공유 메모리와 사용자 수준 통신을 이용한 PVM 성능 개선

*윤종원, *심재홍, **문경덕, *최경희, *김재훈, ***정기현
*아주대학교 정보 및 컴퓨터 공학부, **한국 전자 통신 연구원, ***아주대학교 전자 전기 공학부.

Performance Improvement of PVM by using Shared Memory and User Level Communication

*Jong-Won Yun, *Jae-Hong Shim, **Kyung-Duck Moon, *Kyung-Hee Choi, *Jai-Hoon Kim, ***Gihyun Jung
* Division of Information & Computer Engineering, Ajou University
** Electronics and Telecommunications Research Institute.
*** Division of Electronical & Electronics Engineering, Ajou University

요 약

현재 PVM은 네트워크을 통해 워크스테이션을 연동시키는데 이용되는 미들 웨어이다. 이 PVM의 로컬 태스크끼리의 메시지 교환 시에는 좋은 성능을 보이지만, 리모트 호스트사이에서의 메시지 교환에서의 성능은 로컬 태스크만큼 좋지 못하다. 이를 개선하기위해, PVM에서 한 워크스테이션내의 데몬 태스크와 일반 태스크, 네트워크 카드가 모두 공유하는 공유메모리와 고속 네트워크 카드인 미리넷 카드를 이용해, 다가올 초고속통신망에서 적용되기 위한 새로운 모델을 제시하고, 기존 모델과 비교한다.

1. 서론

네트워크를 통해 연결된 병렬 컴퓨팅 환경은 하드웨어적 멀티프로세서(multiprocessor)에서는 경제적인 면에서 매력적이고, 효과적이다. 이런 병렬 컴퓨팅을 위해 하드웨어나 소프트웨어가 기존에 일반적으로 사용하는 것과 다른, 특수한 것들이 필요할 수도 있지만, 일반적으로 사용하는 네트워크 카드와, 워크스테이션을 연결해서, 병렬적인 응용프로그램을 수행할 수도 있다. [2]

PVM 시스템은 후자쪽에 속하며, 사용자 수준 통신 프리미티브(primitives)들과, 루즐리 커플(loosely coupled) 네트워크상에서 컨커런트 컴퓨팅(concurrent computing)이 가능한 software로 구성되어있다.[2]

현재 버전 3.3의 PVM 에는 메시지 교환 시 유닉스의 소켓(socket)을 이용한 것과, 공유 메모리를 이용한 것 모두 있다. 이중, 공유 메모리를 이용한 기존의 PVM은, 로컬 태스크상에서의 메시지의 교환은 거의 최적화되어 있다. 하지만, 리모트 호스트에 있는 태스크사이의 메시지 교환은 메모리상에서 빈번한 복사가 발생 하기 때문에 성능이 좋지 못하다. 그러므로 본 논문에서는 리모트 호스트간에서 메시지 교환 시, kernel로의 메모리 복사를 줄임으로서 PVM의 성능을 개선 시키려 한다.

2. 기존 모델과 개선된 시스템의 모델 비교

기존의 PVM 시스템에서 로컬 태스크들은 공유 메모리를 이용하지만, 리모트 호스트에 있는 태스크 사이에는 결국 *socket*을 이용한다. 그래서 커널로의 메모리 복사가 발생하게 된다. 이장에서는 기존의 시스템 모델을 간략하게 살피고, 나아가 PVM의 로컬 태스크와 데몬(daemon), 네트워크 카드가 모두 공유하는 공유 메모리를 이용하여 메모리 상에서의 메시지 *copy*의 수를 줄이는 방법으로 개선된 PVM 시스템 모델을 제시하겠다.

2.1 기존의 시스템 모델

기존의 모델에서는 아래의 그림 1 처럼, PVM 태스크에서 공유 메모리로, PVM 데몬의 로컬영역, Kernel 네트워크 card의 순으로 데이터 copy가 발생한다. 하지만, Kernel 이후의 데이터 copy는 여기서 *copy*로 생각하지 않는다. 왜냐하면, Kernel이 데이터를 DMA 영역으로 보내면, 이후에는 네트워크 카드에서 처리하기만 하고, 더 이상 CPU time을 소비하지 않기 때문이다. 그리고, PVM 태스크의 로컬 메모리에서 공유 메모리 영역으로 copy되는 것은, 같은 프로세스의 영역에서

데이터가 오가는 것이지 다른 프로세스의 영역으로 데이터가 전달 된 게 아니므로 *데이터 copy*라고 여기지 않는다. 그래서, 기존의 PVM 공유 메모리 모델에서는 *데이터 copy*가 2 회 발생한다.

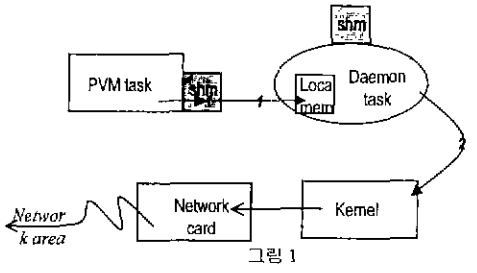


그림 1

2.2 개선시키려는 시스템의 모델

위의 모델에서 발생하는 메모리 copy의 횟수를 미러넷 카드와 한 호스트에서의 global 공유 메모리를 이용해서 *zero copy* 로감소시키는 모델을 제시한다.

리모트 호스트로 전송하는 경우, 태스크의 로컬영역에서 공유 메모리의 DMA 영역으로 copy 해주면, 여기서부터는 위에서 말한 바와 같이, CPU 의 관여 없이 미러넷 card가 일아서 네트워크로 데이터를 전송해준다. 그러므로 DMA 영역 이후에 발생하는 데이터 copy는 CPU time에 영향을 주지 않는다. 그래서 우리는 커널의 DMA 영역이후에 네트워크 card로 발생하는 메모리의 copy는 *메모리 copy*라고 여기지 않는다. 때문에 본 논문에서 제시한 새로운 모델은 기존 2 회의 메모리 copy 횟수를 *zero copy*로 감소시킴으로써 성능을 향상 시킨다.

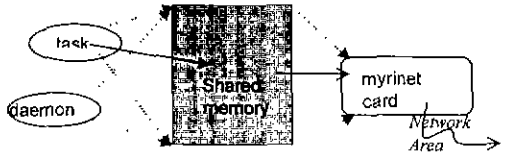


그림 2

3. Implementation 할 방법

현재 PVM에서 태스크, 데몬 태스크, 네트워크 카드가 모두 공유하는 공유 메모리 시스템을 이용해 성능을 개선하기 할 때 미러넷 카드가 사용하는 physical address와 OS인 Linux에서 사용하는 virtual address의 맵핑 문제가 발생한다. 이번 절에서는 이 맵핑 문제에 대한 해결책을 제시하겠다 그리고 개선된 방식으로 메시지를 교환 시 어떤 절차로 이루어지는지도 보이겠다

3.1 메모리의 맵핑

일반적으로 사용하는 OS는 대게 가상 메모리 시스템을 가지고 있다. 그렇기 때문에, 물리적으로는 같은 메모리의 주소 위치를 나타내도, 각 프로세스는 다른 주소로 그 공간을 지시하게 된다 또한, 가상 메모리 시스템에서 연속적인 address로 표시된다고 해서, 물리적인 메모리 address도 연속적으로 가리켜 지는 것은 아니다. 하지만, 미러넷 card는 가상 메모리 시스템을 사용하지 않고, physical address만을 인식하기 때문에, 미러넷 카드와 각 프로세스는 바로 그 주소 공간이 UNIX 시스템에서 제공하는 공유 메모리를 이용한 방법으로 맵핑 되지 않는다.

이 두 가지의 상이한 주소공간을 맵핑 시키기 위해 우리는 Linux의 boot time에 미리 정해진 일정영역을 할당한다. 그리고 우리는 이 할당된 영역을 Bigphysarea라고 명칭하겠다. Bigphysarea 할당 시에 물리적 메모리에서의 시작 주소와 크기를 얻어내서, 일반 테스크에게 알려준다 각 테스크는 자신의 가상 메모리 공간에 할당된 크기의 Bigphysarea가 들어 갈 수 있는 부분을 찾아서 mmap을 이용해 물리적 메모리 공간과 가상 메모리 공간을 맵핑시켜준다. 그래서 이 이후에 태스크는 자신의 가상메모리 공간에 읽고 쓰면, 실제로는 Bigphysarea 영역에 읽고 쓰게 된다.

그리고, 각 테스크가 데이터를 위한 버퍼공간에 데이터를 쓰고, 읽어내는 일을 미러넷 카드도 같이 하기위해서, send queue와 receive queue 부분도 공유한다.

3.2 리모트 호스트간의 메시지 교환

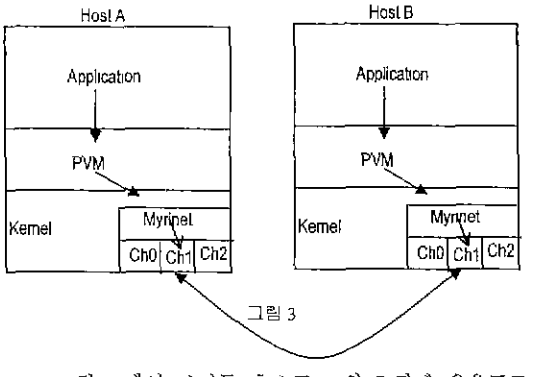


그림 3

그림 3에서 보이듯 호스트 A와 B 간에 응용프로그램에서 필요하거나 메시지를 교환하거나, 미들 웨어인 PVM에서

사용하는 제어 메시지를 주고 받을 때, 개선된 구조를 이용하기 때문에 커널로의 메시지를 복사하는 오버헤드는 없어졌다. 그리고, 커널에서 처리하던 멀티 플렉싱을 위해 select 함수를 만들어 넣었다. 그래서, 이전의 시도된 PVM 모델에서는 보내는 메시지의 크기에 따라 두 가지로 나누어 미러넷 카드의 채널 0 과 1 번 두개에 각각 UDP/IP 와 미러넷 API 를 이용해 메시지를 전송했다. 하지만 지금은 select 함수를 미러넷의 통신 프리미티브로 사용하기 때문에, 이전처럼 데이터의 전송을 알리기 위해 미리 메시지를 전송하는 오버헤드조차도 필요가 없어졌다. 또한 이전에 사용하던 두개의 채널 중 UDP/IP 통신을 위한 0 번은 사용하지 않고 1 번만을 사용한다.

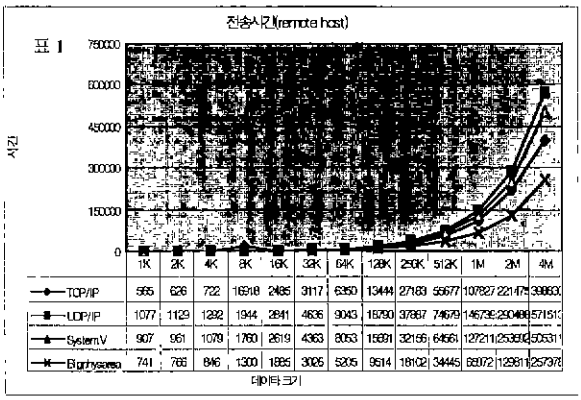
4. 기존모델과의 성능 비교.

현재 위에서 제안된 모델은 구현이 되어 있다. 기존 PVM 과 성능을 비교하기 위한 실험과 결과를 여기서 소개하겠다. 실험을 한 환경은 펜티엄 프로 200, 주기억장치가 64Mbyte 인 PC 2 대를 미러넷 card 로 연결하였다. 운영체제로는 linux red hat 4.2 를 설치하여 실험했다. 그리고, 여기에 설치된 미러넷 카드는 Myricom 사에서 생산된 것으로 전송속도는 650Mbps 이고, 메모리는 256k 이다.

두 모델간의 성능 비교를 위한 실험은 미러넷이 제공하는 TCP/IP, UDP/IP, 시스템 V 의 공유메모리, Bigphyarea 네 가지 방법을 로컬 태스크 사이의 통신과 리모트 호스트에 있는 태스크 사이의 통신 각각에서 모두 전송시간을 측정, 비교하였다. 결론적으로 말하면, XDR 로 packing 하느데는 거의 같은 시간이 걸리지만, 메시지 전송 시에는 Bigphysarea 모델의 성능이 기존방식보다 향상되었음을 보여주고 있다.

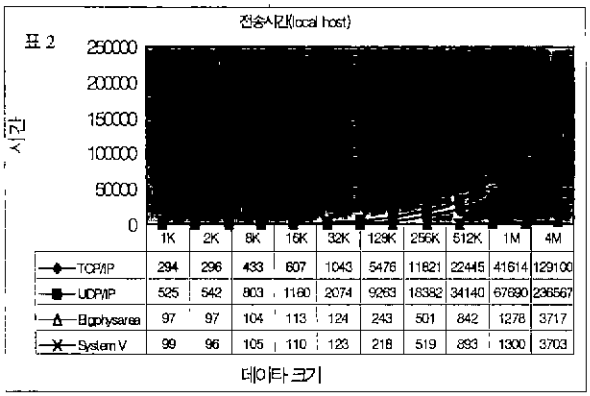
아래 표 1 은 리모트 호스트간의 결과이고, 표 2 는 로컬 태스크사이의 결과이다.

표 1



	1K	2K	4K	8K	16K	32K	64K	128K	256K	512K	1M	2M	4M
TCP/IP	565	626	722	16918	2465	3117	6350	13444	27183	55677	107827	221475	39863
UDP/IP	1077	1129	1232	1944	2941	4636	9043	18250	37887	74679	145736	250498	571513
System V	907	961	1079	1760	2619	4363	8053	15691	32156	64566	127211	253896	505331
Bigphysarea	741	766	846	1300	1885	3025	5205	9514	18102	34445	68972	129816	257978

표 2



	1K	2K	8K	16K	32K	128K	256K	512K	1M	4M
TCP/IP	294	296	433	607	1043	5476	11821	22445	41614	129100
UDP/IP	525	542	803	1160	2074	9263	18382	34140	67690	236567
Bigphysarea	97	97	104	113	124	243	501	842	1278	3717
System V	99	96	105	110	123	218	519	893	1300	3703

5. Conclusion 및 future work

본 논문에서는 고속 통신에서 PVM 성능을 개선하기위한 한가지의 모델 제시했다. 우리가 제시한 모델은 Bigphysarea 라는 공유 메모리를 태스크와 미러넷 카드가 같이 사용하고, 메시지를 네트워크로 전달하기위해 커널로 메시지가 복사되는 오버헤드를 없앴다. 그리고, 커널이 담당하던 멀티 플렉싱을 미러넷 카드에서 처리하기위해 select 함수를 구현해 넣었다 개선된 PVM 의 성능을 검증하기 위해 현재까지는 간단한 전송 시간정도만 한 상태이지만, 그 실험에서는 개선된 PVM 이 더 나은 성능을 보였다. 하지만, 이런 간단한 전송시간만 측정한 것에서 더 나아가, 실제로 병렬 컴퓨팅 문제로 사용되는 실험용 프로그램을 이용해 더 많은 테스트를 할 예정이다.

참고문헌

[1] Geist, A Beguelim, J. Dongara, W.Jiang, R. Mancheck, and V. Sunderam, *PVM 3 User's Guide and Reference manual.* Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, Sept 1994

[2] Thorsten von Eicken, Anindya Basu, Vineet Buch, and Werner Vogels *U-Net. A User-Level Network Interface for Parallel and Distributed Computing* Proc of the 15th ACM Symposium on Operating Systems Principles, Copper Mountain, Ccolorado, December 3-6, 1995

[3] Geist, G A , Sunderam V.S , *Network-Based Concurrent computing on the PVM System, Concurrency. Practice and Experience*, 4 (4),293-311, June 1992.

[4] M. Welsh and A. Basu and T. von Eicken, *Low-Latency Communication over Fast Ethernet,* Proc EUROPAR 96. August 1996.

[5] M. Welsh and A Basu and T von Eicken, *Incorporating Memory Management into User-Level Network Interfaces,* Proc Hot Interconnects V, August 1997