

# 단방향 이중연결 CC-NUMA 시스템의 동적 부하 대응 경로 설정 기법

서 호 중<sup>†</sup>

요 약

높은 클럭으로 동작하는 고속의 프로세서를 다수 이용한 다중프로세서 시스템 성능은 프로세서 자체의 성능보다 상호연결망의 트랜잭션 처리 능력 및 지연에 의하여 큰 영향을 받게 된다. 따라서 상호연결망의 성능은 대역폭 및 지연시간 측면으로 시스템 성능에 큰 비중을 차지한다. 단방향 이중 연결을 이용한 CC-NUMA 구조는 이중 연결을 이용한 대역폭 증가효과와 고속 단방향 링크를 이용한 적은 지연시간으로 인하여 고성능 시스템에서 많이 채용되고 있다. 한편, 이중 연결구조로 인하여 시스템의 상호연결망의 최단 경로는 단일하게 형성되지 않으며, 여러 개의 최단 경로가 구성될 수 있다. 그러나 실제 응용프로그램을 수행할 때, 동일한 홉 수를 나타내는 경로일지라도 하더라도 각 연결 링크의 부하 및 경쟁에 따른 지연 시간의 차이를 나타내게 되며, 만일 노드 간의 트랜잭션 전달 경로가 정적으로 구성되어 있을 경우, 실제 프로그램의 수행에서 균일하지 못한 연결 링크 부하에 따른 지연 시간의 차이가 나타날 수 있음을 의미한다. 이는 곧 고속의 상호연결망 전체의 대역폭을 균일하게 사용하지 못함으로 나타나는 부가적 지연 시간으로 볼 수 있으며, 이로 인한 응용 프로그램의 수행 성능이 저하될 수 있음을 의미한다.

본 논문은 기존 연구된 단방향 이중 연결을 이용한 CC-NUMA 시스템에서, 노드간 트랜잭션 전달 경로가 정적으로 구성될 경우 발생할 수 있는 성능 저하를 평가하고, 정적 경로와 동일한 홉 수의 경로를 나타내며 링크 부하에 따라서 동적으로 전달되도록 부하에 따르는 동적 경로 설정 방법을 제시하였다. 논문에서 제시하는 방법은 기존 경로설정 방법에 대하여 동일한 홉 수를 나타내며, 링크 부하에 따라서 동적으로 경로를 설정함으로써 실시간 경로 분배가 자연스럽게 이루어지도록 하였고, 링크 경쟁을 완화함으로써 보다 균일한 링크 사용을 나타냈고, 링크 획득 실패로 인한 지연시간을 감소시켰다. 프로그램 구동 시뮬레이션을 통한 성능 검증 결과, 논문에서 제시한 동적경로 설정 방법은 기존 정적 경로 설정 방법에 비해 링크 점유시간 편차가 1~10% 낮게 나타났고, 링크의 획득 실패 횟수가 1~3%감소하였으며, 그 결과 1~6%의 수행 시간 감소를 나타냈다.

키워드 : 이중 연결 CC-NUMA 시스템, 최적 경로, 동적 경로 설정, 부하 분배

## Load Balancing of Unidirectional Dual-link CC-NUMA System Using Dynamic Routing Method

Suh, Hyo-Joon<sup>†</sup>

ABSTRACT

Throughput and latency of interconnection network are important factors of the performance of multiprocessor systems. The dual-link CC-NUMA architecture using point-to-point unidirectional link is one of the popular structures in high-end commercial systems. In terms of optimal path between nodes, several paths exist with the optimal hop count by its native multi-path structure. Furthermore, transaction latency between nodes is affected by congestion of links on the transaction path. Hence the transaction latency may get worse if the transactions make a hot spot on some links.

In this paper, I propose a dynamic transaction routing algorithm that maintains the balanced link utilization with the optimal path length, and I compare the performance with the fixed path method on the dual-link CC-NUMA systems. By the proposed method, the link competition is alleviated by the real-time path selection, and consequently, dynamic transaction algorithm shows a better performance. The program-driven simulation results show 1~10% improved fluctuation of link utilization, 1~3% enhanced acquirement of link, and 1~6% improved system performance.

Key Words : Dual-link Interconnection Network CC-NUMA System, Transaction Path, Dynamic Routing, Load Balancing

### 1. 서 론

단일 프로세서를 고속화 하여 성능 향상을 얻고자 하는

노력에 반하여 여러 프로세서를 이용하고, 프로그램의 병렬 성능을 높임으로서 고성능을 얻고자 하는 다중 프로세서 시스템은 고성능 시스템의 주류로 자리 잡혀 있다[1]. 최근에는 점차 고속화, 고성능화 되어가는 마이크로프로세서의 발달에 따라 고속의 마이크로프로세서를 이용하여 노드를 구성

※ 본 연구는 2005년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

† 종신회원 : 가톨릭대학교 컴퓨터정보공학부 조교수

논문접수 : 2005년 7월 6일, 심사완료 : 2005년 10월 7일

하고 수 개 이하의 프로세서로 구성된 노드를 고속의 상호 연결망을 이용하여 구성된 시스템이 다수 사용되고 있다. 이렇게 GHz 이상의 클럭 속도로 동작하는 고속의 마이크로 프로세서에 비하여 상대적으로 저속으로 동작하는 상호연결망의 대역폭 및 접근 속도는 응용 프로그램의 처리에 더욱 큰 성능저하 요소로 대두되었고[2], 특히 공유메모리 방식의 다중프로세서 시스템의 경우 메모리 접근의 지연 시간에 의한 성능 저하를 줄이는 것이 성능 개선의 중요한 목표가 되었다. CC-NUMA(Cache Coherent Non Uniform Memory Access) 구조[3]는 이러한 측면에서 공유 메모리를 분산하고 각 노드 내에 지역메모리를 포함시킴으로써 메모리 접근을 지역화하여 높은 성능을 얻고자 하는 방법으로써 최근 고성능 시스템에 다수 채용되어 있다. 그러나 CC-NUMA 구조의 시스템은 지역 메모리 접근은 상호연결망을 접근하지 않음에 비하여 원격 메모리의 접근은 상호연결망을 거쳐야 하므로 상대적으로 상호연결망의 접근 지연 시간에 의한 원격 메모리 접근의 병목 및 지연이 크게 나타나게 된다.

CC-NUMA 시스템에 있어서 상호연결망의 중요성 및 구조는 이러한 이유로 강조되어 왔으며, 현재 각 노드를 연결하는 상호연결망으로 Fast Ethernet, Gigabit Ethernet, Myrinet, SCI(Scalable Coherent Interface)[4] 등이 채용되고 있다. 이들 중 특히 SCI의 경우 1GB/s의 대역폭을 지원하고 점대점 연결 및 스위치 구조를 사용하고 있으므로 고속의 점대점 통신을 통한 대역폭 및 지연 시간의 고성능화에 다수 채용되고 있다. 점대점 연결을 이용한 CC-NUMA 시스템의 경우, IBM의 NUMA-Q[5], IBM의 최신 메인프레임 시스템인 zSeries 990[6], Data General의 AViiON 시스템[7] 등이 산업계에서 적용되었고, 서울대학교의 PANDA 시스템[8] 등 학계에서도 활발히 연구되고 있다.

이처럼 고속의 이중 점대점 연결을 이용하여 CC-NUMA 시스템을 구현할 경우, 높은 대역폭과 고성능 연결구조의 이점을 활용할 수 있어 고성능 시스템에 다수 채용되고 있으나, 노드간 연결 경로가 다수 발생하게 됨으로써 여러 개의 최단경로가 발생하게 된다. 노드간 트랜잭션의 전송 경로가 다수 발생할 수 있는 경우, 트랜잭션의 경로 설정은 일정 알고리즘을 통하여 최소의 홉 수(hop count)와 균일한 노드 부하가 발생하도록 잘 설계되어야 한다. 이는 노드간 링크의 경쟁에 따른 시간 지연을 최소화하기 위한 것으로, 트랜잭션 처리 시간에 있어서 정적인 요소인 상호연결망의 물리적 성능과 더불어 노드의 경쟁에 따르는 대기 시간이 트랜잭션 처리 시간에 포함되기 때문이다[2]. 또한 고속의 링크에 원활한 데이터 전달을 위해서는 트랜잭션의 경로 설정 알고리즘이 고속으로 처리될 수 있어야 하며, 만일 일정 수준 이상으로 복잡한 알고리즘을 적용할 경우 알고리즘 자체의 처리를 위한 시간 지연이 트랜잭션 처리 시간에 영향을 주게 된다. 이와 같은 이유로 고속의 경로 설정을 위하여 정적인 최단 경로설정을 통하여 트랜잭션의 전송이 연구된 바 있다[9, 10, 11]. 그러나 단순한 정적 알고리즘이 사용될 경우 알고리즘 처리 시간의 지연은 발생하지 않으나, 시

간에 따른 응용의 특성 변화에 따라 특정 노드간 불균일한 트랜잭션이 발생하는 경우 시스템의 노드간 연결 링크의 부하 불균형을 초래할 수 있으며, 이는 곧 노드 경쟁에 따르는 대기 시간 발생에 따라서 트랜잭션 처리 시간이 길어질 수 있음을 의미한다.

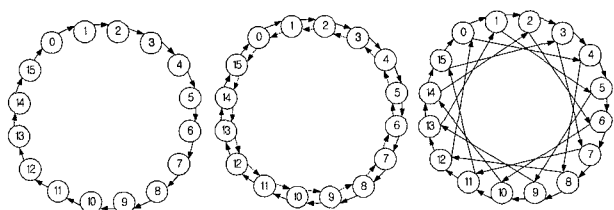
본 논문은 이중 연결 구조를 이용한 CC-NUMA 시스템에서 노드간의 전달경로가 다수 발생하는 경우, 응용 프로그램의 동적 특성에 의하여 노드간 부하 분배가 불균일하게 나타날 수 있고, 이로 인하여 시스템 성능이 저하될 수 있음을 착안하여, 기존에 연구된 정적 경로를 이용하는 방법에 대비하여 노드의 부하에 따라 전송 경로를 선택하면서 정적 경로와 같은 물리적 거리의 경로를 따라 트랜잭션이 이루어질 수 있도록 개선된 경로 설정 방법을 제시한다. 또한 이러한 경로 설정 방법이 링크의 경쟁을 고려하면서도 충분히 단순한 방법을 사용하도록 함으로써 빠른 속도의 링크에 대응할 수 있도록 하여, 동적 경로 선택에 따른 시간 지연을 발생하지 않도록 하고, 결과적으로 정적 경로 설정 방법을 이용한 경우에 비하여 개선된 트랜잭션 전송 지연과 균등한 노드 부하 분배를 얻고자 하는 것이다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문에서 적용하는 이중 연결구조의 CC-NUMA 시스템과 트랜잭션의 전송 경로를 설명하고, 3장에서 다중 프로그램을 위한 시뮬레이션 도구와 환경을 서술하며, 4장에서 시뮬레이션 결과를 제시하고, 5장에서 결론을 맺는다.

## 2. 이중 연결 CC-NUMA 시스템과 트랜잭션 전달 경로

### 2.1 점대점 연결을 이용한 CC-NUMA 시스템

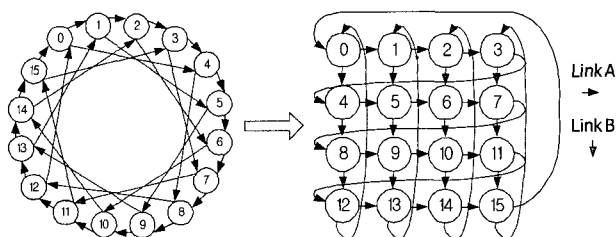
(그림 1)은 링 구조의 점대점 연결을 이용한 CC-NUMA 시스템이다. 그림 (a)와 같은 가장 단순한 구조로써 단일 연결을 이용한 링 형태의 시스템은 IBM의 NUMA-Q[5], Debois의 Express Ring[12], PANDA 시스템[8] 등이 있으며, 노드간 트랜잭션 경로는 단일하게 구성된다. 단일 연결 구조를 이용한 시스템의 경우, 노드 간 트랜잭션의 부하에 따라서 링크의 부하가 불균일하게 나타날 수 있으며, 링크에 대한 경쟁에 따른 트랜잭션 전달 지연이 나타난다. 그림 (b)는 링크를 이중으로 하고 반대 방향으로 연결한 형태로, Data General의 AViiON 시스템[7]과 PANDA-II 시스템[8], IBM의 zSeries 990[6] 등에 채용된 형태이다. 이 경우 링크의 개수가 단일 링에 비해 두 배가 되므로 링크의 대역폭이 두 배로 되고, 링크에 대한 경쟁이 완화된다. 그러나 트랜잭션의 최단 전달 경로는 앞의 단일 링과 마찬가지로 단일하게 구성되므로 노드간 트랜잭션에 따른 링크의 부하가 불균일하게 나타난다. 그림 (c)의 경우 건너뛰의 형태로 링크를 연결한 것으로, 트랜잭션의 전달 원천 노드와 목적 노드에 따라서 다수의 최단 경로가 구성된다. 앞서 연구된 바에 의하면, 이러한 건너뛰 링크를 이용할 경우, 노드간 트랜잭션이 비교적 분산되는 효과가 있으며, 노드간 평균 트랜잭션 전달 경로의 길이가 짧아지는 효과가 있음이 연구된 바 있다[9].



(a) 단일링크 (b) 반대방향 이중 링크 (c) 건너뛴 이중 링크  
(그림 1) 링 구조의 CC-NUMA 시스템 (16 노드)

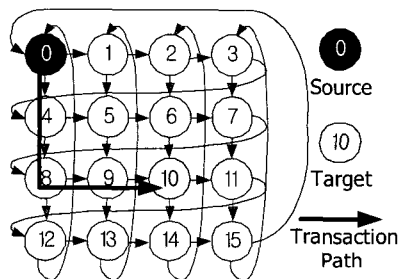
2.2 이중 연결 CC-NUMA 시스템의 트랜잭션 전달 경로

건너뛴 링크가 있는 이중 연결 CC-NUMA 시스템은 이중 연결 구조로 인하여 단방향 링크로 구성된 토로이달 메쉬(toroidal mesh)와 유사한 구조의 시스템으로 변환하여 생각해볼 수 있다. (그림 2)는 16노드로 구성되고 4 건너뛴 수를 갖는 형태의 시스템을 토로이달 메쉬 구조와 유사하게 배치한 것이다. 이러한 배열에 의하여 건너뛴 링크는 Y축 방향(Link B)으로 배치되며, 노드 번호 증가에 따라 연속적으로 연결된 링크는 X축 방향(Link A)으로 연결된다.



(그림 2) 메쉬 형태로 변환한 건너뛴 이중 링크 CC-NUMA 시스템 (16 노드, 4 건너뛴)

건너뛴 링크를 갖는 이중 연결 CC-NUMA 시스템을 이와 같이 메쉬 구조와 유사하게 재배치하여 볼 경우, 트랜잭션을 발생시킨 노드를 가장 왼쪽의 위에 위치시켜 볼 때, 앞서 제시된 논문[9]에서 트랜잭션의 전달 경로는 일단 트랜잭션의 목적 노드가 동일 행에 위치되어 있지 않을 경우, (그림 3)과 같이 Y축 방향의 건너뛴 링크를 통하여 목적 노드의 동일 행에 해당되는 노드까지 전달되며, 그 이후에는 x축 방향의 일반 링크를 통하여 트랜잭션을 전달하도록 구성되어 있다. 그러나 보는 바와 같이, 트랜잭션의 근원 노드와 목적 노드가 동일한 행 또는 동일한 열에 위치되어 있지 않을 경우, 두 노드 간의 최단 경로는 여러 개가 발생하

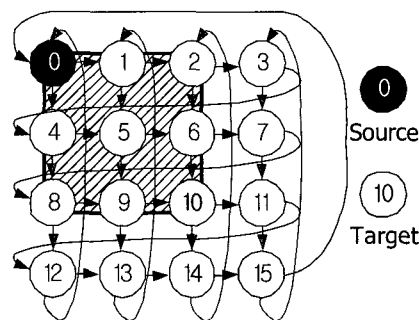


(그림 3) 정적인 트랜잭션 전달 경로 (16 노드, 4 건너뛴)

며, 노드 간 트랜잭션의 발생 빈도가 균일하지 않을 경우, 시스템 전체의 링크에 대한 경쟁이 유발할 수 있음을 알 수 있다.

2.3 동적인 부하대응 경로설정 기법

건너뛴 링크를 갖는 이중 연결 CC-NUMA 시스템에서 트랜잭션의 근원 노드를 가장 왼쪽 위에 배치하도록 하여 토로이달 메쉬 구조와 유사하게 형태로 변형하여 볼 때, 근원 노드와 목적 노드를 대각 위치로 하는 사각형 내의 링크를 경유하는 모든 경로는 최단 경로를 구성하고 있다. (그림 4)는 이와 같이 형성되는 사각형을 나타내고 있으며, 사각형 내의 모든 링크는 단방향으로 이루어져 있으므로 사각형 내의 링크를 거치는 어떠한 경로나 동일한 홉 수를 나타내게 된다.



(그림 4) 사각형 내의 트랜잭션 전달 경로 (16 노드, 4 건너뛴)

따라서 트랜잭션의 전달을 위하여 경로상의 노드는 다음과 같은 알고리즘에 따라 트랜잭션의 전달을 최단경로를 통하여 이룰 수 있다.

- 자신의 노드 번호와 트랜잭션의 목적 노드가 같은 행에 있으면 X축 방향의 링크 A로 전달
- 자신의 노드 번호와 트랜잭션의 목적 노드가 같은 열에 있으면 Y축 방향의 링크 B로 전달
- 자신의 노드 번호와 트랜잭션의 목적 노드가 같은 행, 같은 열이 아니면 x축 또는 y축 방향의 링크 중 사용하지 않는 링크로 전달, 두 링크 모두 사용중일 경우, 먼저 가용한 링크로 전달

위와 같은 단계를 이용하면 어떠한 경우나 최단 경로를 통하여 목적 노드로 전달이 이루어지게 되며, 링크의 부하에 따라 동적인 경로 설정이 이루어지게 된다. 또한 같은 행 또는 같은 열에 해당되는지의 연산은 간단한 마스크(mask) 연산을 통하여 비교할 수 있으므로 고속의 링크 속도를 저하시키지 않으면서 빠른 판단을 완료할 수 있다.

3. 시뮬레이션 방법

동시에 여러 프로세서에서 다수의 프로세스를 수행하게

되는 다중 프로세서 시스템의 시뮬레이션 도구로 본 논문에서는 Augmint를 사용하였다[13]. Augmint는 MIPS 명령을 기반으로 만들어진 다중프로세서 시뮬레이터인 Mint를 x86 명령어 프로그램을 추적할 수 있도록 변경된 시뮬레이터로써 실제 실행할 프로그램에 삽입되어 하나의 실행 파일을 생성하며, 이 실행파일을 리눅스 상에서 수행시켜 다중 프로세서를 모사함과 동시에 여러 프로세서에서 발생하는 모든 메모리 접근을 추적한다. 시뮬레이터 프로그램은 전단부와 후단부로 나뉘는데, 전단부는 다중 프로세서 시스템을 흉내내어 여러 프로세서로부터 발생하는 메모리 접근을 후단부로 전달하여 후단부에서 메모리 및 상호연결망을 구현할 수 있도록 하고, 프로그램의 후단부는 전단부에서 발생하는 모든 메모리 접근에 대하여 접근 시간 및 상호연결망에 대한 경쟁과 지연을 필요에 따라서 구현하도록 되어 있다. 본 연구에서는 후단부에 해당되는 건너뛴 링크를 갖는 이중 연결 CC-NUMA 시스템을 구현하였으며, 앞서 제시한 정적 경로설정 방법과 동적 경로설정 방법을 각각 적용하였다.

경로 설정 방법에 따른 개선 정도를 가늠하기 위하여, 공유 메모리 다중프로세서 시스템의 성능 평가에 표준적으로 사용하는 SPLASH-2 벤치마크[14] 프로그램의 일부를 이용하였으며, 해결할 문제를 총 32 개의 프로세서에 균등하게 분할하였으며, 다음 <표 1>과 같은 부하를 가지도록 다섯 개의 프로그램을 적용하였다.

시뮬레이션을 위하여 사용한 시스템은 16 노드를 설정하였으며, 각 노드 내에는 두 개씩의 프로세서를 가지고 있는 형태로 하여, 최신의 CC-NUMA 시스템과 유사하게 구성하였으며, 각 프로세서의 속도 및 링크의 속도 등 기타

<표 1> 시뮬레이션에 이용된 프로그램 및 부하

프로그램	부하
FFT	65536 데이터
LU	512x512 행렬
Radix	131072 키, 1024 기수
Barnes	16384 입자
Mp3d	test.geom. 입력, 5 스텝 진행

<표 2> 실험 대상 시스템 환경

항목	값
프로세서 클럭 속도	2 GHz
시스템의 프로세서 수	32 개
노드 당 프로세서 수	2 개
프로세서 당 캐시 크기	128 KByte
프로세서 캐시 연관	4 way
노드 개수	16 개
노드 내 버스 속도	512 MHz
노드 당 원격 캐시 크기	256 Kbyte
원격 캐시 연관	8 way
링크 전송 대역폭	2G Byte/s
캐시 교체 정책	최근 최소 사용(LRU)

사항은 다음 <표 2>와 같이 최신 프로세서의 속도 및 SCI 링크의 성능을 기준으로 하였다.

#### 4. 시뮬레이션 결과

<표 3>은 다섯 개의 응용에서 시스템 내의 모든 링크에 대해 트랜잭션의 전달로 인해 발생한 링크 점유 시간의 평균값과 표준 편차 및 링크의 평균 점유 시간에 대한 편차의 비율을 제시한 것이다.

정적 경로 설정 방법에 비하여 동적 경로 설정 방법을 적용한 경우, 모든 응용에서 링크 점유 시간의 편차 값이 크게 줄어들었다. 이 결과가 의미하는 것은, 정적 경로 설정 방법을 이용할 경우, 동적 경로 설정 방법에 비하여 링크의 점유 정도가 보다 불균일하게 나타남을 의미하며, 이로 인한 지역적 링크에 대한 경쟁이 높게 나타날 수 있음을 의미한다.

다음의 <표 4>는 트랜잭션이 전달될 때, 링크에 대한 경쟁에 의하여 링크의 획득을 실패하여 전달이 일어나지 못하고 대기가 발생한 횟수와, 두 가지 경로 설정 방법에 따른 차이를 백분율로 나타낸 것이다.

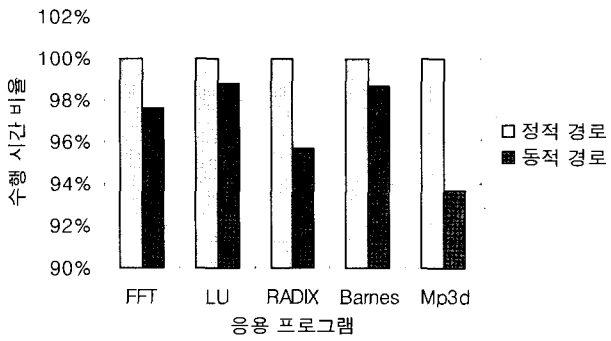
트랜잭션이 전달 경로상 링크를 획득하지 못할 경우, 그로 인한 트랜잭션 전달 지연 시간의 증가가 나타난다. 동적 경로 설정 방법은 정적 경로 설정 방법에 비하여 보다 융통성 있는 경로를 제공하게 되므로 <표 4>에서 나타난 바와 같이 링크의 획득에 실패한 횟수가 줄어들게 된다. CC-NUMA

<표 3> 평균 링크 점유 시간 및 편차 (사이클)

	정적 경로 설정		
	평균점유시간	편차	비율
FFT	16141760	1326978	8.2%
LU	116900160	35596659	30.5%
Radix	20579344	2989769	14.5%
Barnes	34497952	2610282	7.6%
Mp3d	22229968	2857866	12.9%
	동적 경로 설정		
	평균점유시간	편차	비율
FFT	16755600	1188891	7.1%
LU	123215008	25738189	20.9%
Radix	21602736	2446802	11.3%
Barnes	35197792	2106382	6.0%
Mp3d	21728320	2062455	9.5%

<표 4> 트랜잭션의 링크 획득 실패 횟수

	정적 경로 설정	동적 경로 설정	비율
FFT	131272	129757	98.8%
LU	15808062	15458348	97.8%
Radix	89322	86386	96.7%
Barnes	112499	109008	96.9%
Mp3d	105944	101920	96.2%



(그림 5) 정적 경로 방법에 대한 동적 경로 방법의 수행 시간 개선 정도

시스템에 있어서, 트랜잭션의 전달이 일어나지 못하고 대기하게 되는 경우, 노드 간 전송을 유발시키지 않는 노드 내의 트랜잭션에 비하여 훨씬 긴 시간을 소모하게 되므로, 링크의 획득 실패는 시스템 성능에 큰 영향을 주게 된다.

다음 (그림 5)는 정적 경로의 설정 방법을 이용한 경우에 대한 동적 경로 설정 방법의 응용 수행 시간의 비율을 나타낸다.

앞서 살펴본 바와 같이 정적 경로 설정 방법은 동적 경로 설정 방법에 대비하여 평균 링크 점유 시간이 링크에 대해 보다 불균일하게 나타났으며, 이로 인한 트랜잭션 전달을 위한 링크의 획득 실패 횟수가 증가되고 있다. 이는 응용 프로그램의 수행 시간에 영향을 끼치게 되며, (그림 5)에서 나타난 바와 같이 응용에 따라서 최대 6.3%, 최소 1.2%의 수행 시간의 차이를 관찰할 수 있었다. 시스템 전체의 응용 프로그램 부하에 따라서 다소간의 차이는 있겠으나 노드간 트랜잭션이 노드 내 트랜잭션에 비해 많이 발생하게 되면 동적 경로 설정 방법이 보다 유연한 대응 및 보다 적은 링크 획득 실패율을 나타내게 될 것임을 예측할 수 있다. 한편 경로 설정 방법을 비교하기 위한 응용으로 본 연구에서 채택한 응용들은 다중 프로세서 시스템에서 병렬성을 최대화할 수 있는 형태로 구성되어 있으며, 이로 인해 각 프로세서 및 노드 간 부하와 통신 분배를 균형적으로 할 수 있도록 작성된 것이다. 만일 실제의 시스템이 사용되는 경우와 같이 다수의 응용 프로그램이 동시에 불균일한 시스템 부하를 발생시키며, 노드 간 트랜잭션이 높게 나타나는 불균등한 프로그램 수행 환경이 될 경우, 본 논문에서 제시한 동적 경로 설정 방법은 정적 경로 설정 방법에 대비하여 더 큰 성능 차이를 보일 수 있을 것으로 생각된다.

### 5. 결 론

고속의 단일 프로세서의 성능을 개선시키려는 노력은 컴퓨터의 도래와 동일하게 지속적으로 이루어져 왔다. 그러나 최근 단일 프로세서의 클럭 속도는 한계에 다다르고 있으며, 상대적으로 반도체 공정 기술의 개선으로 인한 집적도의 증가는 단일 칩에 보다 많은 트랜지스터를 집적할 수 있도록 발전되었다. 고성능 워크스테이션이나 메인프레임에서 사용

하던 다중프로세서는 이제 데스크탑에서 사용하는 마이크로 프로세서의 단일 칩 위에 구현되는 정도로 시스템 고성능화의 어쩔 수 없는 선택으로 간주되고 있으며, 이와 같은 발전에 의하여 다중프로세서 시스템의 프로세서는 더 높은 상호연결망 대역폭 및 낮은 지연 시간을 요구하고 있다. 이중연결 구조의 CC-NUMA 시스템은 현재 고성능 메인프레임 등에 사용하는 분산 공유 메모리 구조의 시스템으로 노드간 링크의 대역폭 및 접근 시간에 유리한 장점을 갖추고 있으나, 각 노드의 트랜잭션 발생 빈도와 응용 프로그램 부하의 차이 등에 의해 발생하는 링크 부하의 불균형은 전체 시스템 처리 능력을 저하시킬 수 있는 단점이 존재하였다.

본 연구는 건너뛴 연결을 갖는 이중 링크 구조의 CC-NUMA 시스템에서 정적 경로 설정으로 발생하는 링크의 부하 분배 불균형이 발생할 수 있으며, 이로 인한 트랜잭션 처리 지연이 발생할 수 있음에 주목하여, 정적 경로 설정과 동일한 홉 수의 물리적 경로 길이를 가지면서 트랜잭션의 전달 경로는 링크의 부하에 따라 동적으로 선택되도록 방법을 제시하였고, 결과적으로 링크의 경쟁 완화에 따라 트랜잭션의 처리 지연을 감소시켰다.

논문에서 제시한 방법에 따른 실험 결과, 동적 경로 설정 기법을 적용한 경우 기존 방법에 비하여 최대 6.3%의 수행 성능 개선을 나타냈고, 링크의 경쟁에 따른 링크 점유 시간의 편차는 최대 9.6% 감소되었다. 결과적으로 본 논문에서 제시한 동적 경로 설정 방법을 이용할 경우, 링크에 대한 균형적 사용이 동적으로 이루어짐을 확인할 수 있었으며, 이러한 특성은 보다 높은 노드간 트랜잭션 부하에 따라 더욱 장점을 갖게 되므로 차후 점차 높아지는 프로세서의 성능에 비례하여 심화되는 상호연결망의 경쟁 및 부하 분배에 좋은 해결책이 될 수 있음이 확인되었다.

### 참 고 문 헌

- [1] J. Torrellas and V. Krishnan, "A Chip-Multiprocessor Architecture with Speculative Multithreading," *IEEE Trans. Comp.*, Sept., 1999.
- [2] S. S. Nemawarkar, R. Govindarajan, G. R. Gao, and V.K. Agarwal "Performance of Interconnection Network in Multithreaded Architectures," *Lecture Notes in Computer Science*, Vol.817, pp.823-826, 1994.
- [3] Daniel Lenoski, James Laudon, Kourosh Gharachorloo, Wolf-Dietrich Weber, Anoop Gupta, John Hennessy, Mark Horowitz, and Monica S. Lam, "The Stanford Dash multiprocessor," *IEEE Trans. Comp.*, Vol.25 No.3, pp.63-79, Mar., 1992.
- [4] IEEE Computer Society, *IEEE Standard for Scalable Coherent Interface(SCI)*, Institute of Electrical and Electronics Engineers, Aug., 1993.
- [5] Tom Lovett and Russel Clapp, "STING: A CC-NUMA Computer System for the Commercial Marketplace," *Proc.*

of the 23th International Symp. on Computer Architecture, pp. 308-317, May, 1996.

[6] P.Mak, G.E.Strait, M.A.Blake, K.W.Kark, V.K.Papazova, A.E. Seigler, G.A.Van Huben, L.Wang, and G.C.Wellwood, "Processor subsystem interconnect architecture for a large symmetric multiprocessor system," IBM Journal, Vol.48. No.3/4, May/July, 2004.

[7] <http://www.dg.com/>

[8] <http://panda.snu.ac.kr/nrl/>

[9] Hyo-Joong Suh, Sung Woo Chung, "DRACO: Optimized CC-NUMA system with Novel Dual-Link Interconnections to Reduce the Memory Latency," Proc. MEDIA-2004 Workshop, July, 2004.

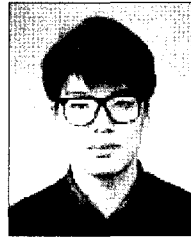
[10] 서효중, "컨버텨 이증링크를 갖는 고크확장성 CC-NUMA 시스템", 한국정보과학회논문지A, Vol.31, No.9, pp.487-494, 2004. 10.

[11] 서효중, "다중 프로그램 환경에 적합한 이증 연결 CC-NUMA 시스템", 한국정보처리학회논문지A, Vol.11, No.3, pp.321-328, 2004. 06.

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

[13] A-T. Nguyen, M. Michael, A. Sharma, and J. Torrellaz, "The Augmint multiprocessor simulation toolkit for Intel x86 architecture," Proc. of the IEEE Conf. Computer Design, Oct., 1996.

[14] S.C.Woo, M.Ohara, E.Torrie, J.P.Singh, and A.Gupta. "Methodological considerations and characterization of the SPLASH-2 parallel application suite," Proc. International Symp. on Computer Architecture, pp.24-36, 1995.



### 서 효 중

e-mail : hjsuh@catholic.ac.kr

1991년 서울대학교 이학사

1994년 서울대학교 컴퓨터공학  
(공학석사)

2000년 서울대학교 컴퓨터공학  
(공학박사)

2002년 지씨티 리서치 선임연구원

2003년~현재 서울대학교 컴퓨터연구소 객원연구원

2003년~현재 가톨릭대학교 컴퓨터정보공학부 조교수

관심분야: 컴퓨터 구조, 병렬처리 시스템, 내장형시스템,  
클러스터 시스템