

PARSEC을 이용한 분산공유메모리 다중프로세서 시스템의 성능분석

박 준 석[†] · 전 창 호^{††}

요 약

본 논문에서는 분산공유메모리 다중프로세서 시스템에서 하드웨어 구성요소와 실행환경이 시스템의 전체 성능에 미치는 영향을 시뮬레이션을 통하여 분석한다. PARSEC[1, 2]을 이용하여 분산공유메모리 다중프로세서 시스템을 실제 실행환경에 근접하게 모델링하고 그 모델링된 시스템상에 2D FFT를 가상 실행하는 방식의 시뮬레이션 결과, 일반적으로 성능분석을 할 때 성능요소로 고려하지 않는 군소 하드웨어 요소들이 시스템 구성에 따라 시스템의 전체 성능에 상당한 영향을 미침을 밝힌다. 또한 반복순환 구문의 오버헤드, 코드최적화 등 실행조건에 따른 성능의 변화도 정량적으로 분석한다.

Performance Analysis of A Distributed Shared Memory Multiprocessor System Using PARSEC

Joon-Seok Park[†] · Chang Ho Jeon^{††}

ABSTRACT

In this paper, the effects of the hardware components and runtime environments on the overall performance of a distributed shared memory system are analyzed through simulation. In simulation, the system is modeled using PARSEC[1, 2] closely to the real runtime environment and the 2D FFT is virtually executed on it. The results of simulation show that the minor hardware components such as bus interfaces and local bus of a processor, which are usually ignored or neglected when analyzing performance, have significant impacts on the overall system performance. Performance variations caused from runtime environments such as loop overhead and code optimization are also analyzed quantitatively.

1. 서 론

현재 대부분의 병렬시스템은 최대 수백 MFLOPS 이상의 성능을 가진 마이크로프로세서 또는 DSP칩을 연산소자로 사용하고 있다[3-7]. 그러나 프로세서 자체의 속도가 빨라짐에도 불구하고 시스템의 전체 성능은 프로세서의 속도에 비례하여 증가하지 못하고 있다. 이것은 프로세서 이외의 다른 하드웨어 구성요소 및 실행환경에서 성능저하 요인이 있음을 의미한다. 이러

한 성능저하 요인의 비중은 프로세서의 속도가 빨라지면 빨라질수록, 그리고 많은 수의 프로세서를 사용하면 할수록 점점 커지게 될 것이다. 시스템의 전체 성능에 미치는 이들의 영향이 커짐에 따라 다중프로세서 시스템의 성능분석에서 이들이 성능인자로 사용되어야 함은 보다 정확하고 현실적인 결과를 얻기 위한 필수 조건이라고 할 수 있다. 그러므로 시스템의 성능에 미치는 하드웨어 구성요소와 실행환경의 영향을 정량적으로 분석하여 성능분석에서 반드시 포함되어야 할 성능인자의 선택기준을 마련할 필요가 있다.

이에 본 논문에서는 TMS320C67x와 PCI 버스 등

[†] 정 회 원 LG전자/정보통신중앙연구소 선임연구원
^{††} 중신회원 한양대학교 전자컴퓨터공학부 교수
논문접수 2000년 7월 4일, 심사완료 2000년 10월 16일

상용 하드웨어 소자를 사용하여 병렬 디지털 신호처리 용으로 개발 중에 있는 분산공유메모리 다중프로세서 구조[8, 9]를 대상으로 2D FFT를 이용하여 시스템의 전체 성능에 미치는 하드웨어 구성요소와 실행환경의 영향을 시뮬레이션을 통하여 분석한다. 시뮬레이션에서는 PARSEC[1, 2]을 이용하여 시스템의 실제 실행환경을 모델링하고 그 모델링된 시스템상에 2D FFT를 구현하여 실행시킴으로써 프로세서의 연산시간과 지역 메모리 및 공유메모리의 사용시간 등 시스템의 전체 성능에 지배적인 영향을 미치는 성능요소 뿐만 아니라 버스 인터페이스와 프로세서의 지역버스와 같은 군소 하드웨어 구성요소들의 영향과 반복순환 구문의 오버헤드, 버스 중재시간, 그리고 코드 최적화 등과 같은 실제 실행환경에서의 성능요인들이 어떤 영향을 미치는지 정밀 분석한다.

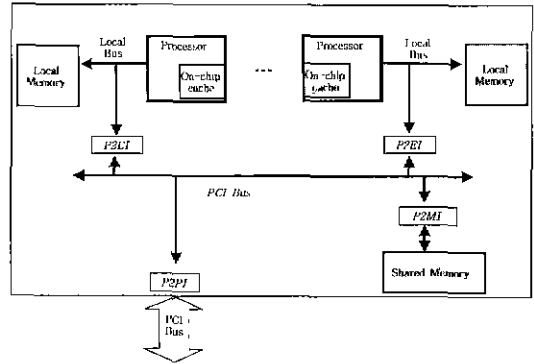
본 논문의 2장에서는 성능분석의 대상인 분산공유메모리 다중프로세서시스템과 성능분석용 어플리케이션인 2D FFT에 대해서 설명하고 3장에서는 시뮬레이션 언어인 PARSEC에 대해 간략히 기술하고 시뮬레이션 과정 및 결과를 논한다. 그리고 4장에서 결론을 맺는다.

2. 성능분석의 대상

2.1 분산공유메모리 다중프로세서 구조

(그림 1)은 본 논문에서 성능분석의 대상으로 하는 분산공유메모리 다중프로세서 구조의 구성도로서 보드에는 TMS320C67x가 최대 네 개까지 장착될 수 있으며 각각의 프로세서는 자신의 지역메모리를 가지고 있다. 그리고 이 프로세서는 내부에 캐시 메모리를 가지고 있으며 지역메모리와의 연결에는 EMIF라는 TMS320C67x의 지역버스를 사용한다 또 보드에는 보드내의 프로세서들 뿐만 아니라 시스템의 모든 프로세서들에서 접근이 가능한 공유메모리가 있으며, 공유메모리와 보드내의 프로세서 또는 지역메모리와의 연결은 PCI 버스를 통해서 이루어진다 그리고 보드 내의 PCI 버스와 프로세서의 지역버스의 연결은 PCI-to-EMIF 인터페이스(P2EI)를 통해서, PCI 버스와 공유메모리 사이에는 PCI-to-메모리 인터페이스(P2MI)가 있다. 이 구조는 보드 단위로 성능확장이 가능한데 다중보드일 경우, 보드와 보드간의 연결은 또 다른 PCI 버스를 통해서 이루어지며 보드내의 PCI 버스와 보드간 연결을 위한 PCI 버스 사이에는 PCI-to-PCI 인터페이스(P2PI)가 있다.

스(P2PI)가 있다.



(그림 1) 분산공유메모리 다중프로세서 모델

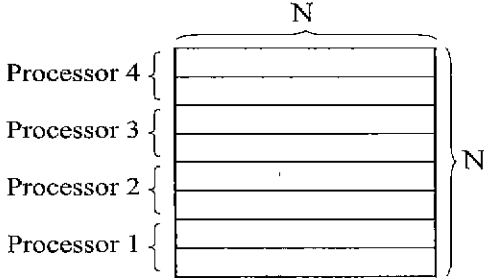
2.2 2D FFT

본 논문에서는 성능분석을 위한 어플리케이션으로 2D FFT를 사용한다. 많은 산술 연산을 필요로 하며 신호처리 응용분야의 대표적인 알고리즘인 2D FFT는 2차원 입력데이터에 대해서 1D FFT 연산을 반복적으로 수행한다[10]. 즉, 2D FFT는 1D FFT의 확장으로써 전체적인 실행과정은 세 단계로 이루어진다. 첫 번째는 2차원 매트릭스 데이터에 대해서 행 방향으로 1D FFT를 수행하는 단계이다. 두 번째는 매트릭스 변환을 하는 단계로써 이것은 아직 연산이 이루어지지 않은 열 방향의 데이터를 행 방향 1D FFT가 가능하도록 행 방향으로 재배치하는 작업이다 세 번째는 행 방향으로 변환된 열 방향의 데이터에 대해서 다시 행 방향 1D FFT를 수행하는 단계이다.

2D FFT 데이터를 성능분석의 대상인 분산공유메모리 다중프로세서 구조에서 병렬처리할 수 있도록 데이터를 분할하고 할당하는 것은 (그림 2)와 같은 방식으로 한다 예를 들어, N*N 2D FFT 데이터를 네 개의 프로세서를 가진 시스템에서 실행할 경우, (그림 2)에서처럼 데이터를 행 단위로 분할함으로써 1D FFT 연산 도중에 프로세서간 통신이나 보드간 통신이 발생하지 않도록 한다. 그리고 이렇게 행 단위로 분할된 데이터는 시스템의 모든 프로세서에게 균등하게 분배함으로써 프로세서간 부하균형이 이루어지도록 한다.

(그림 2)와 같은 방식으로 데이터를 분할하여 (그림 1)과 같은 시스템의 각 프로세서의 지역메모리로 데이터를 할당한 경우, 각각의 프로세서가 계산을 위해서는 자신의 지역메모리를 사용하고 매트릭스 변환을 위

해서는 공유메모리를 사용한다



(그림 4) 2D FFT 데이터의 분할 및 할당 방식

3. PARSEC을 이용한 시뮬레이션

3.1 시스템 모델링

PARSEC(PARallel Simulation Environment for Complex systems)은 ULCA에서 개발된 C언어 기반의 이산 사건 방식의 시뮬레이션 환경이다[1, 2]. 병렬시스템을 시뮬레이션 할 수 있는 시뮬레이터나 도구에는 EPPI[11], WARPED[12], Chaos[13] 등 여러 가지가 있으나 본 논문에서 성능분석의 대상으로 하는 시스템의 구성요소들과 그 상호간의 동작을 세밀하게 모델링하기에는 부족하거나 부적합한 부분들이 있어 PARSEC을 이용하여 시뮬레이터를 구현한다.

PARSEC에서는 실제 시스템의 하드웨어 구성요소를 entity라는 논리 프로세서로, 하드웨어 구성요소간의 상호작용인 이벤트는 message로 정의한다. 이 message는 실제 시스템에서 이벤트가 언제 발생하는지를 나타내는 시간정보인 타임스탬프(time stamp)를 가지고 있다. 전체적으로 PARSEC을 이용한 시뮬레이션의 흐름은 시간적인 순서로 실행되어야 하는 연속적인 이벤트들로 구성되며 PARSEC 프로그램은 논리 프로세서인 entity와 C 함수로 이루어진다. 분산공유메모리 다중프로세서 시스템을 모델링하기 위해서는 먼저 (그림 1)의 각 하드웨어 소자들을 entity로 구현한다. 하드웨어 구성요소들이 entity로 구현되었으면 다음으로는 이들 entity사이의 상호작용을 정의한 message가 필요하다. 본 논문에서는 구현한 message의 종류를 성격에 따라서 크게 세 가지로 구분한다. 첫째, 통신경로를 형성하는데 필요한 메시지, 둘째, 데이터 전송에 관련된 메시지, 셋째, 자원의 요청과 할당에 관련된 메시지가 그것

이다. 시스템을 모델링하려면 entity들 사이에 가상의 경로를 형성하여 어떤 entity와 어떤 entity가 서로 연결되어 있는지를 먼저 정의해 줄 필요가 있다. 이때 사용하는 message가 바로 통신경로의 형성에 관련된 message이다. Entity들 사이에 통신경로가 형성되면 entity들간에 데이터를 주고 받을 수가 있는데 이때 사용하는 message가 데이터의 전송에 관련된 message이다. 그리고 자원의 요청과 승인에 관련된 message는 PCI 버스와 같은 공유자원을 요청하고 사용권한을 획득하는 과정에서 필요한 message이다. 시뮬레이션에서는 이렇게 구현한 entity와 message를 사용하여 분산공유메모리 구조를 모델링한다. <표 1>은 시뮬레이션에 사용한 주요 파라미터들의 수치와 범위를 요약한 것이다.

<표 1> 시뮬레이션에서 사용된 파라미터 값의 범위

파라미터	범위(단위)
보드의 수	1~10(개)
보드당 프로세서의 수	1~4(개)
2D FFT의 데이터 크기	64~1024(points)
데이터 포인터의 크기	8(bytes)
프로세서의 사이클 시간	6(ns)
지역메모리의 크기	128(MB)
공유메모리의 크기	제한없음
캐쉬의 크기	64(KB)
메모리 소자의 쓰기 접근시간	50~70(ns)
메모리 소자의 읽기 접근시간	65~90(ns)
캐쉬 접근시간(읽기/쓰기)	24(ns)
EMIF 버스의 대역폭	400(MB/s)
PCI 버스의 대역폭	132(MB/s)
PCI-to-EMIF 인터페이스 지연시간	40(ns)
PCI-to-메모리 인터페이스 지연시간	50(ns)
PCI-to-PCI 인터페이스 지연시간	40(ns)
버스 인터페이스 버퍼 크기(쓰기)	32~128(bytes)
버스 인터페이스 버퍼 크기(읽기)	16~64(bytes)
버스중재 시간	20~50(ns)

3.2 시뮬레이션 결과 및 고찰

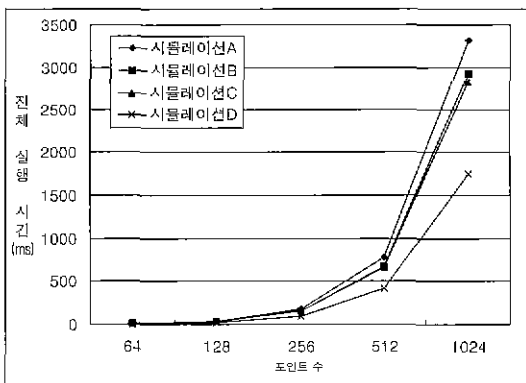
(그림 3)은 분산공유메모리 구조에서 1024*1024 포인트 2D FFT를 실행할 경우 소요되는 전체 실행시간을 보인 그래프이다. 본 논문의 시뮬레이션에서는 버스 인터페이스의 내부 버퍼의 영향과 TMS320C67x의 내부 캐쉬 및 코드 최적화에 따른 성능변화를 알아보기 위하여 <표 2>와 같이 실행조건을 네 단계로 변화시키면서 전체 실행시간을 측정하였다.

〈표 2〉 실행조건에 따른 시뮬레이션의 종류

	버스 인터페이스 버퍼의 유부	코드최적화 여부	캐쉬의 유부
시뮬레이션A	×	×	×
시뮬레이션B	○	×	×
시뮬레이션C	○	○	×
시뮬레이션D	○	○	○

TMS320C67x는 내부에 캐쉬 메모리를 가지고 있으며 여섯 개의 ALU와 두 개의 승산기를 가지고 있어서 프로그램의 코드를 최적화할 경우 한 사이클에 최대 8개의 연산을 수행할 수 있다. 또 버스 인터페이스는 내부에 최대 64바이트 크기의 버퍼를 가지고 있다.

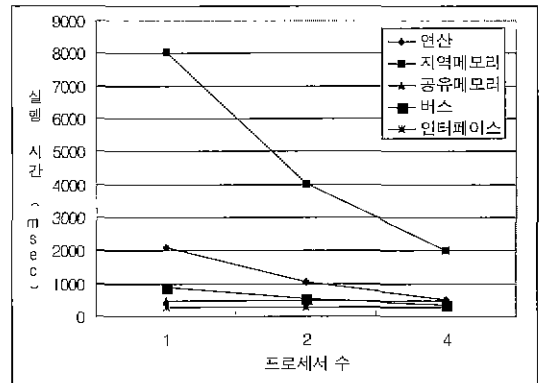
(그림 3)에서 처럼 버스 인터페이스의 내부 비퍼, 코드 최적화, 그리고 캐쉬를 사용할수록 전체 실행시간이 짧아짐을 알 수 있다 프로세서의 수가 네 개이고 데이터의 크기가 1024*1024일 경우, 전체 실행시간에서 버스 인터페이스의 내부 버퍼의 영향은 약 11.9%, 코드 최적화의 영향은 2.9%, 그리고 캐쉬의 영향은 약 32.7%로 나타난다 특히 TMS320C67x의 내부 캐쉬를 사용할 경우 전체 실행시간이 현저히 감소함을 알 수 있다 이것은 메모리 계층구조를 갖는 분산공유메모리 다중프로세서 시스템에서 데이터의 효과적 캐싱이 전체 시스템의 성능에 큰 영향을 미침을 보여준다.



(그림 3) 전체 실행시간

(그림 4)는 1024*1024 포인트 2D FFT를 실행할 경우, 전체 실행시간에서 인산 및 하드웨어의 구성요소들에서 소요되는 지연시간을 나타낸 것이다. (그림 4)에서 처럼 시스템에서 2D FFT를 실행할 경우, 지역메모리가 전체 시스템의 성능에 가장 큰 영향을 미치는 요인임을 알 수 있다. 이것은 프로세서가 지역메모리

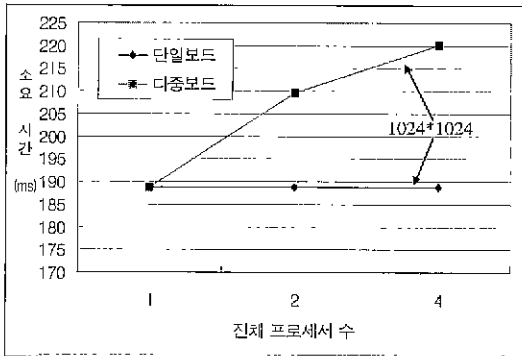
로 할당된 데이터를 대상으로 FFT 연산을 수행하기 때문에 지역메모리를 빈번히 접근하게 되고 또, 지역메모리의 접근시간이 다른 구성요소들에 비해 상대적으로 길기 때문이다 공유메모리의 사용으로 인한 지연시간과 버스 인터페이스의 사용시간은 프로세서의 수에 상관없이 일정하게 나타나는데 이것은 매트릭스 변환을 위해 공유메모리를 사용하는 빈도는 프로세서 수가 아닌 데이터의 크기에 영향을 받기 때문이다. 그리고 프로세서에서의 연산시간과 버스의 지연시간은 프로세서의 수가 증가하면 증가한 만큼 병렬 수행되기 때문에 적은 비율로 감소한다.



(그림 4) 전체 실행시간에 대한 하드웨어 구성요소의 지연시간

(그림 5)는 1024*1024 2D FFT를 실행할 경우, 전체 실행시간 중에서 버스 인터페이스에서 소요되는 시간을 보인 그래프이다. 버스 인터페이스에는 PCI-to-EMIF, PCI-to-메모리, 그리고 PCI-to-PCI 인터페이스가 포함된다. (그림 5)에서 알 수 있듯이 버스 인터페이스에서의 소요시간은 단일 보드일 경우, 프로세서의 수에 상관없이 일정한 소요시간을 보임을 알 수 있다. 이것은 버스 인터페이스가 매트릭스 변환을 위해 공유메모리를 접근할 때 사용되는 구성요소이며 공유메모리의 사용횟수는 프로세서의 수와는 무관하기 때문이다 그러나 시스템의 전체 프로세서 수는 동일하지만 다중 보드일 경우는 보드간 통신이 발생하고 보드의 수가 증가할수록 PCI-to-PCI 인터페이스의 사용횟수가 증가하므로 보드의 수가 늘어날수록 버스 인터페이스에서의 소요시간도 길어짐을 알 수 있다 참고로, (그림 5)는 시스템의 전체 프로세서 수를 단일 보드와 동일한

조건에서 비교하기 위하여 다중 보드일 경우, 각각의 보드에서 사용하는 프로세서의 수를 하나로 고정하여 시뮬레이션한 결과이다.

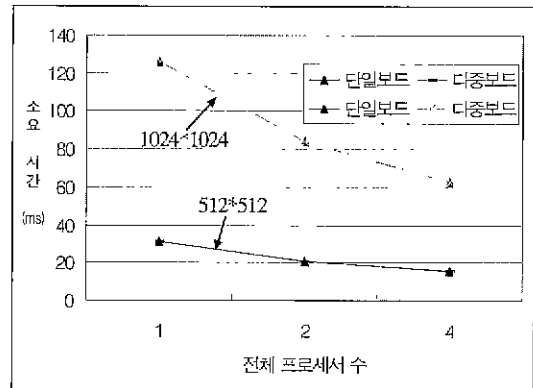


(그림 5) 버스 인터페이스에서의 소요시간

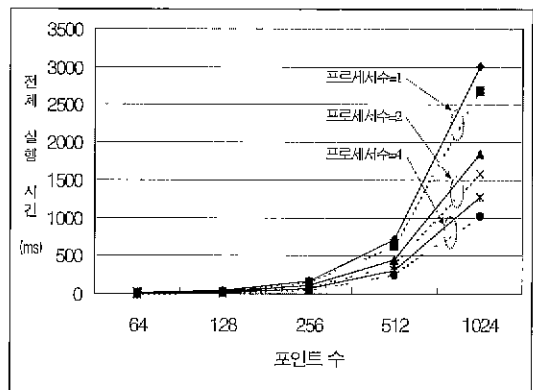
(그림 6)은 전체 실행시간 중에서 프로세서의 지역 버스인 EMIF에서 소요되는 시간을 보인 그래프이다. EMIF 버스의 소요시간은 단일보드와 다중보드 상관없이 시스템의 전체 프로세서의 수에 의해서 영향을 받는다. 왜냐하면, 모든 EMIF 버스는 병렬로 사용되는 구성요소이고 전체 프로세서의 수가 증가하면 증가한 만큼 사용횟수가 감소하기 때문이다.

(그림 7)은 버스 인터페이스와 EMIF 버스의 영향을 포함한 전체 실행시간과 포함하지 않은 전체 실행시간을 비교하여 보인 그래프이다. 실선으로 표시한 것이 이들의 영향을 포함한 전체 실행시간이며 점선으로 표시한 것이 이들의 영향을 포함하지 않은 전체 실행시간이다. 프로세서의 수가 하나일 경우, 전체 실행시간에 대한 이들의 영향은 약 10% 정도이지만 프로세서의 수가 네 개일 경우, 이들의 영향은 20% 까지 증가함을 알 수 있다.

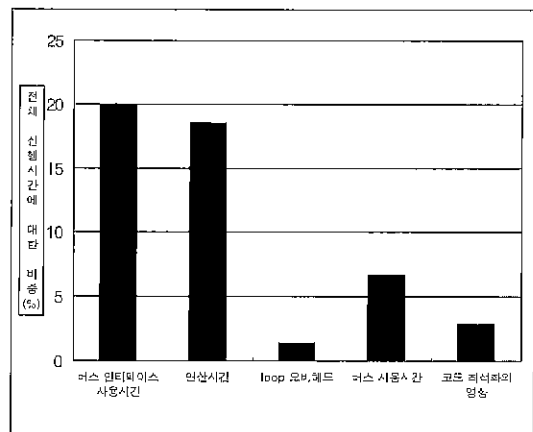
(그림 8)은 전체 실행시간 중에서 버스 인터페이스의 지연시간과 실제 실행환경의 성능요소들의 비중을 비교하여 보인 그래프이다. 프로세서의 수가 네 개일 경우, 전체 실행시간에서 프로세서의 연산시간이 약 18.5%, 반복순환 구문에서의 지연시간이 1.4%, 버스 중재시간을 포함한 PCI 버스의 사용시간이 6.7%, 그리고 코드 최적화의 효과가 대략 2.9% 정도인 것과 비교하면 버스 인터페이스와 지역버스의 영향이 결코 무시할 수 없음을 알 수 있다.



(그림 6) EMIF 버스에서의 소요시간



(그림 7) 전체 실행시간에 미치는 버스 인터페이스와 EMIF 버스의 영향



(그림 8) 실행환경에 따른 성능요소들의 비중

4. 결 론

본 논문에서는 분산공유메모리 다중프로세서시스템 구조를 대상으로 2D FFT를 이용하여 시스템의 전체 성능에 미치는 하드웨어 구성요소들과 실행환경의 영향을 시뮬레이션을 통하여 정량적으로 분석하였다.

PARSEC을 이용한 본 논문의 시뮬레이션은 TI의 벤치마크[14]와 동일한 조건에서 92.4%의 신뢰도를 보였으며 시뮬레이션 결과, 시스템의 프로세서가 많아지면 많아질수록 일반적으로 성능분석을 할 때 성능요소로 고려하지 않는 버스 인터페이스와 프로세서의 지역버스가 시스템의 성능에 미치는 영향이 점점 커지게 될 수 있다. 따라서 이러한 하드웨어 구성요소들이 성능분석을 할 때 무시할 수 있는 성능요소가 아니라 오히려 시스템의 정확한 성능분석을 위해서는 반드시 포함해야할 성능인자라는 것을 시뮬레이션을 통하여 확인하였다. 더우기 최근과 같이 마이크로프로세서의 속도가 점점 빨라질수록 시스템의 전체 성능에서 이들이 차지하는 비중은 상대적으로 점점 커지게 될 것이므로 성능분석에서 신중하게 고려해야할 성능인자가 된다.

참 고 문 헌

[1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H. Y. Song, "Parsec: A Parallel Simulation Environment for Complex Systems," IEEE Computer, pp.77-85, Oct. 1998.

[2] PARSEC User Manual, UCLA Parallel Computing Lab., Jan. 1999

[3] A. J. Anderson, "Selection Criteria in the Development of a Multiple Processor Based DSP System," Journal of Microcomputer Application, pp.327-345, 1992.

[4] P. V. R. Raja and S. Gancsan, "An SIMD Multiple DSP Microprocessor System for Image Processing," Microprocessors and Microsystems, pp.493-501, Nov. 1991.

[5] T. Sawabe, T. Fujii, H. Nakada, N. Ohta, and S. Ono, "A 15GFLOPS Parallel DSP System for Super High Definition Image Processing," IEICE Trans. Fundamentals, Vol.39, No.7, pp. 786-792, Jul. 1992.

[6] F. Kurugollu, H. Palaz, H. Gumuskaya, E. Harmanci and B. Orencik, "Advanced Educational Parallel

DSP System Based on TMS320C25 Processors." Microprocessors and Microsystems, Vol.19, No 3, pp 147-156, Apr. 1995.

[7] Navy Signal Processors Program Office, PMS 429.

[8] 박준석, 전창호, 박성주, 이동호, 오원천, 한기택, "병렬 디지털 신호처리시스템의 성능분석", 한국음향학회논문지. 제18권 제1호, pp 3-9, Jan 1999

[9] 문병표, 박준석, 전창호, 박성주, 이동호, 한기택, "TMS320C67x 기반 병렬신호처리시스템의 설계와 성능분석". 한국정보처리학회논문지, 제 7권 제1호, pp.65-73, Jan. 2000

[10] Parallel Processing With the TMS320C4x, Texas Instruments, 1994.

[11] T. Stiermerling, "A Parallel Simulator of a Multiprocessor Design Implemented on a Transputer Network," Int'l Journal of Computer Simulation, Vol.2, No.2/3, 1992

[12] <http://www.ece.uc.edu/~paw/warped/doc/index.html>

[13] <http://www.ntua.gr/parallel/simulation/communications/chaos/simulator/>

[14] <http://www.ti.com/sc/docs/products/dsp/c6000/62b ench.htm#ift>

박 준 석



e-mail: ds2fcy@lgic.co.kr

1993년 한양대학교 전자계산학과 졸업(학사)

1995년 한양대학교 대학원 전자계산학과(공학석사)

2000년 한양대학교 대학원 전자계산학과(공학박사)

2000년~현재 LG전자/정보통신중앙연구소 선임연구원
관심분야 : 성능분석, 병렬처리, 차세대 네트워크

전 창 호



e-mail : chjeon@paral.hanyang.ac.kr

1977년 한양대학교 전자공학과 졸업(학사)

1986년 Cornell 대학교(박사)

1986년 성균관대학교 전기공학과 조교수

1989년~현재 한양대학교 전자컴퓨터공학부 교수

관심분야 : 컴퓨터 구조, 병렬처리시스템, 성능분석