□조사연구□

담당편집위원 · 한국방송통신대학교 전자계산학과 곽덕훈 교수 Tel · 02-7404-652, Fax 02-7404-208

상용 병렬처리 시스템의 비교 분석*

컴퓨터시스템연구회

목 차

1. 서 론
2. 병렬처리 시스템 분류
3. 상용 병렬처리 시스템
3.1 CM-5[3]
3.2 KSR-1[4]
3.3 Convex-Meta[6]
3.4 NCR 3600[7]
3.5 NCUBE-2[8]
3.6 Paragon[9]
3.7 병렬처리 시스템 비교
4. 결 론

1. 서 론

우리 주변에는 현존하는 최고 성능의 컴퓨터로도 시간이 너무 걸려 풀수 없는 문제들이 많이 있고, 또한 앞으로 고성능 컴퓨터의 요구는 정보화 사회의 발전과 더불어 점점 늘어갈 것이 분명하다. 지금까지 이런 문제들은 주로 슈퍼컴퓨터가 해결하였다. 그러나 슈퍼컴퓨터가 너무 고가이기 때문에 가능하면 가격이 비교적 싼 컴퓨터의 필요성이 대두되고 있다. 현재까지 병렬처리형 컴퓨터가 가장 적합한 해결 방법으로 알려져 있고, 따라서 병렬처리 컴퓨터에 대한 연구와 상용화에 많은 노력들을 하고 있다. 그리고 지금까지는 대부분의 병렬처리 컴퓨터가 과학 계산용으로 사용되어 왔지만 최근에는 상업 자료 처리용으로 발전하고 있다. 이런 추세는 병렬처리 기술이 점점 범용으로 발전되고 있음을 말해 주고 있다. 이렇게 병렬처리 컴퓨터를 발전하게 하는 기술 요인들로는 다음과 같은 것들이 있다.

1) 마이크로프로세서 기술의 발전 : 성능 증가, 크기 감소, 가격 감소
2) 고속 상호연결망 기술의 발전 : 고속, 저지연 상호연결망 기술
3) 소프트웨어 기술의 발전 : 병렬 알고리즘, 병렬 운영체제, 병렬 컴파일러

앞으로 이런 기술의 발전은 병렬처리 컴퓨터 개발을 더욱 가속 시킬 것으로 예상되지만, 아직은 성숙 단계에 있지 못하기 때문에 주도적인 병렬처리 구조나 기술이 없다. 지금까지 개발된 시스템들은 특정 응용 분야에 적합하도록 특수한 구조를 갖고 있다. 따라서 병렬처리 컴퓨터를 설계할 때 고려해야 할 중요한 설계 변수로는 다음과 같은 것들이 있다.

첫째, 프로세싱 요소(처리기)를 어떻게 설계할 것인가?

1) 얼마나 많은 처리기를 둘 것인가?
2) 각 처리기의 성능은 어느 정도로 할 것인가?
3) 각 처리기당 메모리 크기는 어느 정도로 할 것인가?
4) 각 처리기당 입출력 밴드 폭은 어느 정도로 할 것인가?
5) 어떤 기능을 지원할 것인가? : 정수연산, 부동소숫점 연산, 벡터연산, 정렬(sorting)
6) 어떻게 패키징할 것인가?

둘째, 통신 또는 상호연결망의 구조와 성능은 어떻게 설계할 것인가?

1) 어떤 통신 프로토콜을 사용할 것인가? : synchronous, asynchronous, packet switching, circuit switching

*본 기사는 한국정보과학회 컴퓨터시스템연구회에서 학회의 지원을 받아 수행한 연구 결과임

2) 어떤 상호연결망 구조를 가질 것인가? : topology
3) 어떤 종류의 전송 매체를 쓸 것인가? : electric, optic
4) 어느 정도의 전송속도가 필요한가? : frequency, bandwidth, latency

셋째, 응용 분야에 따른 병렬처리 방법을 어떻게 할 것인가?

1) 어떤 프로그래밍 모델을 지원할 것인가? : shared memory, message passing
2) 처리기들 사이의 동기화는 어떻게 지원할 것인가? : lock, semaphore, barrier, critical section
3) 각 처리기에 할당될 작업의 크기는? : job, process, thread, instruction
4) 작업의 분할은 어떻게 할 것인가? : static, dynamic
5) 소프트웨어와 하드웨어의 역할 분담은 어느 정도로 할 것인가?

넷째, 다음과 같은 사항도 충분히 고려되어야 한다.

1) 어떤 프로그래밍 언어와 소프트웨어 개발 도구를 지원할 것인가?
2) 병렬성을 어디까지 보장할 것인가? : user, compiler, operating system, hardware
3) 가용성 지원의 정도는? : availability
4) 신뢰성 지원의 정도는? : reliability

본 논문에서는 위에서 열거한 설계 변수들을 기준으로 현재까지 상용으로 개발된 병렬처리 시스템을 조사하여 각 시스템의 특성을 비교 분석하였다. 제2장에서는 병렬처리 시스템들을 특성에 따라 분류하는 방법에 대해 기술하였다. 제3장에서는 현재까지 개발된 주요 상용 병렬처리 시스템들에 대해 조사 분석하였다. 끝으로 제4장에서는 병렬처리 시스템 발전의 문제점과 앞으로의 방향에 대해서 언급하였다.

2. 병렬처리 시스템 분류

다중처리 혹은 병렬처리 시스템을 분류하는 데에는 크게 3가지 방법이 있다. i) 명령어와 데이타 흐름으로 분류하는 방법, ii) 메모리 구조와 프로세서 동기화 기법으로 분류하는 방법, iii) 메모리 접근 기법으로 분류하는 방법이 있다.

첫째, 명령어와 데이타 흐름에 의한 분류 방법은 Flynn이 처음 제안하였다. [1] Flynn은 명령어(instruction) 흐름과 데이터(data) 흐름으로 컴퓨터 시스템을 다음과 같이 분류하였다.

1) SISD : Single-Instruction, Single-Data : uniprocessor
2) SIMD : Single-Instruction, Multiple-Data : array and systolic machine
3) MISD : Multiple-Instruction, Single-Data : 현존하는 컴퓨터는 없음
4) MIMD : Multiple-Instruction, Multiple-Data : general purpose multiprocessor

둘째, Johnson은 Flynn의 MIMD 구조를 더욱 세분화하여 메모리의 하드웨어적 구조(memory structure)와 프로세서 사이의 동기화(synchronization) 관점에서 다음과 같이 분류하였다. [2] 즉, 물리적으로 메모리 구조가 집중되어 있는지 아니면 분리되어 있는지와 프로세서간의 동기화를 위해 메세지 전달 방식을 사용하는지 아니면 공유메모리 방식을 사용하는지로 구분하였다.

1) GMSV : Global Memory Shared Variables : shared memory machine
2) DMSV : Distributed Memory Shared Variables : hybride machine
3) DMMP : Distributed Memory Message Passing : message passing machine
4) GMMP : Global Memory Message Passing

GMSV는 프로세서와 메모리가 상호연결망에 의해 분리되어 있는 형태로 일반적인 다중프로세서인데 대표적인 것으로는 NYU Ultracomputer가 있다. 이런 시스템들은 메모리 충돌(memory contention)문제를 해결하기 위해 캐쉬를 사용하고, 프로세서와 메모리 사이에 고성능 상호연결망을 사용한다. DMSV는 메모리가 여러 곳에 분산되어 있는 형태이면서 공유메모리 모델을 지원하는 경우인데 대표적인 경우로 BBN Butterfly가 있다. 이런 시스템은 성능을 높이기 위해서는 데이터와 코드(data & code)를 잘 분배시켜야 하는 문제(data placement problem)를 갖고 있다. DMMP는

메모리가 분산되어 있고 동기화를 위해 메세지 전달 기법을 사용하는 경우인데 대표적인 경우로 NCUBE가 있다. GMMP는 메모리는 집중되어 있으나 동기화를 위해 메세지 전달 방식을 사용하는 데 대표적인 경우로 ELXSI 6400가 있다.

셋째, 메모리 접근 기법 및 주소 공간 할당에 따라 다음과 같이 분류할 수 있다. 이것을 그림으로 나타내면 그림 1과 같다.

1) NORMA : NO Remote Memory Access
2) UMA : Uniform Memory Access
3) NUMA : Non-Uniform Memory Access
4) COMA : Cache-Only Memory Architecture

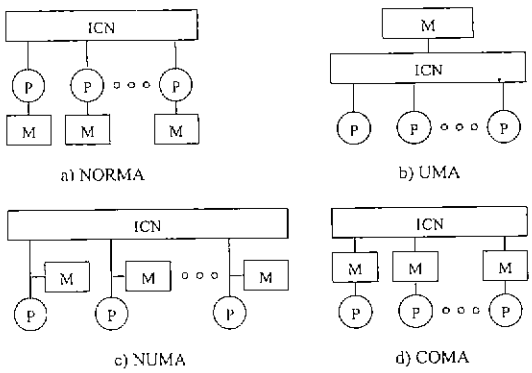

그림 1 메모리 접근 기법에 의한 분류

NORMA의 경우는 독립적인 주소공간을 갖는 프로세서-메모리 노드가 상호연결망으로 연결되어 있다. 각 프로세서는 다른 프로세서에 연결되어 있는 메모리를 직접적으로 접근(read 또는 write)할 수 없고, 필요에 따라 메세지 전달을 통해 통신이 이루어 진다. UMA의 경우는 집중된 메모리를 상호연결망을 통해 여러 프로세서가 접근할 수 있다. NUMA의 경우는 각 프로세서에 근접한 메모리(near memory 또는 local memory)는 짧은 시간에 접근 가능하지만 멀리 있는 메모리(remote memory) 접근은 긴 시간이 필요하다. 하지만 시스템내의 모든 메모리는 직접적인 참조로 접근이 가능한 공유 메모리이다. COMA는 프로세서에 가까이 있는 메모리(near memory)가 캐쉬처럼 동작하는데 각 메모리를 어트렉션 메모리(attraction memory)라 부른다. 특정 프로세서가 원하는 데이터는 해당 프로세서 가까이 있는 어트렉션 메모리로 이동 또는 복사하여 사용한다.

그외 MIMD 시스템을 단순히 다중프로세서(multiprocessors)와 다중컴퓨터(multicomputers)로 분류하기도 하는데, 전자는 단일 주소공간(single address space)을 지원하고, 후자는 각 프로세서가 독립적인 주소공간(multiple address space)을 갖고 메세지 전달 기법을 사용하는 경우이다. 그리고 다중프로세서는 더욱 세분하여 메모리가 물리적으로 분리되어 있는지 아니면 집중되어 있는지에 따라 구분하여 전자를 분산 메모리 다중프로세서(distributed memory multiprocessors)라 하고 후자를 집중 메모리 다중프로세서(central memory multiprocessors)라 한다.

3. 상용 병렬처리 시스템

3.1 CM-5 [3]

CM-5는 Thinking Machines사에서 최근에 개발한 상용 시스템이다. CM-2 이전에 나온 시스템들은 general-purpose 응용에 부적합하고 융통성이 부족한 SIMD 구조였다. CM-5는 이를 보완하여 SIMD와 MIMD의 장점을 모두 갖는 시스템 구조로 병렬 계산시 MIMD와 SIMD 특성을 모두 지원하는 동기화 구조를 갖고 있다. 그림 2는 CM-5 시스템의 구조를 보여주고 있다. CM-5는 32개에서 16,384개의 processing node를 가질 수 있고, 각 processing node는 SPARC 프로세서, 32 Mbytes 메모리, 64-bit 실수 정수 연산을 하는 128 Mflops 성능의 벡터 처리기를 갖고 있다. CM-2가 단일 sequencer를 사용하는 반면 CM-5는 여러 개의 control processors(Sun Microsystems 워크스테이션)를 사용한다. control processors의 수는 시스템 구성에 따라 수십개까지 구성할 수 있으며, 각 control processor는 필요에 따라 메모리와 디스크를 가질 수 있다. 또한 그래픽 장치, 대용량 저장 장치, 고성능 네트워크 입출력 interface를 통해 고성능 입출력 기능을 제공하며, Ethernet을 통해 저속의 입출력 기능을 제공한다. CM-5의 상호연결망 구조는 data network, control network, diagnostic network으로 구

성된다. data network는 processing node들 사이에 고속의 point-to-point 데이터 통신 채널을 제공하고, control network는 시스템 관리 기능외에 broadcasting, 동기화 등의 기능을 제공하고, diagnostic network는 모든 시스템 구성 요소를 시험하며 오류들을 검사하고 이 결과에 따라 구성 요소를 분리시킨다. data와 control network는 processing nodes와 control processors와 입출력 채널들을 연결한다. data와 control network는 확장성이 좋기 때문에 시스템 구조나 프로세서 수에 상관없이 확장할 수 있다.

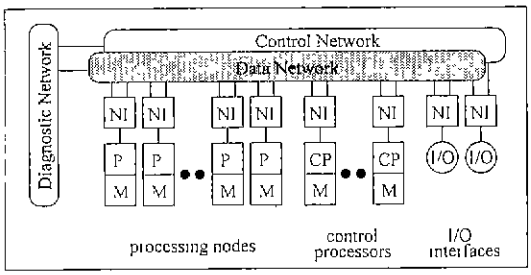

그림 2 CM-5 시스템 구조

• Control Processor

그림 3은 control processor의 구조를 나타낸다. control processor는 RISC 프로세서(CPU), 지역 메모리, 지역 디스크와 Ethernet 연결을 갖는 I/O, 그리고 control과 data network interface로 구성된다. control processor은 워크스테이션 급의 컴퓨터 시스템으로 UNIX 확장 운영체제인 CMOST를 수행한다. 또한 계산 자원과 입출력 자원을 관리한다.

• Processing Node(PN)

그림 4는 PN의 구조를 나타낸다. PN은 SPARC 프로세서, 8·16·32 Mbytes의 메모리와 메모리 제어기, network interface로 구성되고, 내부 버스는 64-비트 크기를 갖고 있다. 빠른 context switching을 위하여 multi-window 특성을 갖는 RISC 프로세서를 선택하여 processing node들의 동적인 수행을 용이하게 하였다. 그리고 벡터 유니트를 탑재할 수 있게 만

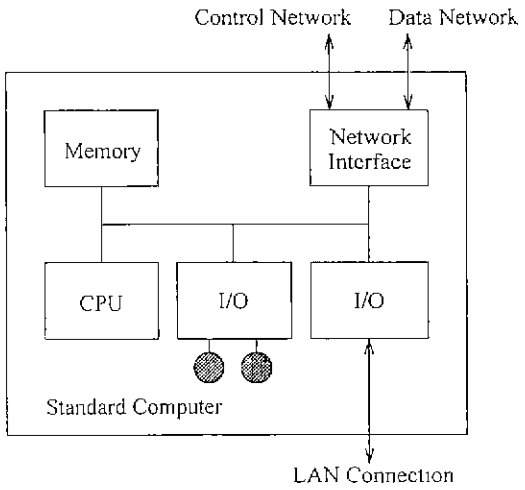

그림 3 CM-5의 Control Processor

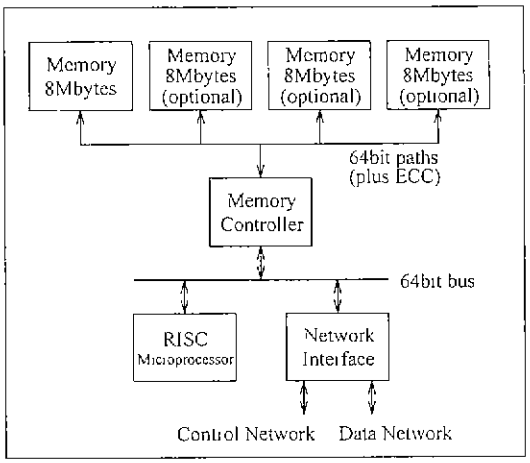

그림 4 CM-5의 Processing Node

들어 다양한 응용에 적응할 수 있는 구조를 갖고 있다.

• Data Network

CM-5의 data network는 fat-tree 구조이다. binary tree와 달리 fat-tree의 채널 능력은 뿌리(root)로 갈수록 증가한다. fat-tree의 계층적 특성은 각 계층에서 독립적인 subtree를 구성할 수 있다. CM-5의 data network는 실제 그림 5의 4-ary fat-tree의 구조로 구현되었다. 각 내부 스위치 노드들은 여러개의 라우팅 칩들로 구성되며 각 라우팅 칩은 4개의 아들(child)

칩들과 2개 혹은 4개의 부모(parent) 칩들에 연결된다. 국부성을 높히기 위해 서로 다른 subtree를 구성할 수 있으며 그 크기는 각 국부화 정도에 따라 가변적이다. 입출력 채널들도 다른 subtree를 구성할 수 있으며 이 subtree는 공유 자원으로 접근 가능하다.

그림 5의 fat-tree에서 각 채널은 20 Mbytes/s의 전송 용량을 갖고 있으며, 각 노드는 data network에 두개의 채널을 갖고 있다. 이는 각 노드의 40 Mbytes/sec 입출력 능력에 상응한다. 전송 용량은 CM-5의 최대 구성인 16,384개의 노드들에 따라 선형적으로 늘어난다. 메세지가 tree의 위로 올라감에 따라 어느 상위 연결을 사용할 것인지는 여러가지 선택이 있다. 이의 결정은 다른 메세지에 방해 받지 않는 연결들 중에서 pseudo-random하게 선택함으로써 해결하고 있다. 각 레벨에서의 pseudo-random한 선택은 각 채널의 부하를 균등하게 만들어 준다.

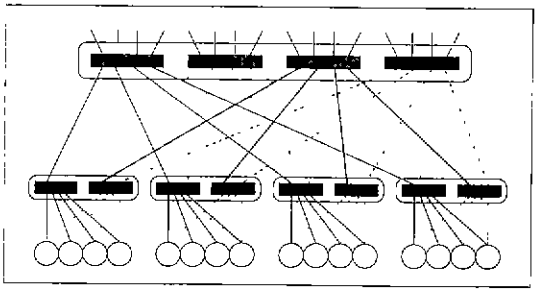

그림 5 CM5의 4-ary fat-tree의 구조

• Control Network

Control Network은 binary tree 구조를 갖고 있다. Control processor는 사용자 프로그램 중 스칼라 부분을 실행하고, processing node들은 병렬처리 부분을 실행한다. Data network의 가변 길이의 메세지와 달리, control network의 패킷들은 65비트의 고정된 길이를 갖는다.

Control network에서의 세가지 주요 기능은 broadcasting, combining, global operations이다. 이들 동작은 프로세서간 통신을 제공하고, 병렬처리를 효과적으로 지원하는 수단을 제공하고, general-purpose 응용을 위한 MIMD 실행을 지원한다.

• Diagnostic Network

Diagnostic network은 주소 지정을 쉽게 하기 위해 binary tree 구조를 갖고 있다. Root에는 한개 이상의 diagnostic processor들이 있다. Diagnostic interface는 JTAG(Joint Test Action Group)를 지원하는 CM-5의 모든 칩들과 네트워크들을 종합적으로 시험할 수 있게 설계되었다. 또한 diagnostic network은 자체로 시험 및 진단이 가능하고 시스템의 주요 부분들의 전력을 차단할 수 있다.

3.2 KSR-1[4]

KSR-1은 메인 프레임으로서의 기능뿐만이 아니라 슈퍼 컴퓨터로서의 연산 능력을 갖는 범용 병렬처리 컴퓨터를 목표로 Kendall Square Research사에 의해서 개발되었다. 최소 8개의 프로세서에서 최대 1088개의 프로세서를 2단계 계층 구조의 고속 링으로 연결하여 최대 43,520 MIPS와 43,520 MFLOPS 성능을 갖고 있으며, 입출력 전송량은 15,300 Mbyte/sec이다.

• 연결망 구조

KSR-1의 상호연결망은 2단계 계층 구조를 갖는 링으로 구성되며, 상위 단계로 올라 갈수록 전송량이 증가하는 fat-tree 형태의 토폴로지를 갖고 있다. 하단 링은 자체 개발한 프로세서와 32 Mbyte의 지역 메모리(로컬 캐쉬 : ALLCACHE architecture)와 링 인터페이스로 구성되는 프로세싱 노드가 32개까지 연결된다. 이와 같이 구성된 하단 링들이 다시 상단 링에 34개까지 연결되어 최대 1088개의 노드를 연결할 수 있다. KSR-1의 전체적인 상호연결망 구조는 그림 6과 같다. 각 노드는 링 인터페이스인 CI(Cell Interface)를 통하여 링에 연결된다. 링은 1 Gbyte/sec의 밴드폭을 갖는 단방향 슬롯 링으로 구성되며, 하나의 슬롯에는 16 바이트의 헤더 정보와 128 바이트의 데이타 정보로 구성되는 하나의 패킷을 전송할 수 있다.

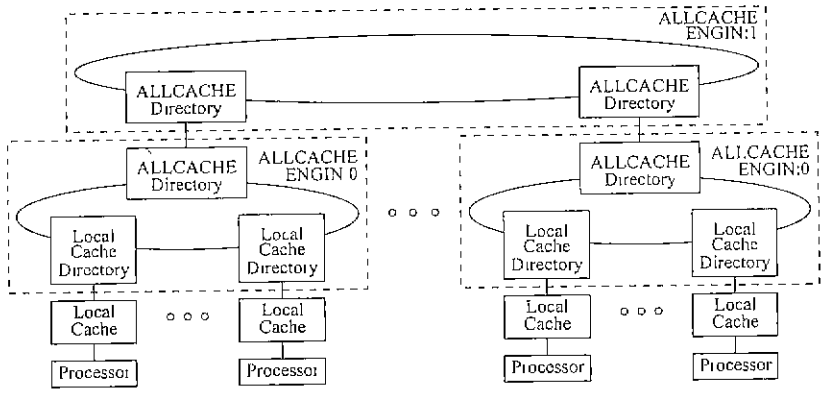

그림 6 KSR-1의 구조

• 메모리 구조

KSR-1은 cache-only-machine으로 분류되며, ALLCACHE 메모리 구조를 갖고 있다. ALLCACHE 메모리 구조는 기존의 페이징 기법을 사용하는 가상 메모리 가법과 캐쉬 메모리의 개념을 혼합한 구조로써, 프로세서에서 발생된 메모리 요구에 대하여 데이타뿐만 아니라, 실제 물리적 주소까지 노드에 전달되는 구조이다. 따라서 시스템상에서 동일한 물리 주소가 서로 다른 노드상에 복수개 존재할 수 있으며, 각 노드에 있는 메모리를 로컬 캐쉬라 명명하고 있다. 각 노드는 128 bytes 단위의 서브 페이지(cache block size)로 구성되는 32 Mbytes의 로컬 캐쉬를 갖고 있다. 데이타 일관성 유지를 위하여 각 링에 ALLCACHE ENGINE이라는 별도의 노드를 두고 각 로컬 캐쉬의 상태(Eexclusive, Copy, Non-Exclusive, Invalid)를 full-map 디렉토리 방식을 사용하여 중앙 집중식으로 제어한다. 그리고 상단 링에 존재하는 ALLCACHE ENGINE은 자신에게 연결된 모든 서브 링에 존재하는 ALLCACHE ENIGINE의 캐쉬 디렉토리를 복사해서 갖고 있다.

KSR-1의 ALLCACHE 메모리 시스템은 병렬처리 구조에서 지원하는 대부분의 동기화 방식을 제공한다. 각 서브 페이지별로 'atomic'이라는 별도의 상태를 정의하여 data locking, barrier, critical region 등의 동기화 프리미티브를 구현할 수 있게 했으며, 이를 위하여 get, get-and-wait, release라는 동기화 전용 명령어를 제공한다.

• 입출력 구조

각 노드는 KSR 자체에서 개발한 프로세서를 사용하여 구성되었으며, 각 프로세서는 입출력 주변장치와 인터페이스를 위하여 30 Mbytes/sec의 입출력 채널을 가지고 있다. 이 입출력 채널을 이용하여 하나의 링 상에 최대 15개의 입출력 채널을 연결할 수 있으며, 전체 시스템 상에는 510개의 입출력 채널(15,300 Mbytes/sec)을 연결할 수 있다. 각 입출력 채널은 사용자의 요구에 따라 전용 입출력 어뎁터나 범용 입출력 시스템을 통하여 MCD(Multiple Channel Disk), MCE(Multiple Channel Ethernet), MCF(Multiple Channel FDDI), VCC(VME Channel Controller), HiPPI 등을 연결할 수 있다.

• 소프트웨어

KSR-1 오퍼레이팅 시스템은 OSF/1을 기본으로 한 유닉스 운영체제로서 AT&T의 System V는 물론 BSD4.3, BSD4.4와 호환성을 가지며, ANSI C, FIPS 15-1, POSIX 1003.1 XPG3과 같은 표준 소프트웨어 환경을 제공한다. 또한 많이 쓰이고 있는 C, FORTRAN77, C++ 등의 컴파일러를 제공하며, 소프트웨어 개발을 지원할 수 있는 소오스 코드 디버거인 udb 디버거를 제공한다. 그리고 소프트웨어의 실행 환경으로는 사용자가 프로그램을 통하여

쓰래드 단위로 생성하고 제어할 수 있도록 PRESTO를 제공한다. 응용 소프트웨어 부분에서는 관계형 데이타베이스인 ORACLE을 지원하며, 특히 의사결정 응용에 빠르게 처리하기 위해서 ORACLE을 위한 KSR Query Decomposer를 제공한다.

3.3 Convex-Meta[6]

Convex-Meta 시리즈는 RISC와 벡터처리 기술의 장점을 결합하여 만든 시스템으로 HP사의 PA-RISC 워크스테이션 클러스터링 기술과 Convex사의 Convex 3000 슈퍼컴퓨터 기술을 묶은 metacomputing 시스템이다. metacomputing이란 하나의 네트워크로 연결된 이기종 시스템들을 소프트웨어를 통해 하나의 동일기종 시스템으로 사용하는 기술을 말한다. 이러한 기술은 사용자 및 시스템 관리자에게 하나의 시스템 이미지를 제공함으로써 다양한 작업에 대해 적합한 구조를 동적으로 제공할 수 있다. 그림 7은 Convex-Meta 시스템의 구조를 보여 주고 있다.

• 연결망 구조 및 메모리 구조

하드웨어 측면에서 볼 때 벡터처리 슈퍼컴퓨터와 클러스터링 구조를 단순히 결합한 형태이므로 각각 독립적인 구조를 갖고 있다. Convex C-Series CPU가 8개까지 4GB/sec 크로스바 스위치를 통해 메모리에 연결되어 있다. 또한 클러스터링 구현을 위한 워크스테이션의 연결은 Ethernet 혹은 FDDI를 통해 입출력 서브시스템에 연결되어 있다. 하나의 워크스테이션은 PARISC를 4개까지 탑재할 수 있고, 125 Kbytes 명령어 캐쉬와 256 Kbytes 데이타 캐쉬와 128 Mbytes 메모리와 840 Mbytes 디스크를 갖고 있으며 최대 8개까지 연결할 수 있다.

• 입출력 구조

Convex C-Series 슈퍼컴퓨터에서 사용하는 입출력 장치를 그대로 사용한다. 6 GB에서 tera bytes까지 지원할 수 있는 RAID 구조의 대용량 디스크(Disk Farm)와 로보틱 테이프 장치를 고성능 입출력 서브시스템에 연결하여 사용한다. 입출력 대역폭은 100 MB/sec까지 제공한다.

• 소프트웨어

소프트웨어 측면에서도 슈퍼컴퓨터와 워크스테이션 클러스터링의 결합된 특성을 제공한다. 기존의 워크스테이션에 사용되던 응용소프트웨어를 클러스터링 구조에서 그대로 사용 가

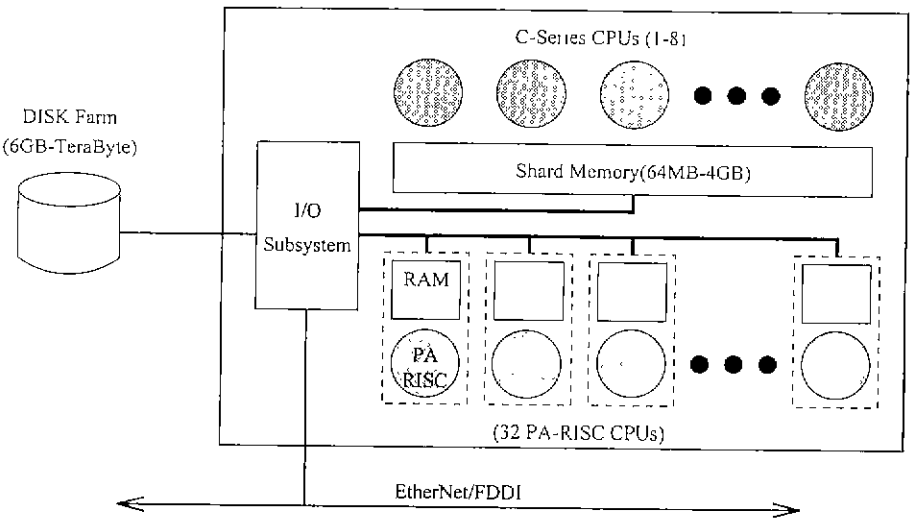

그림 7 Convex-Meta 시스템 구조

능하다. 통합된 환경에서 시스템의 유용성을 보장하는 중요 소프트웨어로는, 일의 분배를 효율적으로 관리하는 ConvexNQS+와, homogeneous 환경을 제공하는 ConvexPVM과, 기본적인 수학 및 계산에 대한 커널 최적화를 위한 ConvexMLIB가 있다.

3.4 NCR3600[7]

NCR 3600 시스템은 OLTP와 DSS와 같은 상업 자료처리 응용을 효율적으로 지원하기 위해 AT&T에서 개발한 병렬처리 시스템이다. NCR 3600 시스템은 기본적으로 클러스터 구조를 가지며, 클러스터는 APs(Application Processors), AMP(Access Module Processors), PEs(Parsing Engines), 그리고 Y-NET라 부르는 상호연결망으로 구성된다. 그림 8은 NCR 3600 시스템의 구조를 보여 주고 있다.

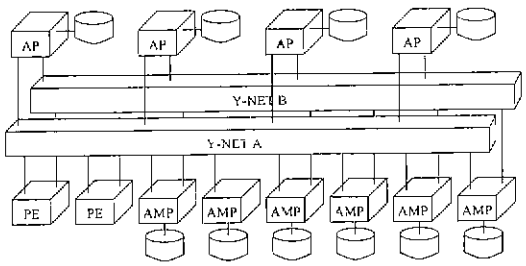

그림 8 NCR 3600 시스템 구조

• NCR 3600의 구조

각 노드(AP, AMP, PE)는 모두 Intel 486 프로세서를 기반으로 하고 있으며 메모리와 캐쉬와 그리고 입출력을 지원한다. AP 노드는 밀결합 다중처리 시스템으로 최소 2개에서 최대 8개의 486 프로세서를 가지며, AMP 노드와 PE 노드는 1개의 486 프로세서를 가진다. 각 AP 노드는 최대 320 MIPS의 처리 능력과 최대 512 MBytes의 공유메모리를 갖고 있으며, AP 노드의 프로세서는 256 KBytes의 external copy-back 캐쉬를 갖고 있다. 또한 AP 노드의 가장 큰 특징은 최대 400 MBytes의 데이타 전송률을 갖는 이중 시스템 버스이다. AP 노드내의 공유메모리는 이중 시스템 버스를 지원하기 위하여 이중 포트, 4-way 인터리브 메모리를 갖고 있다. 그리고 각 AP 노드는 디렉토리 기반 캐쉬 일관성 프로토콜을 사용한다.

• 연결망 구조

NCR 3600 시스템 내의 각 노드는 Y-NET이라는 버스 기반 트리 구조의 상호연결망으로 연결된다. Y-NET은 두개의 독립된 채널을 가지는데 각 채널은 약 6 MBytes의 실처리율(effective throughput)로 실제 약 12 MBytes의 실처리율을 갖는다. 또한, Y-NET은 256개의 노드를 지원할 수 있도록 하드웨어 중재기능과 브로드 캐스팅 기능과, 그리고 멀티 캐스팅 기능을 갖고 있다. AP 노드내에는 64-비트 데이타 폭을 갖는 이중 버스 구조를 사용한다. 이중 버스는 25 MHz에서 동작하여 최대 400 MBytes의 전송율을 갖고 있다.

• 메모리 구조

AP 노드내의 프로세서는 8 KBytes의 4-way set associative 명령어/데이타 캐쉬를 가지며, 외부에 256 KBytes의 copy-back 캐쉬를 갖고 있다. 각 AP 노드는 최대 512 MBytes의 공유메모리를 가지며, 각 AMP/PE 노드는 16 MBytes의 메모리를 갖고 있다.

• 입출력 구조

NCR 3600 시스템은 두 종류의 디스크 서브시스템을 지원한다. 하나는 SCSI 디스크 서브시스템으로 1.6 GBytes 5.25″ 디스크를 16개까지 지원하며 다른 하나는 디스크 어레이 서브시스템으로 약 20개의 디스크를 matrix 형태로 지원한다. 디스크 서브시스템은 구성에 따라 AP 노드나 AMP 노드에 연결된다.

• 소프트웨어

NCR 3600 시스템에서 제공하는 주요 소프트웨어는 아래와 같다.

1) UNIX/NFS Operating Environment
2) Oracle Database Environment
3) Teradata Database Environment
4) TOP END Environment

3.5 NCUBE-2[8]

NCUBE-2 시스템은 NCUBE사에서 cube 구조를 이용해 상용화시킨 병렬처리 시스템이다. NCUBE 계열의 시스템으로는 1985년에 발표한 NCUBE/ten(1024 노드) 시스템과 1989년에 제2세대 시스템으로 발표한 NCUBE-2 (8192 노드) 시스템이 있다. NCUBE/ten 시스템은 10-D hypercube 구조이고 NCUBE-2 시스템은 13-D hypercube 구조이다. 자체 개발한 NCUBE-2 시스템의 64-비트 프로세서는 FPU, MMU, routing 하드웨어, 그리고 14쌍의 입출력 DMA 포트를 가지고 있다.

• 연결망 구조

14쌍의 DMA 포트 중 13쌍은 13-D hypercube를 구성하는 각 노드를 연결하기 위해 사용되고, 나머지 1쌍은 입출력 서브시스템을 연결하기 위해 사용된다. NCUBE-2 시스템은 최소 32노드에서 최대 8192 노드까지 확장 가능하며, 메모리 용량은 최소 1 Mbytes/node 최대 64 MBytes/node 까지 확장 가능하다. 하나의 프로세서 보드는 64개의 프로세서 노드를 갖고, 하나의 I/O 보드는 16개의 프로세서로 구성된다. NCUBE/ten 시스템과 마찬가지로 하나의 마더보드는 16개의 프로세서 보드와 8개의 I/O 보드를 장착할 수 있으며, 8개의 마더보드를 서로 연결하여 13-D hypercube 시스템을 구성할 수 있다. 각 I/O 보드는 각 슬롯당 568 MBytes /sec 전송량을 갖고 있다. NCUBE-2 시스템의 최대 구성은 128 프로세서 보드와 64 I/O 보드로 8192 프로세서와 512 GBytes 메모리를 구성하여 최대 27 GFLOPS의 성능을 얻을 수 있다. 이때 I/O 전송폭은 36 GBytes/s가 되며 4096개의 디스크를 지원한다.

• 메모리 구조

NCUBE/ten 시스템의 각 프로세서 노드는 프로세서마다 128 KBytes의 메모리를 갖고 있고, 각 I/O 보드는 4 MBytes의 메모리를 갖고 있다. 프로세서의 데이타폭은 32-비트이지만 외부 메모리 참조시 한 클럭 동안 단지 16-비트 데이타만을 참조할 수 있어 상대적으로 메모리 참조 성능은 낮은 편이다. NCUBE-2 시스템의 각 프로세서 노드는 64 MBytes의 메모리를 갖고 있고, 각 I/O 보드는 2 MBytes의 버퍼 메모리를 프로세서마다 갖고 있다.

• 소프트웨어

NCUBE-2 시스템에서는 프로세서 보드에서 UNIX나 VMS OS를 수행하는데 각 노드는 각기 다른 OS를 수행할 수 있다. 프로그래밍 언어로는 Fortran 77과 C를 제공한다.

3.6 Paragon[9]

인텔에서 개발한 슈퍼컴퓨터 시리즈 중에서 가장 최신의 시스템으로 성능과 용량의 확장성이 뛰어나고 UNIX 서비스를 기반으로 강력한 병렬 프로그래밍 환경을 갖고 있다. Paragon_XP/S가 정확한 명칭으로 현재 최대 300 MFLOPS의 성능을 갖고 있으며, 추후 개발될 더욱 강력한 마이크로프로세서와 패키징 기술의 발전으로 1990년대 중반까지 TFLOPS 시스템으로 발전될 것이다. 개략적으로 살펴본 Paragon 시스템의 모습은 다음과 같다. 각 노드에 메모리가 분산된 멀티컴퓨터 MIMD 구조로 노드는 i860 마이크로프로세서와 16-128 MBytes의 메모리를 내장하고 있다. 노드의 연결은 2D 매쉬 구조이며 노드간의 전송 속도는 최대 200 MBytes 이다. 시스템 성능은 최대 300 MFLOPS의 64-비트 부동소숫점 연산, 최대 160 GIPS의 정수 연산 능력을 갖고 있다. 사용자 인터페이스를 위하여 완벽한 UNIX 서비스와 프로그래밍 환경 등을 제공한다.

• 노드 구조

노드의 구조는 입출력, 계산, 그리고 서비스의 쓰임새에 따라 약간의 차이가 있다. 단위 노드의 기본 구조는 메세지 전송과 계산 용도로 각각 사용되는 두개의 i860과 16-64 MBytes의 메모리를 포함한다. 메세지 전송용 i860은 계산용 i860에서 요구하는 메세지를 만들고 이를 송신/수신하는 기능을 수행한다. 계산 및 서비스 노드는 기본 구조와 달리 16-128 MBytes의 메모리가 설치되어 있다. 입출력 노드에는 한개의 I/O 인터페이스를 설치할 수 있는데, 사용자의 데이타 요구량에 따라 신축적으로 대응할 수 있

도록 다양한 I/O 인터페이스를 갖고 있다. 즉, I/O 인터페이스는 사용자 요구에 맞춰 SCSI-2, HIPPI, 혹은 VME 등을 선택할 수 있다.

• 연결망 구조

노드는 NIC(Node Interface Controller)라 불리는 인터페이스를 통하여 2D 매쉬 형태로 배열된 PMRC(Paragon Mesh Routing Chip)에 연결되며, NIC와 PMRC의 사이는 양방향으로 전송하며 전송 속도는 200 MBytes를 상회한다. PMRC는 동시에 4방향의 노드로 전송되는 메세지를 중재할 수 있다. PMRC의 스위칭 속도는 40 nano sec.이다.

• 소프트웨어

Paragon의 운영체제는 MACH의 커널과 OSF/1 AD를 바탕으로 한 분산 운영체제이다. 따라서 사용자는 멀티컴퓨터의 구조와 노드 연결 방법을 모르고도 UNIX의 모든 서비스를 사용할 수 있다. 소스 프로그램 호환성을 위하여 OSF/1 API를 제공한다. 그리고 시스템 구조에 최적화된 프로그래밍을 돕기 위하여 병렬 컴파일러(Data-Parallel FORTRAN), 객체지향 언어 컴파일러(ADA, C++, ANSI C), 그리고 병렬 프로그램 개발을 위한 CASE 도구 등을 제공한다.

3.7 병렬처리 시스템 비교

표 1은 앞에서 기술한 병렬처리 시스템들과 Cray사의 T3D 시스템과 BBN사의 TC-9000 시스템을 비교한 것이다. 비교시 각 회사에서 제공하는 자료를 사용하였기 때문에 일부는 과장될 수도 있음을 밝혀둔다. 특히 각 시스템의 최고 성능과 최대 규격은 실제적인 것보다 이상적인 수치가 많이 있다. 다시 말하면 최대 규모의 시스템이 팔리는 것이 아니라 비교적 적은 규모의 시스템이 팔리고 있기 때문이다.

비교 결과 3가지 주요 특징을 발견할 수 있었

표 1 상용 병렬처리 시스템 비교

시스템	CM-5	KSR-1	SPP-1000	NCR 3600
발표시기	1992	1991	1994	1992
제작사	TMC	KSR	CONVEX	AT&T NCR
프로세서 수	16,000	1088	128	256
프로세서 종류	Super SPARC	Custom	PA-7100	Intel 486
최대 메모리	512 GB	–	32 GB	–
캐쉬 구성	–	ALLCACHE	2 MB/proc.	256 KB, W-B
입출력 용량	–	–	4 GB/sec	–
프로그래밍 모델	메시지 전달	공유메모리 (COMA)	분산 공유메모리 (DSM)	메세지 전달, 공유메모리
상호연결망	Trec	계층 링	SCI 링	Y-NET 버스
클러스터 수	–	32개	16개	–
노드 수	–	32개/링	8개	8개/AP, 1개/AMP
성능	128 MFLOPS/node	43.5 GFLOPS, 43.5 GIPS	25 GFLOPS	–
주요 응용분야	과학 계산	과학 계산, OLTP, DB 응용	과학 계산, DB 응용	OLTP, DSS

표 1 상용 병렬처리 시스템 비교(계속)

시스템	NCUBE-2	PARAGON	T3D	TC-9000
발표 시기	1989	1991	1993	1989
제작사	NCUBE	INTEL	CRAY Res.	BBN
프로세서 수	8192	4096	2048	512
프로세서 종류	Custom, 64-bit	Intel 860	Alpha 21064	MC88100
최대 메모리	512 GB	128 GB	–	8 GB
캐쉬 구성	–	–	–	48 KB/proc.
입출력 용량	36 GB/sec	6.4 GB/sec	12.8 GB/sec	2.5 GB/sec
프로그래밍 모델	메세지 전달	메세지 전달	메세지 전달, 공유메모리	분산 공유메모리 (DSM)
상호연결망	13D Hypercube	2D Mesh	3D Toroidal	Butterfly
클러스터 수	8개	–	256개	64개
노드 수	16개	–	–	8개
성능	61 GIPS, 27 GFLOPS	160 GIPS, 300 GFLOPS	–	10 GFLOPS
주요 응용분야	과학 계산, DB 응용	과학 계산,	과학 계산, 상업 응용	과학 계산, 실시간 응용

다. 첫째, 응용 분야가 아직도 과학 계산 분야에 많이 치중되고 있다. 과학 계산 분야는 병렬화 시킬 부분이 많이 있고 시스템 구조도 비교적 간단하기 때문이다. 반면에 OLTP나 DSS 응용에서 상용 자료를 병렬로 처리하기에는 병렬 알고리즘이나 병렬처리 구조에 대한 연구가 많이 진행되어야 할 것으로 생각된다. 둘째, 사용하고 있는 마이크로프로세서가 모두 다르다는 점이다. 마이크로프로세서는 그 시스템의 기본 구조와 특성을 나타내는 주요한 부분으로 시스템 설계시 충분히 고려돼야 한다. 이렇게 각 시스템마다 다른 프로세서를 사용하고 있다는 것은 병렬처리 전용 마이크로프로세서가 아직도 없다는 것을 간접적으로 말해주고 있다. 셋째, 상호연결망이 서로 다르다는 것이다. 상호연결망이란 그 시스템의 병렬처리 구조를 결정하는 가장 중요한 요소인데, 이것이 서로 다르다는 것은 아직도 범용화된 병렬처리 구조가 없다는 것이다.

4. 결 론

본 논문에서는 지금까지 개발된 상용 병렬처리 시스템에 대한 주요 특성을 비교 분석하였다. 표 1에서 보는 바와 같이 각 시스템마다 추구하는 기술의 방향이 상당한 차이를 보여 주고 있다. 이는 아직도 병렬처리 시스템에 대한 확실한 기술이 정착되지 못하고 있음을 보여줄 뿐 아니라, 특정 응용 분야에 특화된 병렬처리 구조를 각각 갖고 있다는 것을 의미한다. 다시 말하면 아직도 범용 병렬처리 시스템으로 발전하기에는 극복해야 할 문제가 남아 있다. 이 문제는 여러 프로그래밍 모델에서 병렬성을 최대한 검출하고, 이를 일반화시켜 많은 시스템에 적용할 수 있는 범용의 병렬처리 구조, 병렬 알고리즘, 병렬 컴파일러, 병렬 프로그래밍 환경 등을 사용자에게 제공해야만 해결할 수 있다.

본 논문에서는 상용 병렬처리 시스템에 국한하여 비교 분석하였지만 외국의 학교나 연구소에서도 다양한 형태의 병렬처리 시스템을 시험적으로 연구하고 있다. 주요 시험 시스템으로는 스탠포드 대학의 DASH 시스템[10], 위스콘신 대학의 Multicube 시스템[11] 등이 있다. 이들 시험 시스템들은 대규모 병렬처리시 발생하는 프로세서간의 통신 지연시간(latency)을 줄이기 위해 분산 공유메모리(DSM) 모델을 많이

사용하는 추세이다. 그리고 상호연결망 구조는 아직 혼재상태에 있지만, MIN이나 Hypercube와 같은 복잡한 구조보다는 비교적 간단한 계층 링, 계층 버스, 크로스바 등을 사용하여 전송폭을 높히고 지연시간을 줄이는 방향으로 연구하고 있는 추세이다.

참 고 문 헌

[1] M. J. Flynn, "Very High-Speed Computing Systems," Proceedings of the IEEE, Vol. 54, No. 12, pp. 1901-1909, Dec. 1966.

[2] E. Johnson, "Comleting an MIMD Multiprocessor Taxonomy," ACM SIGARCH Computer Architecture News, Vol. 16, No. 3, pp. 44-47, June 1988.

[3] K. Hwang, Advanced Computer Architecture : Parallelism, scalability, Programmability, McGraw-Hill Inc., QA76.9.A73H87, pp. 457-623, 1993.

[4] D. Windheiser and et. al., "KSRI Multiprocessor : Analysis of Latency Hiding Techniques in a Sparse Solver," 7th International Parallel Processing Symposium. pp. 454-461, 1993.

[5] R. Saavedra and et. al., "Micro Benchmark Analysis of the KSR1," Processing Supercomputing '93, pp. 202-213, 1993.

[6] 삼성휴렛팩커드, "HP 클러스터 컴퓨팅 세미나," 세미나 자료, 삼성휴렛팩커드, February 1993.

[7] M. Nichols, "The NCR 3600 : A Scalable Parallel Processing System for Commercial Enterprise Applications," The NCR Journal, pp. 4-9, April 1993.

[8] A. Trew and G. Wilson, Past, Present, Parallel, Springer-Verlag, QA76.58.P38, 1991.

[9] Intel Co., Paragon_XP/S Product Overview, Intel Co., 1991.

[10] D. Lenoski and et. al., "The Stanford Dash Multiprocessor," IEEE Computer, pp. 63-79, 1992.

[11] J. Goodman and et. al., "The Wisconsin Multicube : A New Large Scaled Cache-Coherent Multiprocessor," ISCA, pp. 422-431, 1988.