리피터 노드를 장착한 이중 링 CC-NUMA 시스템

경진미[0], 김인석, 김봉준, 장성태
써머스테크놀로지[0], 수원대학교 컴퓨터학과
flora@summustech.com[0], {iskim, bjkimst, stjhang}@suwon.ac.kr

Dual Ring CC-NUMA System using Repeater Node

Jin-Mi Kyoung[0], In-Seok Kim, Bong-Joon Kim, Sung-Tae Jhang
Dept. of Office, Summustech.com[0], Dept. of CS, Univ. of Suwon

요 약

CC-NUMA 구조에서는 원격 메모리에 대한 접근이 불가피한 구조적인 특성 때문에 사호 연결망이 성능을 좌우하는 큰 변수로 작용한다. 기존에 사용되는 버스는 대역폭의 한계와 물리적 확장성 때문에 대규모의 시스템에는 적합하지 않다. 이를 대체하는 고속의 지점간 링크를 도입한 이중 링 구조는 이러한 버스의 한계를 극복하고는 있지만 많은 노드를 거쳐야 하는 문제로 인해 응답 지연 시간이 증가하는 단점을 안고 있다. 본 논문에서는 요청과 응답 패킷의 지연 시간을 줄이는 방안으로 리피터 노드를 이용한 다중 링을 제안한다. 제안된 시스템은 링과 링 사이의 구조가 대칭형을 이루고 있어 요청을 내보내는 링을 제외한 다른 링의 hop수는 똑같은 수치를 갖고 있으며, 이중 링에 비해 최대의 hop수와 최소의 hop수의 차가 적고 평균 hop수 또한 적어 좋은 성능을 보인다. 본 논문에서는 또한 이러한 구조를 유지하기 위한 리피터 노드의 구조를 제안하며 리피터 노드의 구조와 노드의 확장에 따른 다양한 성능을 확률 구동 시뮬레이터를 사용하여 평가를 수행한다.

1. 서 론

최근의 공유 메모리 다중 프로세서 시스템에서, CC-NUMA 시스템이 널리 사용되고 있는 것은 확장성과 프로그래밍이 용이하다는데 기인한다[1,2]. 하지만, 시스템의 크기가 증가함에 따라 원격 메모리로의 빈번한 접근은 메모리 접근 지연 시간(memory access latency)을 증가시킨다. 원격 메모리로의 접근 횟수를 줄이거나 접근 속도를 높이는 것이 CC-NUMA 구조의 성능 향상을 위해 필수적이다[3,4].

서울대학교의 PANDA 연구실에서 제안한 PANDA II 시스템은 스누핑 방식의 캐쉬 일관성 유지방법을 사용하며, 4개의 펜티엄 프로 프로세서가 묶인 하나의 노드를 양방향 지점간 링크로 연결하였다. 이 시스템의 특징은 링을 통한 방송 트랜잭션을 지원하여, 링을 가상적인 버스로 구성하였다는 것이다. 이러한 이중 링 구조의 CC-NUMA 시스템은 버스에 비해 속도 및 물리적 확장성에서 장점을 가지지만, 지점간 링크로 구성되는 특징 때문에 링에 연결되는 노드들의 수가 증가하면 할수록 원격 노드로의 요청 및 응답에 걸리는 시간이 증가하게 된다. 이중 링 구조가 가지는 확장성에서의 장점을 살리기 위해서는 노드 수의 증가에 따른 접근 지연 시간의 증가를 줄이며, 지속적인 성능 향상을 꾀할 수 있는 방안을 강구해야 한다.

본 논문에서는 이러한 문제를 해결하기 위해 스누핑 기반 이중 링 CC-NUMA 시스템인 PANDA II의 구조를 개선하여 방송 패킷을 동시에 중첩해서 여러 노드로 전송함으로써 요청의 전송 시간을 줄이고, 빠른 응답 시간을 제공할 수 있는 새로운 CC-NUMA 시스템을 제시하고자 한다. 본 논문에서 제시하는 시스템의 가장 큰 장점은 더 많은 노드를 장착할 수 있는 확장 가능한 구조라는 것이다.

2. 스누핑 기반 이중 링 구조의 CC-NUMA 시스템

2.1 리피터 노드를 이용한 구조

본 논문에서 제시하는 Scalable CC-NUMA 시스템의 기본 개념은 PANDA II에서 사용된 스누핑 캐쉬 일관성 프로토콜을 그대로 사용하는 동시에, 링을 기존의 한 개의 이중 링에서 여러 개의 이중 링으로 나누어 그룹화하여 링 내의 노드들 간의 hop수를 줄이고자 한다. 링으로 보내는 방송 패킷이 링과 링을 연결하는 노드를 지날 때 이 노드가 이 방송 패킷을 두 개의 링으로 동시에 전송하여 전송 시간의 중첩을 유도하며, 이러한 특징이 기존의 PANDA II 보다 좋은 성능을 나타내게 된다. 이때 링과 링을 연결하는 노드가 본 논문을 통해 제시하는 리피터 노드이다.

그림 1은 노드의 개수가 12개인 PANDA II 구조를 예시한다. PANDA II는 참조하려는 메모리 블록을 요청하는 방송 패킷의 경우 0번 노드에서 11번 노드까지 어느 노드에서 시작을 해도 전체 노드를 한번씩은 방문하며, 이 예의 경우 평균 12번의 hop을 꼭 지나게 된다. 그 요청에 대한 응답 패킷의 경우에는 해당 요청을 보낸 노드까지의 최단 경로를 찾아가도록 구성되어 있어 최대 N/2(N:노드의 수) hop을 거치게 되며, 그림 2의 예에서는 최대 hop 수가 6, 평균 hop 수가 3을 나타낸다.

그림 2는 PANDA II 시스템의 요청과 응답에 걸리는 지연 시간을 줄이기 위해 본 논문에서 제시하는 리피터 노드를 이용하여 다시 구성한 이중 링 구조의 CC-NUMA 시스템을 예시한다. 리피터 노드는 2개의 링을 연결하는 노드로 하나의 노드에서 시작한 방송 패킷을 자신의 링뿐만 아니라 다른 링으로 전송하는 역할과 다른 링을 한 바퀴 회전한 방송 패킷을 그 링에서 제거하는 역할을 수행한다. 리피터 노드를 거치면서 방송 패킷은 두 개의 링으로 동시에 전송되므로 빠르게 방송 패킷을 전송할 수 있으며, 이에 따라 빠른 응답을 기대할 수 있다.

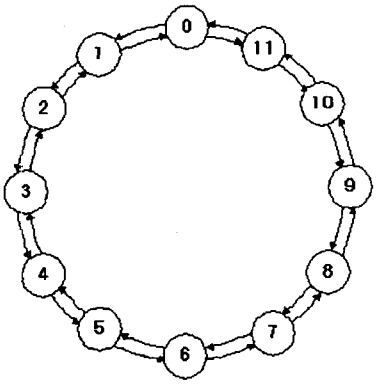

그림 1. PANDA II의 링구조(노드수:12)

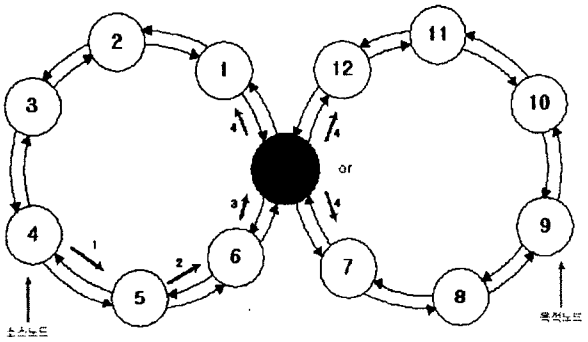

그림 2. 리피터 노드(0번노드)를 이용한 Ring2(2는 링개수)

그림 2는 소스 노드(요청을 보낸 노드)가 4번이며, 요청을 담고 있는 방송 패킷의 전송 방향은 시계 반대 방향인 경우를 예시한다. 4번 노드에서 출발하여 5번, 6번 노드를 거쳐 0번 리피터 노드를 거치면서 방송 패킷은 1번과 7번 혹은 1번과 4번 노드로 중첩된 시간에 전송된다. 소스 노드가 포함되지 않은 링도 이 방송 패킷을 반시계 방향으로 전송한다고 가정할 때 1번 노드와 7번 노드, 2번 노드와 8번 노드, 3번 노드와 9번 노드는 거의 중첩된 시간에 이 방송 패킷을 전송하며, 이는 요청 지연 시간을 줄이고 빠른 응답을 얻는 효과를 얻을 수 있다. 물론 링을 돌고 있는 패킷이 없을 경우 같은 시간에 전송하는 것이 가능하고 이미 전송중인 패킷이 있다면 그 패킷이 지난 후 큐에 놓인 순서대로 링을 돌게 된다. 그러나 이런 제약사항은 기존의 PANDA II에서도 동일하며, 제시한 구조가 PANDA II 보다 빨리 요청을 방송할 수 있게 된다. 응답 패킷도 기존의 PANDA II처럼 가장 짧은 경로를 통해 오게 되는데 그림 3의 예에서는 목적 노드인 9번에서 시작해 8번, 7번, 0번, 6번, 5번, 그리고 소스 노드인 4번으로 응답한다. 그림 3은 리피터 노드가 hop 수에 포함되므로 하나의 노드처럼 고려해 번호를 부여했다.

그림 3은 4개까지의 링을 연결할 수 있는 확장된 리피터 노드 하나만 두고 네 방향으로 노드를 분산시켜 각 링과 링 사이의 요청 지연 시간과 응답 지연 시간을 줄이도록 구성된 구조를 예시하고 있다. 하나의 링에서 나온 트랜잭션은 다른 링들로 동시에 전송될 수 있으며 이러한 중첩된 시간을 이용해 요청 지연 시간과 이로 인한 응답 지연 시간도 줄일 수 있다. 앞의 예제처럼 4번 소스 노드를 시작으로 시계 반대 방향으로 방송 패킷이 전송된다고 할 때 0번 리피터 노드를 지나면서 다른 세계의 링으로 그 방송 패킷을 동시에 전송하게 되어 더 많은 중첩된 시간을 유도하며 빠른 요청 시간과 응답 지연 시간을 기대할 수 있다.

같은 수의 노드를 연결함에 있어서 앞에서 서술한 바와 같이 하나의 링보다는 두 개의 링으로 나누고, 두 개의 링보다는 세 개, 네 개의 링으로 나누어 노드를 분산시키는 것이 응답 지연 시간을 줄이는데 효과를 나타냈으며, 이와 같은 구조는 확장 가능성 면에 있어서 훨씬 효과적이어서 PANDA II의 확장성 문제를 해결하는 하나의 방안으로서 제시될 수 있다. 그림 3은 4개까지의 링을 연결할 수 있도록 확장된 리피터 노드를 이용하여 12개의 노드에서 더 좋은 성능을 보여주는 구조로 노드를 24개까지 확장했을 때도 다른 구조보다 더 좋은 성능을 보이게 된다.

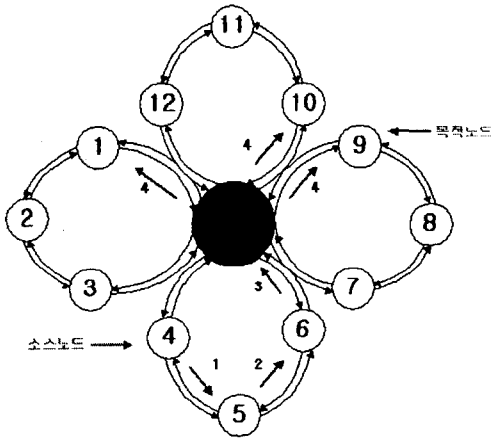

그림 3. 리피터 노드(0번노드)를 이용한 Ring4(4는 링개수)

2.2 리피터 노드의 구조

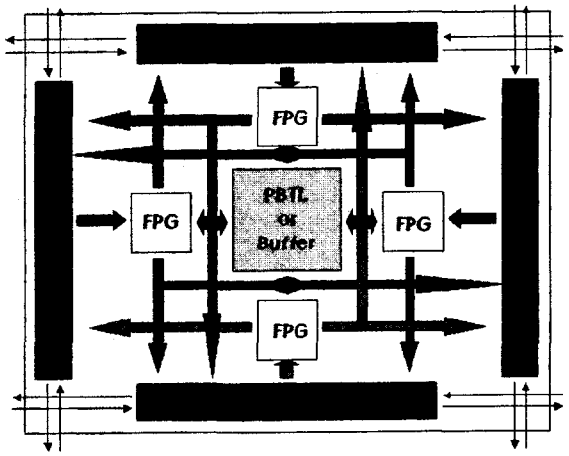

그림 4. Ring4의 리피터 구조

그림 4는 링을 4개갖는 구조의 리피터 노드의 모습이다. PBTL(Pending Broadcast Transaction List)은 포워드한 방송 패킷에 대한 정보를 담아두고 다른 노드에서 보낸 방송 패킷이 현재 Pending 중인 트랜잭션과 동일한 블록에 대한 요청일 경우 그 요청을 재전송할 것을 요구하는 기능을 갖게 된다. 가장 단순하게는 그림 10의 PBTL 없이 단순한 포워드만을 위한 기능을 고려해 볼 수 있으나 한번 전송한 방송 패킷에 대한 응답 패킷이 돌아오지 않은 상태에서 같은 주소 영역에 대한 요청을 또다시 내보낸다는 것은 불필요하고 비용의 낭비도 가져오므로 PBTL을 유지하는 것이 보다 나은 성능 향상을 얻을 수 있다.

3. 실험결과

그림 5는 트랜잭션의 발생 빈도별 트랜잭션의 요청 지연 시간을 보여주는 것이다. 이 그림에서 PANDA II의 구조와 비교해 Ring2나 Ring4는 하나의 트랜잭션의 중복 요청을 통한 요청 지연 시간에 많은 이득을 볼 수 있으며, 이는 Ring을 4개까지 연결한 구조에서 보다 효과적이라 할 수 있다. 100, 200, 300의 값은 트랜잭션의 발생 시간 간격으로 시간 간격이 클수록 발생 빈도가 낮아지며, 발생 빈도가 낮아질수록 보다 좋은 결과를 얻을 수 있다. 노드수가 24(그림5의 오른쪽)인 Ring4의 성능이 노드수가 12인 PANDA II보다 요청 시간 지연이 더 적다. 이는 리피터 노드를 이용한 구조에서는 노드수가 늘더라도 요청 시간의 부담을 줄일 수 있음을 보이며 이는 빠른 응답시간을 기대할 수 있어 전체적인 성능을 높이는 이유라고 볼 수 있다.

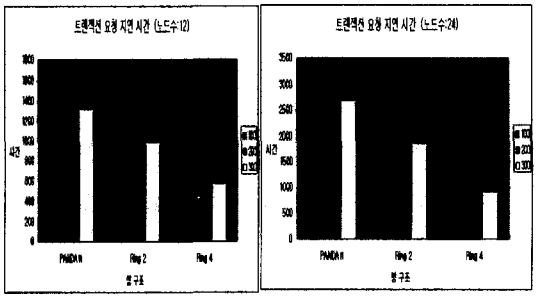

그림 5. 트랜잭션 발생 빈도별 요청 지연시간(노드수:12,24)

그림 6은 트랜잭션의 발생 빈도에 따른 응답 지연 시간을 보여준다. 요청 지연 시간과 마찬가지로 응답 지연 시간 또한 리피터 노드를 이용한 구조에서 기존의 구조보다 좋은 성능을 보인다. 특히 노드를 링에 분산해 주는 Ring4 구조는 요청과 응답 지연 시간 모두에 있어서 많은 효과를 얻어 낼 수 있으며 이는 더 많은 노드를 장착할 수 있는 확장 가능한 구조임을 보여준다.

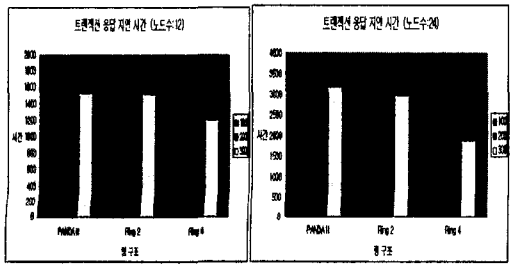

그림 6. 트랜잭션 발생 빈도별 응답 지연시간(노드수:12,24)

그림 7은 요청 트랜잭션과 응답 트랜잭션의 합을 각 구조별로 보여주는 그림으로 한번의 트랜잭션이 요청되면서 응답될 때까지의 평균 지연 시간을 나타내 주는 것이다. 그림에서와 같이 노드 수가 12인 구조에서의 링 구조 변화에 따른 지연 시간의 감소 비율보다 노드 수 24인 구조에서의 지연시간 감소 비율이 더 큼을 볼 수 있으며, 이는 확장성에 있어서 리피터 노드를 이용한 구조의 장점을 보여주는 동시에 기존의 PANDA II에서 보여주는 확장성 문제를 해결할 수 있음을 의미한다.

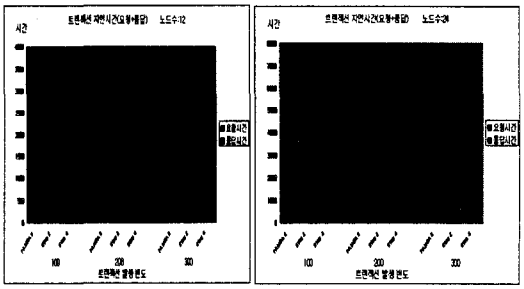

그림 7. 트랜잭션의 지연시간(노드수:12,24)

4. 결론

본 논문에서는 PANDA II의 링 구조를 본 논문에서 제시한 리피터 노드를 이용해 여러 링으로 분리해 줌으로써 각 링에 노드를 분산시키고 노드간 패킷 전송 시간을 증첩시켜서 전체 노드에 걸리는 시간을 줄이는 결과를 얻어 낼 수 있었으며 이는 요청 패킷뿐만 아니라 응답 패킷에도 좋은 결과를 가져올 수 있다. 각 링이 리피터 노드를 사이에 두고 대칭 구조를 보여줌에 따라 각 노드간의 hop의 최대치와 최소치간의 간격을 줄일 수 있으며, 각 링에 연결된 노드의 수가 기존의 것보다 적기 때문에 각 노드간의 지연 시간을 줄일 수 있다. PANDA II 구조보다 노드간의 hop의 수가 적어지므로 요청 시간 및 응답 시간 지연을 줄일 수 있는 장점 외에 더 많은 노드를 장착할 수 있는 확장 가능한 구조인 것이 기존의 구조에서 보이는 단점을 해결한 것이다. 또한 리피터 노드에 PBTL이나 버퍼를 이용해 보다 효과적으로 트랜잭션을 처리해 줄 수 있을 것으로 사료되며, 기존의 PANDA II보다 적은 hop으로 응답을 할 수 있다는 것은 이 구조가 갖는 장점이라고 볼 수 있다.

참 고 문 헌

[1] D.E. Culler and J.P. Singh, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufmann Publishers, 1999.

[2] Kai Hwang and Zhiwei Xu, "Scalable Parallel Computing : Technology, Architecture, Programming", McGraw-Hill, 1998.

[3] Zhang, Z. and J. Torrellas. "Reducing Remote Conflict Misses : NUMA with Remote Cache versus COMA", In Proc. of the 3rd IEEE Symp. on High Performance Computer Architecture(HPCA-3), pp. 272-281, Feb. 1997.

[4] L. Barroso and M. Dubois, "The Performance of Cache-Coherent Ring-based Multiprocessors", In Proceedings of the 20th International Symposium on Computer Architecture, pp.268-277, May 1993.