

□ 기술예설 □

병렬처리 컴퓨터의 추세

삼성전자 박승권* · 유승희**

● 목 차 ●

- | | |
|--|--------------------------------------|
| 1. 서 론 | 3.2 고성능 상용 서버
(Enterprise Server) |
| 2. MPP(MASSIVELY PARALLEL
PROCESSING)의 개요 | 3.3 멀티미디어 서버
(Multimedia Server) |
| 2.1 MPP의 정의 | 4. 국내 MPP 현황 |
| 2.2 MPP의 등장 배경 | 4.1 정보 통신부 주전산기IV |
| 2.3 MPP의 구조 분류 | 4.2 통상 산업부 대형컴퓨터 |
| 3. MPP의 응용 | 5. 결 론 |
| 3.1 고성능 연산 서버(Computation Server) | |

1. 서 론

20세기 후반들어 컴퓨터와 통신 기술의 발달은 인간 사회에 급속도로 많은 변화를 가져오고 있다. 고속의 수치 연산을 목적으로 발명되었던 컴퓨터는 문자(Text), 음성(Voice), 영상(Image)등의 다양한 형태의 정보들에 대하여 저장 및 처리가 용이한 구조로 발전해 가고 있으며 따라서 사회의 거의 모든 분야에서 컴퓨터를 이용한 이러한 정보들의 활용에 크게 의존하고 있다. 한편 고속의 통신 기술의 발달은 이용하고자 하는 정보의 위치에 대한 물리적 거리의 제약을 거의 없게 해주고 있다. 이러한 컴퓨터와 통신 기술을 이용한 정보 처리 형태의 발전을 많은 사람들이 18세기 내연기관 발명에 의한 산업 혁명에 견줄만한 정보 혁명이라고 일컫고들 있으며 다가오는 21세기를 정보화 사회라는 용어로 단정지어 사회 생활에서의 컴퓨터를 이용한 정보 의존도가 더욱 더 깊어

질 것으로 예상하고 있다.

21세기 정보화 시대에서의 다양한 컴퓨터의 응용 분야는 기존의 컴퓨터 구조로서는 효율적인 해결책을 제공하기가 어려우며 특히 초고속 정보통신망(Information Super Highway) 등의 고속화 되고 있는 통신 수단을 뒷받침하고 다양한 형태의 정보를 신속히 처리할 수 있는 새로운 구조의 컴퓨터가 요구되고 있다. 대규모 병렬처리 컴퓨터(Massively Parallel Processing Computer, MPP)들은 통상 수 백 또는 수 천개의 프로세서들이 고속의 상호연결 방식(Interconnection Mechanism)으로 서로 연동되어 동작하는 구조를 가지고 있으며 이러한 새로운 요구들을 충족시킬수 있는 구조로 최근에 많이 거론되고 있다.

MPP들은 이전의 많은 새로운 구조의 컴퓨터들이 그래왔듯이 학교나 연구소 또는 소수의 모험 기업들에 의해 시도되어 그 타당성이 검증되어 왔다. 오늘날에 이르러서는 많은 대형 컴퓨터업체들에 의해 기존의 벡터형 슈퍼 컴퓨터나 메인프레임을 대체할 수 있는 차세대 대

*비 회 원

**중심회원

형 컴퓨터 구조로서 상용 MPP 제품들이 시장에 출시되고 있는 실정이다. 현재 우리나라에서도 정보통신부 주관의 주전산기 IV 개발 사업, 통상산업부 주관의 국산 대형컴퓨터 개발 사업 등 두가지의 대형 컴퓨터 개발 프로젝트 모두가 MPP 구조를 기본 구조로 설정하여 진행되고 있다.

한편 많은 MPP 컴퓨터들이 동시에 실행될 수 있는 작업의 수를 다수의 프로세서를 이용하여 늘림으로써 성능을 극대화 한다는 전략(strategy)상의 공통 목표를 가지고 있으나 그 구현(implementation) 방법에는 여러가지 다양한 구조가 제의되고 있는 실정이다. 본 고에서는 이러한 MPP중 최근에 발표된 대표적인 몇몇 시스템들에서 공통되는 특징으로 나타나는 구현 특징들을 살펴보고 이들로 부터 MPP의 최근 추세에 대하여 알아보며 또한 현재 국내에서 진행중인 MPP들의 구조에 대하여 알아본다.

2. MPP(Massively Parallel Processing)의 개요

2.1 MPP의 정의

MPP 시스템이란 수 백 또는 수 천개까지의 프로세서와 메모리, 그리고 입출력 장치들이 고속의 상호 연결망으로 연결되어 있으면서 사용자에게는 단일 시스템 이미지를 제공할 수 있는 컴퓨터를 말한다.

MPP에서 제공하는 대규모의 다중 구조는 컴퓨터에 부과되는 각종 작업들의 병렬성을 최대한 활용하여 고성능을 제공할 수 있으며 또한 중복된 자원을 이용한 고장 감내 기능의 구현을 용이하게 해줄 수 있다. 이는 종래의 다중 처리(multiprocessing)나 분산 처리(distributed computing)등과 동일한 등장 배경을 가지면서 제공할 수 있는 자원의 규모(scale)면에서 다중 처리에 비해서 대규모화되고 자원들간의 친화도(coupling) 면에서는 분산 처리보다 더욱 더 강화된 형태의 작업 환경을 제공하는 컴퓨터 구조라고 말할수 있다.

2.2 MPP의 등장 배경

일반적으로 컴퓨터의 성능은 하나의 작업에 대한 처리가 얼마나 빠른가에 대한 응답 시간(turnaround time, response time)과 단위 시간내에 얼마나 많은 작업들이 완료될 수 있는가의 작업 처리율(throughput)에의 두 가지 측면에서 분류되며 이들은 컴퓨터를 이용하는 응용 프로그램의 특성에 따라 컴퓨터의 성능을 판단하는 지표로 사용되고 있다[1].

현재 컴퓨터를 이용하는 두가지 대표적인 응용(Application) 분야는 복잡한 수치 연산을 주로하는 과학 및 공학 관련 응용 프로그램들과 문자와 같은 데이터 처리에 주로 치중하는 상업적(Business) 응용 프로그램들로 들 수 있다[2].

전자의 경우 복잡한 수학적 공식을 많이 사용하며 정수 및 부동소수점과 같은 데이터 형에 대한 연산 동작이 대규모로 수행되어진다. 이는 전체 작업이 종료 되어야만 유효한 결과를 얻을 수 있기 때문에 하나의 커다란 작업으로 볼 수 있으며 프로그램의 실행시 빠른 응답시간이 주요 관심사 이었다. 이를 위해서 작업에 내재하는 각종 병렬성을 추출하여 동시에 실행될 수 있는 작은 작업 단위로 나눈 다음 이들을 동시에 실행할 수 있는 구조에 대한 연구가 많이 수행되어 왔으며 그 결과 중첩된 파이프라인 방식으로 대량의 벡터 연산을 할 수 있는 벡터 컴퓨터와 데이터 병렬성을 이용하여 하나의 제어흐름 상에서 벡터 전체에 대한 동일한 연산이 동시에 실행되는 SIMD 형태의 컴퓨터들이 나타났으며 이들은 슈퍼 컴퓨터로 분류되고 있다.

후자의 경우는 데이터베이스와 같은 대용량의 데이터에 대한 관리 체계를 컴퓨터가 제공하여 많은 사용자들이 이들 데이터를 공유할 수 있도록 하며 이들 각자가 공유하는 데이터에 대한 조회나 변경 등과 같은 데이터 처리 작업을 동시에 실행할 수 있도록 하는 것이 주요 작업 내용이다. 이러한 작업 환경에서는 일정 시간 단위로 얼마나 많은 트랜잭션이 처리될 수 있는지가 컴퓨터의 성능을 나타내는 지수가 된다. 작업의 속성상 메모리와 데이터 저장 장치 사이에 매우 빈번한 데이터 입출력이 발생하게 되며 전체 시스템의 성능은 데이터 입출

력 장치에서 얼마나 빠른, 그리고 얼마나 많은 데이터 흐름을 동시에 지원할 수 있는가에 달려있다. 대규모의 상업용 응용 프로그램은 주로 메인프레임에서 운용되고 있다.

대부분의 벡터 슈퍼 컴퓨터나 메인프레임의 경우 복잡하고 값비싼 하드웨어 구성을 가지며 제조업체 고유의 규격을 갖는다. 따라서 새로운 기술의 적용이나 신규 모델의 개발시 많은 시간과 비용이 들게 된다. 또 한편으로 이들의 구조는 확장성의 한계에 쉽게 도달하게 되기 때문에 작업의 규모가 커져서 컴퓨터의 성능이 사용자의 요구를 충족시킬수 없는 경우 사용자는 전체 시스템을 더욱 더 강력한 성능을 제공하는 새로운 컴퓨터로 교체해야 된다는 문제점을 가지고 있다.

이러한 기존 대형컴퓨터들의 문제점은 최근에 컴퓨터 관련 신기술들이 빠른 속도로 발전하고 있으며 따라서 컴퓨터의 구조가 이들을 신속히 채용할 수 있어야 한다는 점과 새롭게 발생하는 컴퓨터의 응용 분야들이 과거에 비해 훨씬 더 복잡하고 커다란 문제들에 대한 해결을 요구한다는 점에서 그 심각성을 더하고 있다. 따라서 이러한 문제들을 효율적인 비용으로 해결할 수 있는 MPP 구조가 관심의 대상이 되고 있다.

2.3 MPP의 구조 분류

Flynn의 분류 방법에 따르면 병렬 처리 컴퓨터의 구조 모델로는 SIMD와 MIMD의 두 가지 형태가 있다[2]. 이중 SIMD의 경우 전체의 제어 흐름을 관장하는 하나의 제어 유닛과 제어 흐름에 동기되어 연산을 실행하는 다수 개의 연산 유닛의 배열로 구성되어 있다. 이는 일련의 데이터 집합에 대하여 동일한 연산을 반복하는 경우에 유리한 구조로 보통 과학 계산용의 응용 분야에서 많이 사용한다. 대표적인 SIMD 형의 MPP로는 TMC의 CM2와 MasPar의 MP1, MP2 등이 있다[2].

MIMD의 경우 각자의 내부에 제어 유닛과 연산 유닛이 내장된 다수 개의 프로세서들로 구성되며 이러한 프로세서들이 각자의 제어 흐름하에서 연산 작업을 실행한다. 병렬 처리를 프로그램의 작성시 MIMD 모델하에서

훨씬 더 자연스러운 프로그램의 서술이 가능하며 따라서 MIMD는 범용성이 뛰어난 구조로 알려져 있다. 보통 SIMD의 경우 범용성에 제약을 받기 때문에 과학 계산용의 응용에 치중하는 면이 있으며 따라서 많은 MPP들이 MIMD 모델의 구조를 채택하고 있으며 향후에도 병렬 처리 컴퓨터에서의 주요 구조로서의 위치를 고수할 것으로 보인다. 이에 대한 상징적인 예로 TMC에서 초기 모델인 CM2의 SIMD 구조를 후속 모델인 CM5에서는 SIMD로서의 동작이 가능한 MIMD 구조로 변경 하였는데[3] 이는 기존의 CM2에서 작성된 응용 프로그램들의 호환성을 유지하면서 향후 MIMD로의 변신을 꾀하기 위한 전략으로 보인다.

MIMD 형태의 MPP는 각 프로세서들 간에 물리적으로 공유된 형태의 메모리가 존재하는지의 여부에 따라 크게 두 가지 구조로 분류될 수 있다.

2.3.1 분산 공유메모리 구조(Distributed Shared Memory Architecture)

분산 공유메모리 구조는 기존의 밀결합 공유메모리 다중프로세서(tightly-coupled shared memory multiprocessor)와 마찬가지로 모든 프로세서에게 메모리에 대하여 동일한 물리적 주소 공간을 제공하여 프로세서들에 의해 메모리가 공유될 수 있는 구조이면서 대규모 확장성을 제공할 수 있는 MPP 구조이다. 따라서 이를 대규모 공유메모리 구조(large scale shared memory architecture)라고도 한다. 병렬처리용 프로그램의 작성이 용이하고 응용 프로그램들의 이식성이 뛰어난 공유메모리 방식의 프로그래밍 모델을 지원하며 프로세서들은 공유메모리 내의 공유 변수를 이용하여 서로간에 통신이나 동기를 수행할 수 있다. 밀결합 다중프로세서에서 문제점으로 지적되어 왔던 확장성 문제를 해결하기 위하여 전체 대역폭의 확장성이 좋은 점 대 점 통신(Point to Point Communication)을 지원하는 상호연결망을 사용하고 각 프로세서들의 메모리에 대한 요구의 집중을 방지하기 위한 분산된 메모리 구조(distributed memory architecture)를 갖는다[4]. 보통 메모리는 일정한 크기로 분할되어 각 프로

세서의 지역메모리(local memory) 형태로 위치하며 프로세서가 자신의 지역메모리가 아닌 다른 프로세서의 지역메모리(보통 이를 원격 메모리, remote memory로 지칭함)를 참조할 때는 상호 연결망을 통하여 직접 참조할 수 있게 된다. 그러나 원격메모리의 참조시 상호연결망을 거쳐야 하므로 지역메모리에 비해 참조 지연 시간이 길어지게 된다. 즉 동일한 물리적 주소 공간 내에서 메모리의 참조 시간이 균등하지 않게 된다. 이러한 구조를 NUMA(Non Uniform Memory Access) 구조라 하며 밀결합 공유 메모리 구조의 경우 공유메모리가 중앙집중식 형태로 되어있어 모든 메모리 영역에 대한 메모리 참조 시간이 동일하므로 이를 UMA(Uniform Memory Access) 구조라 한다[2].

NUMA 구조에서는 지역메모리에 비해 원격 메모리 참조에 걸리는 시간이 너무 길기 때문에 원하는 작업의 성능을 최적화 하기 위해서는 각 지역메모리에 대한 프로그램의 코드 및 데이터의 분할 할당등에 많은 주의가 필요하다. 이러한 문제를 해결하기 위하여 하드웨어나 소프트웨어에 의한 선반입(prefetch), 모든 데이터에 대한 일관성 유지를 지원하는 캐쉬(global coherent cache), 캐쉬의 일관성 유지에 따른 성능 부담을 줄이기 위한 약화된 메모리 일관성 유지 모델(weaker memory consistency model), 원격메모리 참조시 프로세서로 하여금 다른 작업을 하게하여 프로세서의 가동율(processor utilization)을 올리기 위한 다중 쓰레드 기법(multithreading), 운영체제의 메모리 관리자에 의한 지역 메모리간의 효율적인 데이터의 이동(migration) 또는 복사(copy)등이 연구되고 있다[4, 5]. 또한 시스템의 규모가 커질수록 복잡해지는 공유 영역에 대한 동기화(synchronization)를 효율적으로 실행할 수 있는 방법도 연구되고 있다.

원격메모리 참조 지연문제를 해결하기 위하여 위에서 거론된 방법들이 복합적으로 사용되고 있으며 이중 일관성 유지 캐쉬에 대한 사용이 가장 대표적인 해결책으로 주장되고 있으나 하드웨어의 복잡성이 문제점으로 지적되고 있다. 현재 SCI(Scalable Coherent Interface)와 같이 일관성 유지 캐쉬에 대한 IEEE 표준 규

격의 제정과 이를 지원하는 상용 칩세트[6]의 등장 등으로 하드웨어의 복잡성이 어느 정도 해소될 것으로 보이나 컴퓨터의 응용 분야에 따른 성능 효과는 좀더 많은 시간을 거쳐 검증이 이루어져야 할 것이다.

일관성 유지 캐쉬를 채용한 구조를 CC-NUMA(Cache Coherent Non Uniform Memory Access) 구조라고 이야기 하며 한편 CC-NUMA에서 메모리 단계에서도 그 기능을 캐쉬 처럼 확장하여 프로세서의 참조 요구에 따라 데이터들이 지역메모리 간에 동적으로 이동할 수 있는 COMA(Cache Only Memory Architecture) 구조도 대표적인 구조중 하나로 여겨지고 있다[5]. 대표적인 CC-NUMA 구조로는 Stanford의 DASH와 FLASH, MIT의 Alewife, 그리고 상용 제품으로 Convex사의 Exemplar 등이 있으며[4, 7, 8, 9] COMA 구조로는 SICS의 DDM, 상용 제품인 KSR의 KSR-1 등이 있다[5]. 이외에 CRAY의 T3D 또한 분산 공유메모리 구조이나 일관성 유지 캐쉬를 사용하지 않는 구조이다[10].

분산 공유메모리 구조에서 원격 메모리 지연의 가장 주요한 원인은 상호연결망에서의 지연 시간이다. 따라서 고속의 상호연결망 구현이 가장 근본적인 문제로 남아있다.

2.3.2 분산 다중컴퓨터(Distributed Multicomputer)

분산 다중컴퓨터는 기존의 소결합 구조(loosely coupled architecture) 또는 NORMA(No Remote Memory Access) 구조에 대한 다른 표현이라고 할 수 있다. 즉 독립적인 여러대의 컴퓨터들을 고속의 상호연결망으로 연결하고 이들간에 메시지를 주고 받음으로서 상호연동하게 하는 형태의 병렬 처리 환경을 갖는다. 따라서 컴퓨터들 간의 공유된 형태의 메모리는 존재하지 않으며 밀결합 공유메모리 구조에서와 같이 자원의 공유에 의한 병목 현상이 거의 존재하지 않는다.

이러한 구조는 확장성이 좋은 구조로 인식되어 왔으며 따라서 초기의 MPP는 대부분 분산 다중컴퓨터의 형태를 취해왔다. 그러나 병렬 처리를 위하여 사용되는 메시지 전달 방식의 프

로그래밍 모델은 일반 사용자들에게 공유메모리 방식에 비해 어려운 면이 있으며 또한 컴퓨터 간에 주고 받는 메시지의 양이 응용 프로그램의 코드 및 데이터를 각 컴퓨터에 어떻게 분할하여 할당하느냐에 따라 많은 차이가 나게 된다. 이는 NUMA 구조에서도 마찬가지 이지만 특히 분산 다중 컴퓨터의 경우에는 메시지의 전송 시간에 상호연결망에서 지연 시간뿐 아니라 소프트웨어처리 시간도 포함되기 때문에 성능에 더욱 더 많은 영향을 미치게 된다. 따라서 메시지 전달에 걸리는 시간을 줄이기 위한 하드웨어 및 소프트웨어의 구조에 대한 많은 연구들이 수행되어 왔다.

한편 이러한 분산 다중컴퓨터 구조에서 가장 중요한 문제로 거론되는 것은 단일 시스템 이미지(single system image)의 제공이다. 이는 여러대의 컴퓨터가 연결되어 이루어진 전체 시스템 자원들을 하나의 컴퓨터 내의 자원들을 사용하는 것처럼 할 수 있도록 하게 하는 것이며 그 궁극적인 목표는 응용 프로그램 작성자에게 분산 다중컴퓨터가 단일 프로세서로 동작하는 시분할 시스템처럼 보일 수 있도록 하는 것이다[11]. 이에 대한 연구는 기존의 분산 처리(distributed computing)분야에서 많이 수행되어 왔으며 주변 장치의 공유, 화일 시스템의 공유등에서 부터 현재는 각자 컴퓨터 내부의 전용메모리(private memory)들을 공유메모리 처럼 사용할 수 있게 하는 가상 공유메모리(Virtual Shared Memory)에 대한 연구들이 활발히 진행되고 있다. 이러한 기능들은 대부분 운영 체제 내부에서 해결하도록 하고 있으나 완벽한 단일 시스템 이미지의 구현은 아직은 해결되어야 할 문제로 남아있다. 따라서 분산 다중구조에서는 시스템 내부의 자원들을 효율적으로 이용하기 위해서는 사용자가 어느 정도 전체 시스템의 구성에 대한 이해가 있어야 한다.

기존의 UNIX 운영 체제 상에서 분산 처리 환경을 지원하기 위한 것으로는 OSF의 DCE가 있으며 분산 처리를 위한 운영 체제로 Amoeba, Mach, Chorus 등이 있다[11]. 이러한 분산 운영 체제(distributed operating system)들은 마이크로 커널을 기반으로 한 시험적인

운영 체제로 아직은 상용 시장에서의 기반이 구축되어 있지는 않은 상태이나 UNIX와 같은 통합 커널 기반의 운영 체제보다 클라이언트/서버 환경에 적합한 특성들을 가지고 있어 향후 운영 체제의 주도권을 가지게 될 것으로 보인다.

분산 다중컴퓨터 구조에서 병렬 처리 프로그램의 성능을 결정하는 가장 중요한 요소로는 메시지 처리를 위한 소프트웨어 처리 시간과 상호 연결망에서 메시지 전달 지연 시간이다. 소프트웨어에 의한 성능 부담을 덜기 위하여 많은 컴퓨터에서 메시지 처리 전용 프로세서를 두고 있으며 대량의 데이터의 고속 전송을 위한 DMA와 같은 특별한 하드웨어를 사용하고 있다. 한편 메시지 전달 지연 시간은 상호 연결망의 특성에 크게 좌우된다. 상호 연결망의 특성은 연결망의 구조, 매체의 물리적 특성, 메시지 전송을 위한 스위칭 방법등에 의해 결정된다[2]. 상용화된 분산 다중 컴퓨터로 대표적인 시스템은 IBM의 SP2, Tandem의 Himalaya와 AT&T의 GIS 등이 있다[12, 13].

3. MPP의 응용

MPP의 본격적인 보급은 80년대 후반부터 이루어졌으며 현재까지 많은 MPP업체들이 다양한 구조의 MPP를 제공하면서 나름대로 시장에서의 성공 여부를 검증받아 왔다. 93년도까지 전세계에서 가장 강렬한 연산 능력을 갖는 500대의 슈퍼 컴퓨터중 MPP 구조가 약 37%를 차지하고 있다[14]. 대부분의 MPP의 보급은 미국에서 주도적으로 이루어졌으며 또한 MPP 제조업체의 대부분이 미국 회사들이다. 이들은 대부분 과학 계산용으로 사용되고 있으며 이는 냉전 시대에 과학 및 공학에서의 첨단 기술을 이용한 군사적 우위를 유지하기 위한 미국 정부가 시험적인 MPP 제조업체나 구매 기관에 대하여 정책적 지원을 하였기 때문이다. 그러나 냉전이 종식된 지금 이들에 대한 지원도 대폭 감소되었으며 따라서 많은 MPP 제조업체들이 상용(business) 시장으로 눈을 돌리고 있는 실정이다[15].

한편 최근들어 반도체 기술의 발달에 힘입어

마이크로프로세서의 성능은 비약적으로 발전하고 있으며 따라서 많은 MPP에서 자체 설계에 의한 고유의 프로세서 보다는 고성능의 범용 마이크로프로세서를 사용하고 있다. 또한 운영 체제도 개방 환경을 지향하는 표준 상용 제품인 UNIX가 대부분 사용되고 있다. 이러한 결과 MPP는 성능의 확장성, 가격대 성능비, 개방성, 범용성 등에 있어서 매우 뛰어난 구조로 알려지고 있으며 이러한 특성은 클라이언트/서버 환경에서 고성능 서버에 매우 적합한 구조라고 할 수가 있다. 따라서 MPP의 응용 분야는 과학 계산용을 위한 고성능 연산 서버이외에 병렬 데이터베이스를 탑재하여 OLTP, MIS, DSS 등의 상용 목적으로 사용되는 고성능 상용 서버(Enterprise Server)로도 많이 이용되고 있으며 또한 향후 멀티미디어 서버로서 다양한 형태의 정보를 고속으로 처리할 수 있는 구조로 각광을 받을 것으로 예상되고 있다.

3.1 고성능 연산 서버(Computation Server)

초창기 MPP는 방대한 연산 능력을 필요로 하는 과학 및 공학적 문제 해결을 위한 고성능 연산 서버로 사용되어 왔다. 이러한 분야에서의 가장 대표적인 응용은 컴퓨터를 이용한 모의 실험(simulation)이다. 해당 분야의 이론적 배경에 의한 수학적 모델을 모의 실험을 위한 기본 알고리즘으로 구현하고 실제 시험을 통하여 얻어진 데이터들을 도입하여 모델을 검증할 수도 있고 실제 상황에서는 적용할 수 없는 극단

적인 조건들을 모의 실험에 적용하여 그 결과를 미리 예측하는데 이용할 수도 있다. 또한 비행기의 조종이나 발전소의 응용 등과 같은 특수 기술을 익히는데도 모의 실험이 이용된다. 표 1은 고성능의 연산용 서버로 제공되는 MPP인 IBM의 SP2와 CRAY의 T3D에서 대표적인 응용 분야로 제시한 것들이다[12, 16].

3.2 고성능 상용 서버(Enterprise Server)

관계형 데이터베이스를 이용한 OLTP(On Line Transaction Processing)와 DSS(Decision Supporting System)분야는 최근들어 수요가 급증하고 있는 분야이다. 이러한 상용 응용 분야를 위한 컴퓨터는 연산 능력도 중요하지만 특히 입출력 장치에서 빠른 응답 시간과 대량의 데이터 전송 능력, 대량의 데이터 저장 및 관리 능력들이 지원될 수 있는 구조이어야 한다. MPP는 연산 능력에서 뿐만이 아니라 데이터 전송 능력 및 저장 관리 능력에 있어서도 확장성이 보장되는 구조이다. 또한 응용 프로그램에서 제기되는 트랜잭션이나 질의어(query) 수준의 작업에 대하여 대규모 병렬성을 쉽게 지원할 수 있다. 따라서 많은 데이터베이스 업체들이 MPP를 기반으로하여 대규모 데이터베이스(Very Large Database)의 구축이 용이한 고성능 병렬 데이터베이스 제품들을 출시하고 있는 추세이다.

병렬 데이터베이스를 위한 하드웨어의 기반 구조는 각각의 프로세서들간의 메모리와 디스크같은 데이터 저장장치를 공유하는 방식에 따

표 1 고성능 연산 서버용 MPP의 응용분야

IBM SP2	Cray T3D
Aerospace	Chemistry and chemical engineering
Automotive and manufacturing	Oil and gas industry
Chemical and pharmaceutical	Computational electromagnetics
Computational fluid dynamics	Semiconductor Manufacturing
Crashworthiness analysis	Atmospheric modeling
Electronic analysis	Scientific visualization and image processing
Molecular modeling and molecular dynamics	
Petroleum exploration and production	
Photolithography	
Quantum chemistry	
Seismic processing	
Semiconductor device simulation	

라 다음의 3가지로 나누어 진다[13].

(1) Shared Everything : 프로세서들이 메모리와 디스크를 모두 공유하는 구조이다. 이 경우 병렬로 실행되는 모든 사건들은 하나의 데이터베이스 버퍼와 하나의 논리적인 디스크를 이용할 수 있다. 프로세서들이 논리적으로 하나의 버퍼만을 이용하므로 버퍼 관리가 용이하다. 밀결합 다중프로세서 구조를 기반 구조로 하며 따라서 메모리나 버스, 디스크등에서 병목 현상이 발생하게 되어 확장성에 제약을 받는 것으로 알려져 있다. 그러나 분산 공유메모리 구조의 MPP에서도 메모리와 디스크에 대한 공유가 가능하기 때문에 이 경우 확장성에 대한 제약이 없으면서 대규모의 병렬 데이터베이스의 구현이 용이해질 수 있다. 현재 분산 공유메모리 구조의 MPP에서는 KSR-1에 Oracle Ver.7이 탑재된 예가 있다.

(2) Shared Disk : 각자의 프로세서가 자신의 전용메모리를 가지면서 디스크만 공유하는 구조이다. 이 경우 데이터베이스 버퍼는 각 프로세서의 전용메모리 마다 존재하게 되며 하나의 논리적인 디스크가 존재하게 된다. 특정 프로세서가 고장이 난 경우라도 다른 프로세서에 의해 데이터베이스내의 모든 데이터를 사용할수가 있기때문에 높은 가용성을 가지는 구조가 될 수 있다. 그러나 각각의 데이터베이스 버퍼에 동일한 데이터 블록의 여러 이미지가 동시에 존재할 수가 있게되며 따라서 이들간의 일관성 유지 작업이 복잡해진다. 또한 디스크에서의 병목 현상으로 인한 확장성에 제약을 받는 구조이다. 이는 밀결합 다중 프로세서 단계에서 클러스터 단위의 소규모 확장을 용이하게 하며 동시에 가용성을 높일 수 있는 방식이다. 현재 Oracle의 OPS(Oracle Parallel Server)가 이러한 용도로 사용되고 있다. OLTP에서의 성능이 좋은 것으로 알려져 있다.

(3) Shared Nothing : 각자의 프로세서는 자신의 전용메모리와 전용의 디스크를 가지며 따라서 물리적으로는 아무 것도 공유되지 않는 구조이다. 일종의 분산 데이터베이스라고 할수가 있으며 하나의 데이터 블록은 자신이 속한 프로세서의 데이터베이스 버퍼에만 존재하기 때문에 일관성 유지와 같은 절차가 필요치 않

게 된다. 확장성이 뛰어난 구조이나 특정 프로세서의 고장시 그쪽에 연결된 데이터를 이용을 보장할 수 있는 장치가 필요하다. 또한 프로세서 간의 데이터 이동이 빈번하여 상호연결망에 많은 부담을 줄 수 있는 구조이다. DSS에 적합한 구조로 알려져 있으며 Teradata와 Sybase의 Navigation Server 등의 제품이 있으며 Informix에서 진행중인 XMP project도 이러한 구조로 알려져 있다. 분산 다중컴퓨터 구조에 적합한 형태이다.

3.3 멀티미디어 서버(Multimedia Server)

최근에 선진 각국에서는 정보화 시대로의 신속한 진입을 위하여 초고속 정보통신망 사업에 많은 심혈을 기울이고 있다. 이러한 고속 통신망의 구현은 공중망을 통하여 데이터, 음성뿐 아니라 영상 정보까지도 자유롭게 주고받을 수 있게 해주며 이러한 멀티미디어 데이터를 이용하여 교육, 오락, 문화, 보건등등의 사회 제 분야에서 새로운 차원의 시장을 창출해 내고 있다. 아래의 예는 멀티미디어 통신을 이용한 대표적인 응용의 예이다[17].

- VOD(Video on Demand) Service
- Home shopping Service
- Remote Education System
- Remote Medical Service
- Teleconference
- Electronic Library/Museum Services
- Electronic Newspaper
- Public Information Kiosk Service
- Securities and Financial Information Service

이러한 응용 환경의 구축은 사용자에게 서비스를 제공하는 클라이언트와 서비스를 위한 멀티미디어 데이터를 관리하고 제공하는 서버, 그리고 이들을 연결하는 고속의 통신망으로 이루어진다. 클라이언트로선 PC나 STB(Set-Top Box) 등이 이용되며 서버는 불특정 다수의 클라이언트의 요구에 따라 지속적인 흐름을 갖는 영상 및 음향 데이터(data stream)를 적절하게 제공할 수 있는 컴퓨터를 이용한다. 이러한 멀티미디어 서버로 사용되는 컴퓨터는 막대한 용량의 저장 장치를 관리할 수 있고 또한 동시

성(isochronous)을 지닌 초고속 통신 능력을 보유하고 있어야 한다. MPP는 이러한 요구를 수용할 수 있는 최적의 구조로 따라서 향후 수 년내에 멀티미디어 서버로서의 MPP에 대한 시장 수요가 급증 할것으로 예측되고 있다.

4. 국내 MPP 현황

국내에서는 KAIST와 포항 공대에서 과학계산용의 MPP를 연구용으로 설계 제작한바 있으나 상용 제품으로 사용할 수 있는 MPP가 개발된 예는 아직 없다. 또한 지금까지 MPP에 대한 수요도 그리 많지는 않았다. 그러나 최근의 컴퓨터 환경의 변화는 상업용 시장에서 메인프레임을 대체할 MPP의 수요가 점차 증가할 것으로 생각되며 또한 산업 구조의 발전에 따라 발전소, 자동차, 항공기, 반도체 설계등과 같이 고성능 연산 서버를 필요로 하는 분야도 계속 확대되어질 것으로 보인다. 이러한 국내 수요들에 대처하고 또한 향후 국제 시장에의 진입을 위하여 현재 정부의 지원 아래 국내 연구소와 기업들의 참여에 의한 두 가지의 대형 컴퓨터 개발 프로젝트가 진행중이며 이들 모두

MPP 구조를 기본 구조로 채택하고 있다.

4.1 정보통신부 주전산기IV

주전산기IV는 정보통신부의 지원 아래 한국 전자통신연구소와 금성사, 대우통신, 삼성전자 및 현대전자가 94년 2월부터 98년 1월까지 공동으로 개발 중인 MPP 구조의 병렬처리 컴퓨터이다. 병렬 데이터베이스를 탑재하여 OLTP 나 DSS와 같은 상업용 시장을 목표로 하고 있으며 기본적인 구조는 분산 다중컴퓨터 구조로 10×10 구조의 크로스바 스위치상에서 결합된 밀결합 다중프로세서¹를 16개까지 Xcent-Net 이라는 고속의 상호연결망에 연결하는 형태를 취하고 있다. Xcent-Net은 패킷 스위칭(packet switching)방식의 데이터 전송을 지원하고 10×10 크로스바 스위치를 이용하여 구성된 일종의 MIN(Multistage Interconnection Network) 형태의 상호연결망이다. 최고 256개까지의 프로세서 확장이 가능하며 성능 목표는 최대 20 GIPS 이상이고 특히 데이터베이스의 처리 속도는 10,000 tpmC(TPC-C 기준) 이상

¹ ETRI에서는 이를 클러스터라 부름

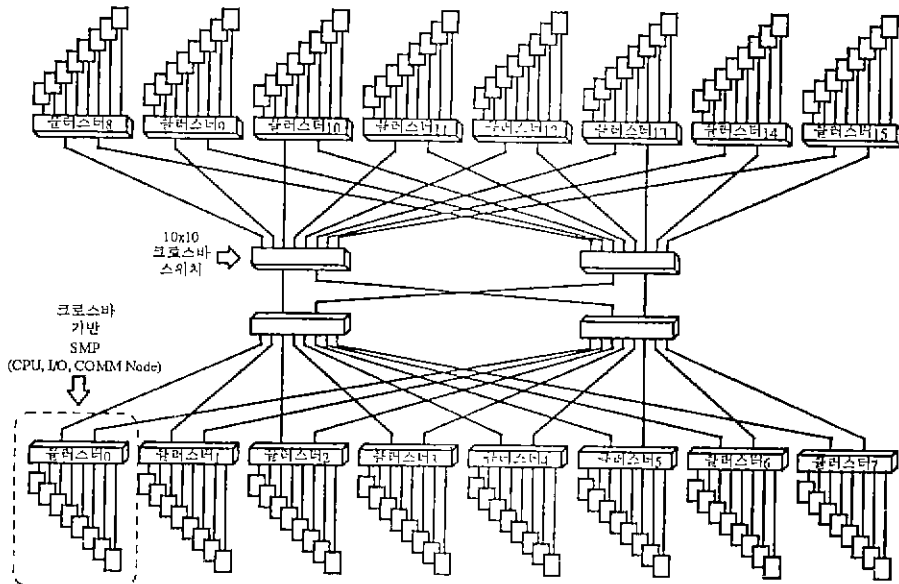


그림 1 주전산기IV의 구조

을 제공할 예정이다. 프로세서는 Intel의 P6를 사용하며 운영체제는 Chorous 마이크로코널에 UNIX와 같은 API를 제공하는 형태를 채택할 예정이다[18, 19].

4.2 통상 산업부 대형컴퓨터

통상산업부의 지원을 받고있는 대형컴퓨터 개발 과제는 서울대 컴퓨터 신기술 공동연구소와 삼성전자, 현대전자의 참여하에 진행되고 있으며 외국의 선진 대형컴퓨터 제조업체로부터 기술 전수를 받아 조기에 이를 국산화하고 개량함으로써 국제 시장에서 경쟁력 있는 국산 대형 컴퓨터를 만드는 것을 목표로 하고 있다. 기술 전수 기종으로 AT&T의 GIS(General Information Solution) 기종이 선택되었으며 97년 3월까지 개량 기종의 개발 완료를 목표로 현재 이에 대한 기술 전수가 진행 중이다.

AT&T의 GIS 시스템은 기존의 NCR 3600 기종의 개량 기종으로 Teradata, Oracle Parallel Server, Sybase Navigation Server 등의 병렬 데이터베이스를 탑재하여 OLTP나 DSS 같은 상업용 시장을 목표로 하는 제품이다. 프로세서는 현재 Intel Pentium 90Mhz가 사용되고 있으며 향후 P6를 지원할 예정이다. 기본적인 구조는 주전산기 IV와 마찬가지로 클러스

터 형태의 여러 개의 밀결합 다중프로세서를 고속의 상호연결망을 이용하여 연결하는 분산 다중컴퓨터 구조이다.

그림 2에서 볼 수 있듯이 기본 구성은 UNIX를 탑재한 밀결합 다중프로세서와 고속의 상호연결망인 BYNET, 그리고 디스크 어레이로 이루어진다. 고수준의 가용성을 위하여 BYNET은 이중 구조를 갖도록 하고 있다. BYNET은 8×8 스위치를 이용한 Folded Banyan Network 형태의 MIN으로 현재는 노드 당 10MB/Sec의 데이터 전송률을 지원한다. BYNET에 연결되는 밀결합 다중프로세서(하나의 노드)는 32개까지의 프로세서를 탑재할 수 있으며 BYNET에는 최대 64개까지의 노드가 연결되어 총 2048개까지의 프로세서가 탑재될 수 있다². 운영체제는 UNIX SVR4에 RAS(Reliability, Availability, Serviceability) 기능이 추가된 AT&T의 UNIX가 탑재된다[13, 20].

결론

최근 병렬 처리 컴퓨터의 주종을 이루는 MPP

² 하드웨어의 규격은 AT&T에서 upgrade가 계속 진행중인 관계로 변경이 있을 수 있음

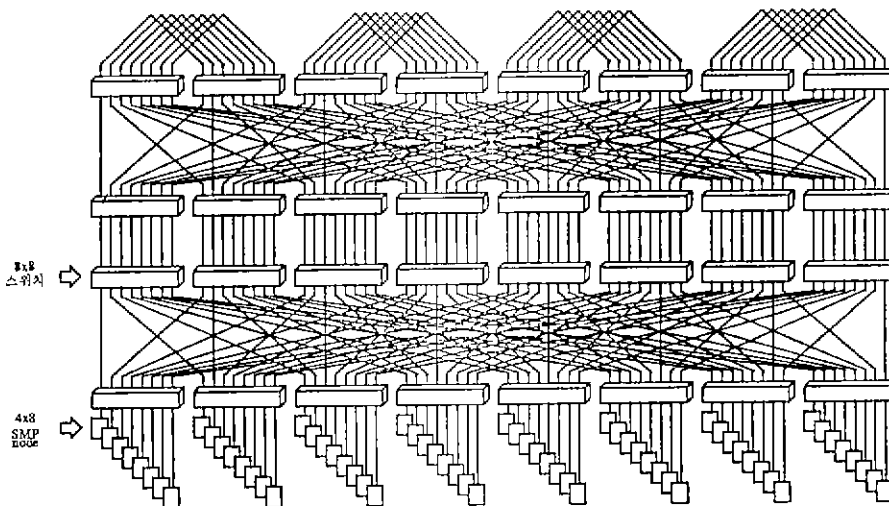


그림 2 Banyan Network을 이용한 GIS 시스템 구성(64노드, 단일 BYNET의 경우)

의 구조와 특성들, 그리고 응용 분야에 대해서 간략하게 알아보았다. 21세기의 정보화 시대에서의 다양한 컴퓨터 응용 분야는 기존의 컴퓨터 구조로서는 효율적인 해결책을 제공하기 어려운 면이 있으며 특히 초고속 정보 통신망등의 통신수단을 뒷받침하고 다양한 형태의 멀티미디어 정보를 효율적으로 처리할 수 있는 새로운 컴퓨터 구조가 요구되고 있다.

MPP 시스템은 초기에는 주로 과학계산용의 고성능 연산 서버로만 사용되어 왔지만 향후 대형 데이터베이스를 신속히 처리할 수 있는 클라이언트/서버 환경에서 OLTP, MIS, DSS 용의 고성능 상용 서버로 많이 사용될 것으로 예상되며 또한 문자, 음성, 영상등의 다양한 형태의 정보 저장 및 처리가 용이하고 대량의 정보를 고속으로 통신할 수 있는 멀티미디어 서버로도 각광 받을 것으로 예상된다. 따라서 선진국의 많은 연구 기관 및 제조업체들에서 MPP에 대한 활발한 연구 개발활동이 진행되고 있으며 그 결과 향후의 MPP는 다음과 같이 발전될 것이며 경쟁력 있는 새로운 컴퓨터 구조로 여겨지게 될 것이다.

첫째, MPP용 프로세서로는 고성능의 상용 프로세서를 사용하여 기존의 슈퍼컴퓨터나 메인프레임 등에 대한 가격대 성능비를 향상 시켜나갈 것이다. 최근의 고성능 프로세서들은 슈퍼스칼라(superscalar), 슈퍼파이프라인(super-pipeline)등의 기술이 적용되고 있으며 향후 VLIW(Very Long Instruction Word) 기술까지 접목될 것으로 보인다. 기능상으로는 멀티미디어 데이터를 신속히 처리할 수 있는 기능들이 지원되기 시작하고 있으며 집적도의 향상으로 향후에는 하나의 칩안에 다중프로세서 구조 자체가 내장되리라는 예측도 있다. 특히 MPP 구조에 필수적인 고속의 IPC(Inter-Processor Communication), 동기화(synchronization) 기능이 지원될 것이며 이에 따른 상용 MPP 구조의 표준화도 생각해 볼 수 있을 것이다. 또한 프로세서들은 매우 큰 주소 공간(address space)을 제공하여 메모리 집적도와 보조를 맞추면서 지금보다 쉽게 대용량의 MPP를 개발할 수 있게 될 것이다³. 이러한 고성능의 상용

프로세서의 사용으로 MPP는 가격대 성능비를 현격히 개선해 나갈것으로 보인다.

둘째, 우수한 확장성을 효율적으로 제공할 수 있는 상호 연결망 구조의 연구 개발이 활발히 이루어지며 상용적인 표준화 구조가 점차적으로 정립되어 갈것이다. 상호 연결망의 표준은 쉽고 값싸게 확장 가능하면서 범용성을 가진 구조가 됨으로써 더욱 MPP의 상용화가 용이하게 될 것이다. 또한 이런 상호연결망을 통한 고속 전송이 가능한 스위칭 매커니즘이 개발 발전되며 따라서 상호연결망에서의 메시지 전송 지연 시간을 극소화 하면서 메시지 처리율을 극대화 할 수 있게 될 것이다.

셋째, MPP 시스템은 사용자에게 완전한 범용성을 제공하고 사용하기 쉬운 컴퓨터라는 인식을 줄 수 있는 충분한 단일 시스템 이미지를 제공하게 될 것이다. 또한 개방형 구조를 가지고 있어 많은 미들웨어(middleware), 그룹웨어(groupware), 유틸리티(utility) 및 다양한 응용 소프트웨어가 쉽게 제공될 것이다. 특히 MPP 시스템의 장점을 살리기 위한 병렬 DBMS가 많이 발전하여 상용 응용 분야에서 많이 사용될 것이다.

넷째, 고속 정보 처리를 할 수 있는 통신 및 입출력 장치에서의 병렬 처리 기법의 향상으로 기존 컴퓨터가 처리하기 어려운 멀티미디어 응용 분야를 해결해 나갈것으로써 향후 변화되는 컴퓨터 환경을 용이하게 대처할 수 있도록 발전 개발될 것이다.

이러한 정보화 시대의 변화에 대처하기 위해서 우리나라에서도 정부의 지원 아래 진행중인 MPP 구조의 대형 컴퓨터 프로젝트를 성공적으로 완수하여야 하며 이를 바탕으로 국제 시장에 진입함으로써 정보화 시대에서 선진국 대열에 들 수 있도록 배전의 노력을 하여야 할 것이다.

참 고 문 헌

- [1] Michel Dubois, Christoph Scheurich, Faye A. Briggs, "Synchronization, Coherence, and Event Ordering in Multiprocessors," *IEEE Computer*, Vol. 21, No. 2, pp. 9-21,

³ 현재 최고 64 비트까지 지원되고 있음

Feb. 1988.

[2] Kai Hwang, *Advanced Computer Architecture : Parallelism, Scalability, Programmability*, McGraw-Hill, 1993.

[3] W. Daniel Hillis and Lewis W. Tucker, "The CM-5 Connection Machine : A Scalable Supercomputer," *Communications of the ACM*, Vol. 36, No. 11, pp. 31-40, Nov. 1993.

[4] Daniel Lenoski, James Laudon, Kourosh Gharachorloo, Wolf-Dietrich Weber, Anoop Gupta, John Hennesy, Mark Horowitz, and Monica S. Lam, "The Stanford DASH Multiprocessor," *IEEE Computer*, pp. 63-79, Mar. 1992.

[5] Per Stenstrom, Truman Joe, and Anoop Gupta, "Comparative Performance Evaluation of Cachecoherent NUMA and COMA Architectures," *Proceedings 19th ISCA*, pp. 80-91, 1992.

[6] *L64601 Scalable Coherent Interface Node Chip Technical Manual*, LSI Logic, 1994.

[7] Jeffrey Kuskin, David Ofelt, Mark Heinrich, John Heinlein, Richard Simoni, Kourosh Gharachorloo, John Chapin, David Nakahira, Joel Baxter, Mark Horowitz, Anoop Gupta, Mendel Rosenblum, and John Hennesy, "The Stanford FLASH Multiprocessor," *Proceedings 21th ISCA*, pp. 302-313, 1994.

[8] *Exemplar Architecture, 1st edition*, Convex Corporation, 1994.

[9] Anant Agarwal, Beng-Hong Lim, David Krantz and John Kubiawicz, "APRIL : A Processor Architecture for Multiprocessing," *Proceedings 17th ISCA*, pp. 104-114, 1990.

[10] *CRAY T3D System Architecture Overview Manual*, Cray Research, Inc., 1993.

[11] Andrew S. Tanenbaum, "Distributed Operating Systems," Prentice Hall, 1995.

[12] *IBM POWERparallel solutions*, IBM Corporation, June 1995.

[13] 병렬 RDBMS 구조, AT&T GIS Presentation Material, 1995.

[14] Jack J. Dongarra, Hans-Werner Meuer, Erich Strohmaier, "TOP500 Report 1993,"

Univ. of Mannheim, 1993.

[15] Borko Furht, "Parallel Computing : Glory and Collapse," *IEEE Computer*, pp. 74-75, Nov. 1994.

[16] *Applications Update on the CRAY T3D System*, Cray Research, Inc., Nov. 1994.

[17] *Technology Background : Intel Scalable Multi-server Technology for Interactive Multimedia applications*, Intel Corporation, May 1994.

[18] 고속병렬컴퓨터 시스템 설계서 Ver. 1.0, 한국전자통신연구소, 12월 1994.

[19] 고속병렬컴퓨터 공동 연구개발사업 제4차 Workshop 자료, 한국전자통신연구소, 3월 1995.

[20] *RIACT Program Update*, AT&T GIS, 5월 1995.

박 승 권



1985 한양대 전자통신공학과 학사
 1995 한국과학기술원 정보 및 통신공학과 석사
 1985~현재 삼성전자 선임연구원
 관심분야: 병렬처리, 컴퓨터 구조

유 승 화



1972 서울대 공대 응용수학 학사
 1983 미 KANSAS대 전산학 박사
 1974~1976 KIST 시스템 프로그래머
 1976~1978 삼성통신(주) 연구실장
 1983~1988 미 AT&T Bell Lab 연구원
 1988~1989 미 Amdahl사 수석 연구원
 1989~1992 삼성전자(주) 이사
 1993~현재 삼성전자(주) 상무 이사

관심분야: 병렬처리 컴퓨터 구조, 운영체제