계층 링 구조의 변형을 통한 시스템 성능 개선

곽종욱° 반형진 전주식
서울 대학교 전기 컴퓨터 공학부
{leoniss°, jin284}@panda.snu.ac.kr, csjhon@riact,snu.ac.kr

Improving System Performance by Modifying Hierarchical Ring

Jong Wook Kwak° Hyong Jin Ban Chu Shik Jhon
Dept. of Electrical Engineering and Computer Science, Seoul National University

요 약

이 논문에서는 기존의 계층 링을 수정한 Torus Ring을 제안한다. Torus Ring은 계층 링과 완전히 동일한 복잡도를 가지면서도 지역 링 간의 연결 방법만을 변경한 형태의 상호 연결망이다. 이 연결망은 역방향 인접 링에 대한 요청에서 홉 수의 이득을 봄으로써 평균 홉 수를 감소시킨다. 또한 접근의 지역성을 고려하지 않은 균등분포의 가정 하에서도 평균 홉수의 기대값에서 계층링과 동일한 값을 가지며, 실제 병렬 프로그램이 수행되는 환경에서는 인접링에 대한 통신 비율이 증가할 가능성이 크기 때문에 더 큰 홉 수의 이익을 기대할 수 있다. 실험 결과 상호 연결망의 요청과 응답의 지연 시간 단축되었으며, 이러한 응답 지연 시간의 단축이 수행 시간을 최대 7%까지 감소시키는 결과를 가져왔다.

1. 서 론

고성능 컴퓨터 시스템의 급격한 성능 향상에 따라 사용자들의 컴퓨터 시스템의 성능 향상에 대한 기대치는 점차 증가하게 되었으며 이에 부응하기 위한 노력의 일환으로 다중 프로세서 시스템이 사용되고 있다. 다중 프로세서 시스템에서 사용되는 메모리 모델은 공유 메모리 모델과 메시지 전달 메모리 모델로 나누어 볼 수 있다[1]. 주로 사용되는 공유 메모리 모델은 메모리 접근 방식에 따라 NUMA 시스템과 UMA 시스템으로 분류할 수 있는데, 분산된 공유 메모리를 사용하는 시스템에서는 시스템 내의 여러 곳에 메모리가 분산 배치되어 각 메모리에 접근하는 시간이 상이한 NUMA 시스템이 주로 사용된다. NUMA 시스템에서는 각 노드를 연결하는 상호 연결망이 시스템의 성능을 크게 좌우하게 된다. 이러한 상호 연결망의 성능 향상을 위한 노력으로는 상호 연결망의 성능 자체를 개선하는 방법과 상호 연결망의 이용을 최적화하는 방법이 있다. 전자로는 연결망의 형태(Topology)를 개선하거나, 라우팅이나 흐름제어(Flow Control) 개선 등의 방법이 있을 수 있겠고, 후자로는 원격 접근 캐시(Remote Access Cache)의 성능을 개선하는 등 상호 연결망의 이용을 최대한 줄여 빠른 메모리 접근을 가능하게 하려는 방법 등이 있다. 본 논문에서는 전자와 같은 노력의 연장선상에서, 새로운 다중 프로세서 상호 연결망의 형태를 제안하고 그의 성능을 기존의 상호 연결망과 비교 평가해 본다. 우선 기존의 링 형태의 상호 연결망들의 성능을 비교해 보고, 계층 링의 경우 어떻게 구성하는 것이 가장 성능 향상에 유리한지를 알아본다. 또한 계층 링을 변형한 형태인 Torus Ring을 제시하고 이 연결망이 기존의 계층 링 연결망에 비해 어느 정도의 성능 향상을 가져오는지 알아본다.

2. 관련 연구

2.1 링 연결망

링 연결망은 그림 1에서와 같이 프로세서 노드를 링 형태로 연결한 비교적 간단한 형태의 연결망이다. 링 연결망은 그 간단한 구성 방법으로 인해 노드의 추가 및 제거가 쉽고, 낮은 복잡도로 인해 빠른 속도의 운용이 가능하다. 또한 링은 캐시 일관성(Cache Coherence)과 메모리 일관성(Memory Consistency)을 효과적으로 구현할 수 있는 특징을 가지는데, 요청 및 응답의 배치와 방송(Broadcast)의 용이함이 그것이다. 예컨대 하나의 무효화(Invalidation) 요청이 링 연결망을 순회하면서 여러 캐시 라인을 무효화 할 수 있도록 구현될 수 있다[2]. 이같은 이익의 극대화를 위해 스누핑(Snooping) 프로토콜을 링 연결망에서 사용하여 성능을 향상시키는 연구도 진행된 바 있다[3].

2.2 계층 링 연결망

그림 2와 같이 단일 링 형태의 지역 링을 상위 계층의 전역 링이 연결하고 있는 형태가 계층 링 연결망이다. 계층 링 연결망은 단일 링 연결망의 장점들을 거의 그대로 취하면서도 단일 링 연결망보다 노드 수의 증가에 따른 부담이 적다는 장점이 있다. 데이터의 지역성이 어느 정도 존재한다면, 128 프로세서 노드 정도까지는 계층 링 연결망이 동일한 비용의 MESH 연결망에 비해서 성능이 나쁘지 않다는 연구 결과도 있다[4].

계층 링은 2단계 이상으로 구성될 수 있는데, 이 논문에서는 2단계의 계층 링으로 비교 대상을 한정한다. 계층 링의 형태는 $m \times n$과 같이 나타낼 수 있고, m은 지역 링의 개수, n은 지역 링 내의 노드의 개수를 의미한다. 그림 2는 8×2의 계층 링 형태라고 할 수 있다.

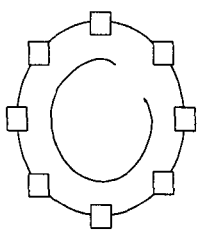

그림 1. 링 연결망

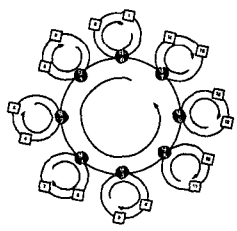

그림 2. 계층 링 연결망

3. Torus Ring의 구성과 특징

3.1 Torus Ring의 구성

본 논문에서 제안하는 Torus Ring은 아래 그림 3과 같은 형태의 링 연결 구조이다.

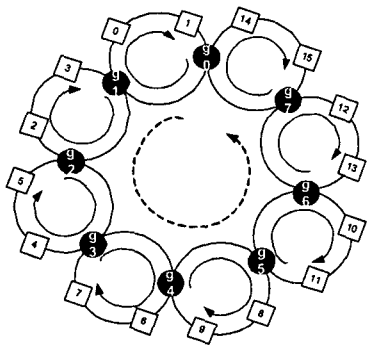

그림 3. Torus Ring 연결망

사각형 내의 숫자는 프로세서 유닛이 달려 있는 노드 번호를 의미하며, 검은색의 원은 프로세서 유닛이 없이 스위치 역할만 하는 "전역 링" 스위치를 의미한다. Torus Ring에서는 계층 링에서와 같이 명시적인 전역 링이 존재하는 것은 아니지만, 검은색의 스위치들이 인접한 지역 링을 연결하고 있으므로 이들을 편의상 전역 링 스위치라고 부르고, 이 스위치들을 통한 지역 링 간의 통신도 전역 링을 통한 통신이라고 지칭하도록 하겠다. Torus Ring도 계층 링과 마찬가지로 동일한 수의 노드를 가진 시스템이라도 여러 가지 방법으로 구성할 수 있는데, 이 구성을 역시 $m \times n$(지역 링의 개수 m, 지역 링 내의 노드 수 n)으로 지칭한다. 즉 그림 3의 경우 8×2 Torus Ring이라고 할 수 있다.

Torus Ring은 기본적으로는 전체 노드를 몇 개의 지역 링으로 나누어 이 지역 링을 연결하는 링을 구성한다는 점에서 계층 링과 비슷하다고 할 수 있다. 하지만 여러 지역 링을 연결하는 방법에 있어서, 계층 링이 명시적으로 전역 링을 만들어 연결한다면 Torus Ring은 지역 링의 링크를 전역 링이 공유한다는 것으로 볼 수 있다. 즉 계층 링에서는 패킷의 출발지 노드와 목적지 노드가 서로 다른 링에 존재할 때에 지역 링의 링크와 따로 존재하는 전역 링의 링크를 거쳐 지나가야 하지만, Torus Ring에서는 전역 링의 링크가 따로 구성되지 않고 지역 링의 링크를 이용하도록, 인접한 두 개의 지역 링들끼리 전역 링 스위치를 이용해 직접 붙어 있는 구조를 취하고 있다. 이에 따라 하나의 지역 링에 전역 링 스위치가 2개씩 붙어 있는 형태가 된다.

3.2 Torus Ring의 특징

Torus Ring의 특징을 알아보기 위해 Torus Ring이 일반적인 계층 링보다 성능이 좋을 수 있는 경우와 그렇지 않은 경우를 나누어 생각해 보도록 하겠다. 혼란을 막기 위해 아래의 설명은 노드의 배치 방법이 각각 계층 링의 경우 그림 2, Torus Ring의 경우 그림 3과 같다고 가정하고 있다.

먼저 Torus Ring이 계층 링보다 성능이 좋을 수 있는 경우는 인접한 지역 링간에서 통신이 이루어지는 경우이다. Torus Ring이나 계층 링에서 각각의 지역 링은 2개씩의 인접한 지역 링을 가지는데, 우리는 각각의 그림에서와 같이 단방향 링을 가정하여 사용하고 있으므로 전역 링의 패킷 전송 방향에 따라 이들을 순방향 인접 링, 역방향 인접 링으로 구분하여 부르기로 한다.

Torus Ring이 계층 링보다 성능상에서 유리한 경우로서의 지역 링을 건너가는 데에는 두가지 경우가 있다. 먼저, 0 홉이 걸리는 경우는 역방향 인접 링이 목적지 지역 링인 경우이고, 1 홉이 걸리는 경우는 순방향 인접 링이 목적지 지역 링인 경우이다. 한편, 계층 링에서 1 홉이 걸리는 경우는 순방향 인접 링이 목적지 지역 링인 경우이고, $m-1$ 홉이 걸리는 경우는 역방향 인접 링이 목적지 지역 링인 경우이다. 따라서 목적지 지역 링이 순방향 인접 링인 경우에는 사실 Torus Ring이 계층 링에서보다 홉 수가 1 홉 더 많다. 이것은 두 연결 구조 모두 전역 링을 사용해 지역 링을 바꾸는 데는 동일하게 1 홉이 걸리지만 Torus Ring은 목적지 지역 링에 도착하고 나서 또 하나의 전역 링 스위치를 지나야 하기 때문이다. 예컨대 노드 0에서 노드 2로 전송되는 패킷의 경우 Torus Ring에서는 0→1→g0→g1→g2→2의 5 홉, 계층 링에서는 0→1→g0→g1→2의 4 홉이 걸리게 된다. 하지만 목적지 지역 링이 역방향 인접 링인 경우에는 Torus Ring은 계층 링과 비교했을 때 $m-1$ 홉의 이득을 그대로 볼 수 있다. 그림 3과 같은 Torus Ring의 구성상 역방향 인접 링에서는 목적지 지역 링에 도착한 후에는 다시 전역 링 스위치를 지나지 않기 때문이다. 노드 0에서 노드 14로 전송하는 패킷의 경우가 이에 해당한다고 할 수 있다. Torus Ring에서는 0→1→g0→14의 3 홉, 계층 링에서는 0→1→g0→g1→...→g7→14의 10 홉이 걸리게 됨을 알 수 있다.

Torus Ring이 계층 링보다 성능이 나쁜 경우는 지역 링 내에서 통신이 이루어 지는 경우나 인접하지 않은 목적지 링으로 패킷을 보내야 하는 경우이다. 지역 링 내에서의 통신의 경우에는 계층 링에서는 1개의 전역 링 스위치만 통과하면 되지만 Torus Ring에서는 2개의 전역 링 스위치를 통과해야 하기 때문에 적어도 1 홉이 반드시 추가될 수 밖에 없다. 또한 목적지 지역 링이 출발지 지역 링과 인접하지 않을 때, 목적지 지역 링 도착 이후 또 하나의 전역 링 스위치를 거쳐야 하기 때문에 Torus Ring이 계층 링에 비해서 1 홉씩 손해를 보게 된다.

이렇듯 Torus Ring은 계층 링에 비해 장점을 가질 수 있는 경우도 있으나 성능상 손해를 가져오는 경우도 있는데, 연결망에서의 통신의 특성을 고려해 보면 Torus Ring이 계층 링에 비해서 사실상 가져오는 이득은 더욱 커질 수 있다. 다중 프로세서를 연결하는 내부 연결망의 프로토콜들의 특성상 대부분의 경우 요청이 보내지면 그에 따른 응답이 최초의 요청 노드로 보내지는데, 이 경우 인접한 지역 링간의 통신에서 순방향 목적지 지역 링은 역방향 목적지 지역 링으로, 역방향 목적지 지역 링은 순방향 목적지 지역 링으로 바뀌게 되므로 이 요청-응답을 한 묶음으로 생각해서 계산한다면 평균적으로 $(m-1)+(-1)=m-2$ 홉의 이득을 보게 된다고 할 수 있다. 또한 인접하지 않은 링 간의 통신에서 발생하는 계층 링에 비한 손해는 1 홉으로 고정적이어서 지역 링의 크기가 커질수록 상대적으로 작아지게 되는데, 이것은 실험을 통해서 재차 확인 할 수 있다. 한편 패킷의 홉 수의 균등분포(Uniform Distribution)를 가정했을 때 Torus Ring의 계층 링에 대한 평균적인 홉 수의 이익 및 손해의 기대값은 아래와 같다.

$$(m-1)\times\frac{1}{m}-1\times(1-\frac{1}{m})=0$$

즉 균등분포일 경우에도 Torus Ring은 홉 수에서 손해를 보는 것이 아니며, 실제 병렬 프로그램이 수행될 때에는 지역성(Locality)의 성질에 의해 가까운 인접 링에 대한 통신의 비

율이 $1/m$보다는 클 가능성이 높다. 따라서 홉 수에서 더 많은 이익을 기대해 볼 수 있다.

4. 성능 평가

4.1 모의실험 환경

본 논문에서의 모의실험은 구동 기반(Execution-Driven) 시뮬레이터인 RSIM[5]으로 진행되었다. 모의 실험에 사용된 벤치마크 프로그램은 SPLASH/SPLASH-2[6]에서 임의로 추출한 5개의 프로그램이다. 모의실험에 사용된 인자는 표 1과 같다.

표 1. 모의실험 인자

인자	값
노드 수	16, 32
한 노드내의 프로세서 수	1
프로세서 클럭 속도	300 MHz
Level 1 캐쉬의 크기 및 접근시간	16 KB, 1 cycle
Level 2 캐쉬의 크기 및 접근시간	64 KB, 3 cycle
연결 망 클럭 속도	150 Mhz
Flit 크기	8 byte
Flit 접근 지연 시간	4 cycle
연결망 스위치 버퍼 크기	64 filt
메모리 접근 지연 시간	18 cycle

4.2 모의실험 결과

그림 4에서 그림 5는 노드 수 별로 지역 링 개수에 따른 계층 링과 Torus Ring의 수행 시간을 보여주고 있다. 여기에서 수행 시간은, 프로그램 전체의 수행 시간을 단일 링에서 해당 프로그램을 수행했을 때의 프로세서 클럭 수에 정규화 시켜서 나타낸 것을 의미한다. 그림을 보면 동일한 구성 내에서 계층 링보다 Torus Ring이 대부분의 경우에 더 좋은 성능을 보이고 있다. 성능이 향상된 경우 동일 구성의 계층 링 대비 최대 7% 정도의 이익을 볼 수 있음을 알 수 있다. 계층 링의 구성과 성능과의 관계에 있어서는, 16 노드의 경우 링의 형태가 8×2 보다는 4×4 형태가, 32 노드의 경우 *water*를 제외한 나머지 경우에 있어서 4×8 형태가 8×4 형태나 16×2 형태에 비해 우수한 결과를 보이는 것을 알 수 있다.

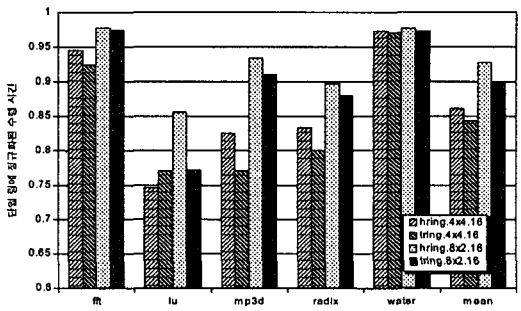

그림 4. 계층 링과 Torus Ring의 수행 시간 (16노드)

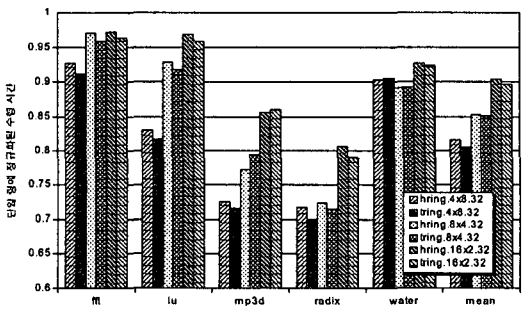

그림 5. 계층 링과 Torus Ring의 수행 시간 (32노드)

5. 결 론

본 논문에서는 계층 링과 형태는 비슷하지만 지역 링을 연결하는 방법에 있어서 차이를 둔 Torus Ring이라는 상호 연결망 형태(Topology)를 제시하였다. Torus Ring은 지역 링의 연결에 따로 전역 링을 두어 링크를 구성하는 계층 링과는 달리 지역 링의 링크를 곧바로 인접한 두 지역 링과의 연결에 활용하는 상호 연결망의 형태이다. 이 연결망에서는 역방향의 인접한 지역 링에 접근하는 경우의 홉 수가 계층 링에 비해서 $m-1$ 만큼 감소한다. Torus Ring이 지역 링보다 홉 수가 적을 수 있는 경우의 확률은 $1/m$ 정도이고 실제로 벤치마크 프로그램을 수행하였을 때 이에 따른 성능 향상을 관찰할 수 있었다. 또한 Torus Ring에서의 응답 지연 시간이 줄어듦으로 인해 프로세서의 전체 수행 시간을 감소시킬 수 있음을 실험을 통해서 확인하였다. 향후 성능 향상을 보이지 않는 일부 벤치마크 프로그램의 지역성을 잘 활용할 수 있도록 Torus Ring 연결망 내에 노드를 효율적으로 배치하는 방법이나, 전역 링 스위치의 이용률을 낮출 수 있는 방안에 대한 연구가 필요할 것으로 보인다.

참고 문헌

[1] David E. Culler and Jaswinder Pal Singh with Anoop Gupta, "Parallel Computer Architecture : A Hardware/Software Approach", Morgan Kaufmann Publishers, Inc, 1998

[2] Sung Woo Chung, Seong Tae Jhang and Chu Shik Jhon, "PANDA: ring-based multiprocessor system using new snooping protocol", Proceedings of International Conference on Parallel and Distributed Systems, pp.10-17, Dec 1998

[3] Byoung Soon Jang, Sung Woo Chung, Seong Tae Jhang and Chu Shik Jhon, "Efficient Schemes to Scale the Interconnection Network Bandwidth in a Ring-based Multiprocessor System", SAC-2001(16th ACM Symposium on Applied Computing), Las Vegas, United States, pp.510-516, March 2001

[4] G. Ravindran and M. Stumm, "A performance comparison of hierarchical ring- and mesh-connected multiprocessor networks", Third International Symposium on High-Performance Computer Architecture, pp. 58-69, Feb 1997

[5] Vijai S. Pai, Parthasarathy Ranganathan and Sarita V. Adve, "RSIM Reference Manual", Dept. of Electrical and Computer Engineering, Rice University, Technical Report 9705, 1997

[6] Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations", Proceedings of the 22nd International Symposium on Computer Architecture, 1995