

Hot Spot 이 Interconnection Network 의 성능에 미치는 영향

○ * * * * *
 김성중, 김태형, 이영노, 신인철

*단국대학교 전자공학과, **서울 산업대학 전자공학과

Effect of Hot Spot to Performance of Interconnection Network

* * * * *
 Seong-Jong Kim, Tae-Hyeong Keem, Young-No Lee, In-Chul Shin
 *Dan-Kook University, **National Seoul Polytechnic University

ABSTRACT

Interconnection network is to provide communication among functional modules. The interconnections considered are Generalized Cube networks. Two situations are examined: a memory module is equally likely to be addressed by a processor and a processor has a favorite memory.

This paper proposes the effective condition of operation in interconnection network through performance evaluation by simulation.

I. 서 본

최근 과학 기술의 급격한 발달은 정보화 시대를 요구하며, 이러한 정보화 시대로의 발전은 처리해야 할 정보량의 급속한 증가와 각종 설비에 필요한 과학 시뮬레이션, 실시간 처리를 위한 고속의 과학 계산 능력 등이 필요하게 되었고 이러한 요구의 해결 방안으로 다수의 프로세서로 구성되는, 그리고 고속 연산 능력과 경제적 타당성을 갖는 병렬 처리 컴퓨터의 개발이 필연적으로 되어 가고 있다. [1-4]

병렬 처리 컴퓨터에서 많은 처리 소자와 메모리 또는 주변 장치와의 원활한 정보 교환은 전체 시스템의 성능을 좌우하는 중요한 요소가 된다. 이들 정보 교환은 각 기능 소자 (functional module) 를 상호 연결 시키는 상호 접속망 (interconnection network; IN) 을 통하여 이루어 진다.

본 연구에서는 PE-Memory 구성의 tightly coupled MIMD 시스템에서 프로세서가 동일한 확률로

메모리 모듈을 요구하는 경우와 특정 메모리 모듈을 요구하는 확률이 더 높은 hot spot 의 경우를 시뮬레이션을 통하여 검토하여 IN 의 효율적인 운용 조건을 제안한다.

II. Multiprocessor System

다중 프로세서 시스템은 동일한 또는 서로 다른 처리 기능을 가진 여러개의 PE (processing element) 가 동시에 서로 다른 subtask 를 실행하는 공간적 병렬성을 이용하여 연산 속도를 향상시킨다. 다중 프로세서 시스템의 각 PE 는 그 자체가 여러개의 기능 유니트를 가질 수 있으며 각 PE 는 필요에 따라 상호 정보 교환을 행한다.

Michael J. Flynn 은 컴퓨터 구조를 데이터 스트림과 명령 스트림의 다중성에 따라 다음과 같이 분류하였다. [1] 스트림 (stream) 이란 명령 (instruction) 이나 데이터의 시퀀스 (sequence) 를 나타낸다.

- 1) SISD System ;
 (Single Instruction Stream - Single Data Stream)
- 2) SIMD System ;
 (Single Instruction Stream - Multiple Data Stream)
- 3) MISD System ;
 (Multiple Instruction Stream - Single Data Stream)
- 4) MIMD System ;
 (Multiple Instruction Stream-Multiple Data Stream)

MIMD 시스템은 모든 PE 가 하나의 integrated OS 하에서 동작하기는 하나 각기 다른 프로그램 시퀀스와

데이터로 독립적인 연산을 실행하며 상호 정보 교환을 함으로서 하나의 task 에 관한 처리 속도를 향상시키며 각 PE 는 고유의 CU (Control Unit) 를 갖는다.

이 시스템은 사용 알고리즘의 효율성이 PE 의 수 및 알고리즘의 vectorization 에 민감하지 않아 응용 분야의 유연성이 높으며 SIMD mode 로도 사용이 가능하므로 유용한 다중 프로세서 시스템이 된다. [7]

MIMD 구조는 다시 시스템의 구조상 PE 사이의 정보 교환 방식에 따라 다음과 같이 구분한다.

Loosely coupled 시스템; 각 PE 가 큰 국부 메모리를 가지고 데이터 전송 시스템을 통하여 정보 교환을 행하며 task 사이의 정보 교환이 적을때 유리하다.

Tightly coupled 시스템; shared 메모리를 통하여 정보 교환을 행하며 다중 포트 메모리 또는 적절한 상호 접속망을 통하여 PE 와 메모리 사이의 접속이 필요하다.

III. 상호 접속망

상호 접속망이란 공간적으로 분리되어 있는 각 기능 모듈을 상호 접속하여 정보 교환을 할 수 있는 경로를 의미한다. 특히 컴퓨터 시스템의 특성상 PE 와 PE 사이 또는 PE 와 메모리 사이의 통신을 위한 경로를 의미한다. 일반적으로 다중 프로세서 시스템의 성능은 사용하는 PE 의 수에 비례하지 않으며 그 가장 큰 이유는 PE 에 충분한 속도의 정보 통신을 제공하지 못하는 IN 때문이다. [1-3, 8]

다중 프로세서 시스템에서 상호 접속망의 구조는 다음의 몇가지로 나눌 수 있다.

- 1) 공용 버스 (Time Shared or Common Bus)
- 2) 크로스바 스위치 (Crossbar Switch)
- 3) 다중 포트 메모리 (Multiport Memory)
- 4) 다단 상호 접속망 (Multistage

Interconnection Network; MIN)

MIN 은 하드웨어의 복잡성을 감소시키며 충분한 정보 교환 능력을 갖는 IN 으로서 대규모 멀티

프로세서 시스템에서 크로스바에 비해 경제적이며 모듈화 할 수 있고 제어하기 쉬우며 또한 구현하는 방법에 따라 크로스바에 버금가는 성능을 얻을 수 있기 때문에 널리 연구되고 있다. [9-11]

MIN 은 topology, 요구되는 기능, 제어 방식 등에 따라 여러가지 종류의 MIN 이 제안되고 있다. 이들은 모두 몇개의 단 (stage) 과 각 단의 스위치 소자를 접속하는 연결 링크로 구성되어 MIN 의 입출력 포트 사이의 실제 통신 경로를 구성한다. 특히 2x2 스위치 소자를 사용하는 단일 경로 저지형 (single path blocking) MIN 이 집중적으로 연구되고 있으며 N 을 PE 의 수라할 때 단의 수는 $\log_2 N$, 각 단의 스위치 소자의 수는 N/2 이 되어 스위치 소자의 수가 크로스바 스위치의 $O(N^2)$ 에 비해 $O(N \log_2 N)$ 으로 되어 PE 의 수가 많아 질수록 유리하며 동일한 스위치 모듈의 반복으로 VLSI화 예도 유리하다. 이때 스위치 소자는 프로토콜에 따라 경로를 설정하고 conflict 및 contention 이 발생할 때 처리할 수 있는 능력등을 가져야 한다.

Generalized Cube IN 에서의 연결은 큐브 상호 접속함수에 기초를 둔다. 큐브 상호 접속함수는 다음과 같이 정의된다.

$$Cube_i(P_{n-1} \dots P_1 P_0) = P_{n-1} \dots \bar{P}_{i+1} P_i P_{i-1} \dots P_1 P_0$$

여기에서 $0 \leq i < N$, $0 \leq P < N$, \bar{P}_i 는 P_i 의 보수이다.

그림 1 에 8x8 Generalized Cube Network 을 도시하였다.

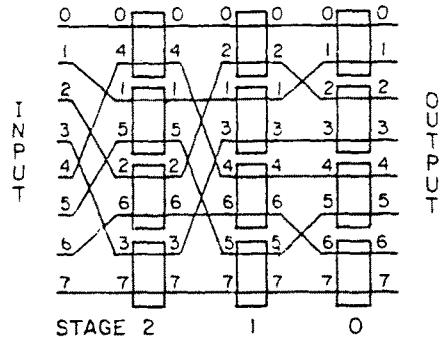


그림 1 . 8x8 Generalized Cube Network

IV. Simulation

시뮬레이션을 위하여 두가지 상황을 제안한다. 각 메모리 모듈이 프로세서에 의해 선택되는 비율이 같은 경우가 그 하나이고 다른 하나는 각 프로세서가 favorite memory 를 갖는 경우이다. 특히 후자의 경우 hot spot 으로 설명될 수 있는데 만약 processor i (P_i) 가 memory module i (MM_i) 를 선택하는 확률이 다른 확률 보다 높다면 MM_i 의 favorite processor 는 P_i 이다.

Processor P_i 가 memory module MM_i 를 선택하는 확률을 m 으로 놓으면 전자의 경우 m 은 $m = 1/N$ 이고 hot spot 인 경우 m 은 $m > 1/N$ 이다. [5,6] 모든 스위칭 소자들의 경로가 straight 로 연결될때 각 프로세서들이 고유의 favorite memory 를 갖는다고 가정하며 그 예를 그림 1 에 도시하였다.

상호 접속망의 성능 평가 factor 로는 request service time (RST) 과 memory bandwidth (MBW) 가 조사 되었으며 이 두가지는 IN 의 성능을 정량적으로 평가 비교하는 요소로 널리 쓰이고 있다.

시뮬레이션은 PE 의 수가 $N = 8, 16$ 의 경우에 대하여 행하였으며 성능평가 요소를 데이터 전송 시간 (D), 메모리 요구 확률 (R) 의 변화와 conflict 처리 방식을 MIN 의 각 단계에서 다르게 취함에 따라 검사하였다. 클럭 cycle time 의 반복 회수는 3000 회로 하였고 transition 효과를 없애기 위하여 초기 100 회의 결과는 데이터의 축적에서 제외하였다.

시뮬레이션을 위하여 다음의 가정을 하였다.

- 1) 메모리 요구에 의한 경로 설정은 매 클럭당 MIN 의 1 단씩 진행된다.
- 2) 경로가 sender (PE) 와 receiver (메모리) 사이에 설정되면 그 완성된 경로는 일정시간 (데이터 전송 시간 = D 클럭 cycle) 동안 정보 전송을 위하여 유지된다.
- 3) 일단 정보 전송이 완료된후 PE 에서 다시 새로운 요구가 발생될 때 까지의 시간은 일정하지 않으며 메모리 요구 확률 (request issue ratio; R) 로서 결정된다. 즉 R 은 PE 가 idle 일때 매 클럭 마다 새로운 요구를

발생시킬 확률이 된다.

- 4) 스위치 소자의 두 입력에 요구가 동시에 도착하여 동일한 출력 포트를 요구하는 contention 이 발생하는 경우에는 임의선택 우선 순위로서 랜덤하게 한 요구가 선택된다.
- 5) conflict 또는 contention 에 의하여 경로 설정에 실패한 저지된 요구는 약속된 방법에 의하여 각 단계에서 hold 또는 drop 방식으로 처리된다.

그림 2에는 시뮬레이션 프로그램의 흐름도를 도시하였다.

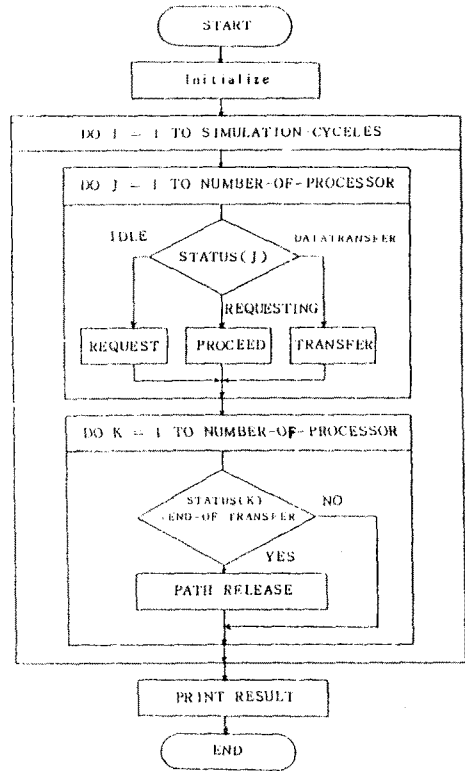


그림 2. 시뮬레이션 프로그램의 흐름도

V. 결과 및 결론

$N = 16$ 일 경우, 두번째 단부터 Hold model로 conflict를 처리한 Mixed 방식 (16-1)의 simulation 결과를 표 1에 정리했다. Hot Spot 확률이 높아질 수록 conflict 발생 확률은 낮아지며 정규화 RST 측면에서 성능은 향상된다.

표 1. N = 16 인 경우 Mixed 방식 Simulation 결과

R	D	정규화 RST			
		Uniform		Hot Spot	
		m(1/N)	m(25%)	m(50%)	m(75%)
0.6	4	1.716	1.627	1.535	1.320
	10	2.021	1.896	1.776	1.479
	16	2.128	1.994	1.910	1.554
1.0	4	1.667	1.641	1.575	1.395
	10	1.957	1.876	1.769	1.517
	16	2.086	1.987	1.856	1.583

표 2. N = 16 인 경우 Simulation 결과

R	D	정규화 RST (m = 50%)			
		Hold	16-1	16-2	Drop
0.6	4	1.558	1.535	1.561	1.623
	10	1.851	1.776	1.754	1.772
	16	2.019	1.910	1.806	1.900

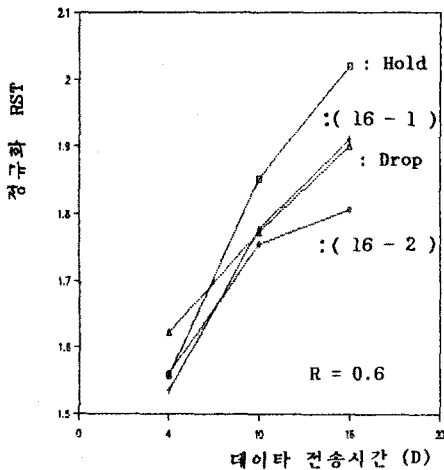


그림 3. 정규화 Request Service Time

표 3. N = 16 인 경우 Simulation 결과

R	D	정규화 RST (m = 50%)			
		Hold	16-1	16-2	Drop
0.2	10	2.018	1.971	1.985	2.069
0.6		1.851	1.776	1.754	1.772
1		1.792	1.769	1.775	1.792

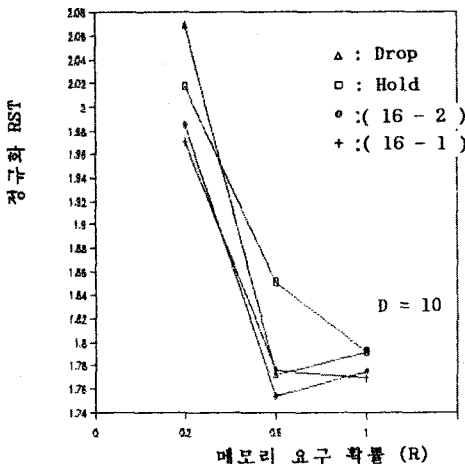


그림 4. 정규화 Request Service Time

표 2 와 3, 그림 3 과 4 에는 N=16이고 m이 50%인 경우에 각 단에서의 conflict처리를 다르게 해준 simulation 결과이다. 데이터 전송시간 (D)가 짧을때는 앞단부터 Hold 모델로 conflict를 처리해 주는 것이 유리하며 D가 길어질수록 (D>10) 세번째단부터 Hold 모델로 conflict를 처리해 주는 것이 가장 유리하다. 메모리 요구확률 (R)의 변화에 따른 conflict 처리 방법은 Mixed model 이 효율적이나 R이 약 0.6 보다 커질 경우에는 Drop model 과 세번째 단부터 Hold model로 conflict를 처리한 Mixed 방식 (16-2)을 제외한 다른 model 들의 정규화 RST 는 증가 하였다.

이는 상호 접속망에서 R 과 D 의 변화에 따른 효율적인 운영 조건을 결정하는데 참고 자료가 될 수 있다.

참고 문헌

- [1] K. Hwang and F. A. Briggs, "Computer Architecture and Parallel Processing," McGraw Hill, 1984.
- [2] H. J. Siegal, "Interconnection Network for Large Scale Parallel Processing," Lexington Books, 1985.
- [3] T. Feng, "A Survey of Interconnection Networks," Computer, Dec., 1981, pp.12-27.
- [4] R. J. Mc-Millen, "A study of multistage Interconnection Network," Ph. D. dissertation, Purdue University, 1982.
- [5] Laxmi N. Bhugan, "An Analysis of Processor-Memory Interconnection Networks," IEBB Trans. Comp., Mar., 1985, pp.279-283.
- [6] Laxmi N. Bhugan and C. W. Lee, "An Interference Analysis of Interconnection Networks," Proc. of the 1983 Inter. Conf. on Parallel Processing, Aug., 1983, pp.2-9.
- [7] H. Stone, "Introduction to Computer Architecture," Computer Science Press, 1980.
- [8] U. V. Premkumar, et al., "Design and Implementation of the Banyan Interconnection Network in TRAC," The Proceedings of the National Computer Conference, 1980, pp.643-653.
- [9] C. Wu and T. Feng, "On a Class of Multistage Interconnection Networks," IEBB Trans. Comp., Aug., 1980, pp.694-702.
- [10] H. Stone, "Parallel Processing with the Perfect Shuffle," IEBB Trans. Comp., Feb., 1971, pp.153-161.
- [11] L. R. Goke and G. J. Liporski, "Baryan Networks for Partitioning Multiprocessing Systems," Proc. 1st Annual Comp. Arch. Conf., Dec., 1973, pp.21-28.
- [12] D. H. Lawrie, "Access and Alignment of Data in an Array Processor," IEBB Trans. Comp. Dec., 1975, pp.1145-1155.
- [13] K. E. Batcher, "The FLIP Network in STARAN," Proc. 1976, Int'l Conf. on Parallel Processing, Aug., 1976, pp.65-71.
- [14] M. C. Pease, "The Indirect binary n-cube Microprocessor Array," IEBB Trans. on Computer, May, 1977.