 그림 3 과 같은 절차를 통해 표현될 수 있다. 토폴로지 맵을 기반으로 분산 메모리 통합 프레임워크에서 분산 노드의 공여 메모리를 등록하고, 분산 메모리 풀을 구성하는 절차에 대해 설명한다. 자원의 연관성 정보인 토폴로지 형상을 기반으로 하는 토폴로지 맵 테이블 구성 및 공여 노드의 공여 메모리 등록 절차는 다음과 같이 수행된다.

1. 분산 메모리 공여자는 시스템(운영체제, 드라이버 등)에서 자원 정보를 수집한다.
2. 프로세서 중심으로 연관 자원(RAM, HCA) 정보를 추출한다.
3. 다중 HCA 시스템인 경우, HCA 와 연관된 메모리 배치 정보를 기반으로 토폴로지 맵 데이터를 생성한다.
4. 토폴로지 맵 정보를 메타데이터 저장소에 저장한다.
5. 각 HCA 와 연관된 메모리를 공여 메모리로 할당하고 초기화한다.
6. 할당된 공여 메모리를 HCA 의 메모리 영역으로 등록한다.
7. gagent 는 분산 메모리 공여 노드의 공여 메모리를 DMI manager 에 등록 요청한다.
8. DMI manager 는 DM pool 을 재구성하고 관련 메타데이터를 업데이트 한다.

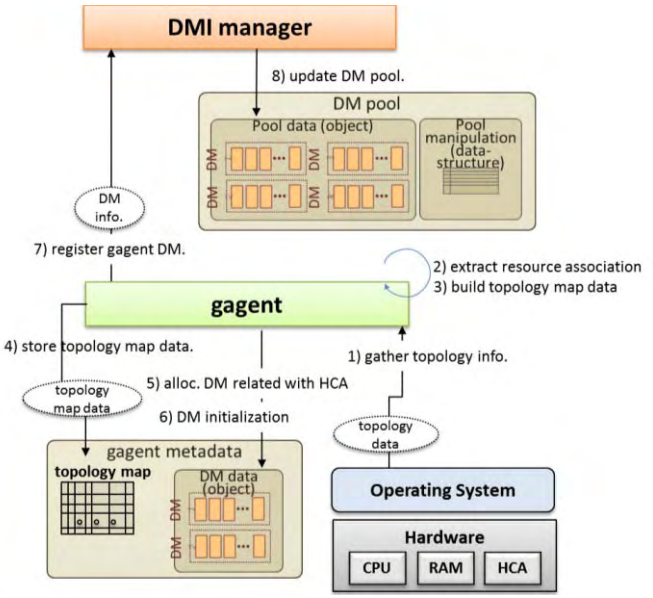

그림 3 토폴로지 기반 분산 공여 메모리 관리

분산 공여 노드의 공여 메모리 등록이 완료되면 분산 메모리 풀을 통한 메모리 클라우드 서비스가 이루어진다. 분산 메모리 클라이언트는 분산 메모리 통합 관리자를 통해 메모리 할당, 해제 등을 요청하고 메모리 서비스를 제공받는다. 할당된 메모리의 실제 메모리 데이터 읽기, 쓰기 작업은 분산 메모리 공여자를 통해 이루어지는데, 토폴로지 맵을 통해 서비스를 처리하는 프로세서와 연관된 HCA 메모리가 할당되었기 때문에, 메모리 접근 및 데이터 서비스 작업은 응답 지연 시간이 최소화되도록 동작한다. 즉, 분산 메모리 구성과 등록 시에 토폴로지를 인지한 관리 데이터 정보가 구성되어, 할당 시에는 별도의 고려 없이 이전과 동일한 할당 정책을 이용해도 HCA 와 연관된 공여 메모리가 할당되어 서비스된다.

4. 결론

본 논문에서 제안하는 토폴로지 기반 공여 메모리 관리 메커니즘은 대용량의 메모리와 다중 HCA 를 가

진 분산 공여 메모리 노드의 초기화, 등록, 그리고 메모리 할당 등의 공여 메모리 관리 시에 토폴로지 구조를 기반으로 생성된 토폴로지 맵을 적용하여 효율적인 공여 메모리 관리를 수행하는 것이다. 기존 기술에서는 공여 노드의 공여 메모리 초기화 및 등록 단계에서 HCA 와 프로세서 및 메모리의 연결 구조를 고려하지 않아 최적의 등록과 할당이 어렵고, 이로 인해 공여 메모리 응답 시간이 증가하여 성능 저하가 발생할 수 있다. 제안하는 토폴로지 인지 방식은 토폴로지 맵을 이용하여 메모리 서비스의 응답 지연 시간을 최소화 할 수 있는 공여 메모리 관리가 가능하다.

향후 계획은 제안된 방식을 적용한 분산 메모리 통합 프레임워크를 구현하고 다양한 워크로드 및 벤치마킹을 이용한 성능 측정을 통해 다중 HCA 와 대용량 메모리를 장착한 분산 공여 메모리 노드에서 데이터 서비스의 응답 성능 향상을 검증하는 것이다.

참고문헌

[1] InfiniBand Trade Association. Infiniband technology overview. http://www.infinibandta.org/about/, accessed on 30th September 2008.

[2] P. Keleher, S. Dwarkadas, A. L. Cox, and W. Zwaenepoel. Treadmarks: Distributed shared memory on standard workstations and operating systems. In Proc. of the Winter 1994 USENIX Conference, pages 115–131, 1994.

[2] Brett D. Fleisch and Gerald J. Popek. Mirage: A coherent distributed shared memory design. In Proceedings of the 12th ACM Symposium on OS Principles, pages 211–223, 1989.

[4] Alex Vasilevsky. Linux virtualization on Virtual Iron VFe. In 2005 Ottawa Linux Symp., Jul 2005.

[5] M. Chapman and G. Heiser. Implementing transparent shared memory on clusters using virtual machines. In Proc. of USENIX Annual Technical Conference, 2005.

[6] The Versatile SMP (vSMP) architecture and solutions based on vSMP Foundation. ScaleMP White Paper

[7] F. Mietke, R. Rex, R. Baumgartl, T. Mehlan, T. Hoefler, and W. Rehm, Analysis of the Memory Registration Process in the Mellanox InfiniBand Software Stack. ;In Proceedings of Euro-Par. 2006, 124-133.