클러스터 컴퓨팅 기술동향

Cluster Computing Technology

김진미(J.M. Kim) 프로그래밍환경연구팀 선임연구원
온기원(G.W. On) 프로그래밍환경연구팀 선임연구원
김학영(H.Y. Kim) 프로그래밍환경연구팀 선임연구원
지동해(D.H. Chi) 프로그래밍환경연구팀 책임연구원, 팀장

고성능 프로세서와 초고속 네트워크 등의 하드웨어 기술이 발전하게 됨에 따라 최근 슈퍼컴퓨터와 같은 고가의 대형 컴퓨터를 사용하는 대신에 여러 개의 프로세싱 노드들을 클러스터링 기술을 사용하여 고속의 네트워크로 묶는 클러스터 시스템을 많이 활용하고 있다. 클러스터 시스템의 응용 분야는 병렬 처리를 비롯하여 멀티미디어나 대용량 데이터 베이스와 같은 입출력 중심적인 분야까지 넓어지게 되었다. 본 고에서는 클러스터 시스템을 구축하기 위하여 필요한 전반적인 클러스터 컴퓨팅 기술에 대하여 설명하였다.

I. 머리말

최근 슈퍼컴퓨터와 같은 고가의 대형 컴퓨터를 사용하는 대신에 여러 개의 프로세싱 노드들을 클러스터링 기술을 사용하여 고속의 네트워크로 묶는 클러스터 시스템을 많이 활용하고 있다. 클러스터 시스템은 클러스터 컴퓨팅 기술을 이용함으로써 가용성 및 확장성이 우수하고 성능이 뛰어나다[1].

일반적으로 클러스터에서 사용되는 기술로는 고속의 통신 기술, 파일시스템 기술 및 운영체제에서의 프로세스 관리 기술 등을 들 수 있다. 통신 기술의 경우에는 서로 다른 컴퓨터에서 병렬적으로 수행하는 작업들간에 프로세스 제어와 데이터 전송 등을 제공해주며, 파일 시스템의 경우 효율적인 데이터 관리를 위하여 대규모 데이터를 여러 노드에 분산 저장하고 일관성 있게 접근할 수 있는 메커니즘을 제공해줄 수 있다. 클러스터링을 위한 프로세스 기술의 경우에는 병렬처리를 위해서 여러 노드에 나누어져서 실행되는 프로세스를 관리하기 위하여 작업 스케줄링, 자원 할당, 제어 등의 기능을 제공한다. 이러한 클러스터링 기술들을 사용하여 단일 시스템 이미지를 제공해줄 수 있으며 클러스터를 관리해줄 수 있는 소프트웨어를 구축할 수 있다.

본 고에서는 클러스터 시스템을 구축하기 위하여 필요한 전반적인 클러스터 컴퓨팅 기술에 대하여 설명한다. II장에서는 클러스터 시스템의 구조와 구성요소 및 클러스터링 기술에 대해서 좀 더 세부적으로 알아보고, III장에서는 이러한 기술을 이용한 클러스터 환경 기반을 위한 소프트웨어와

클러스터를 관리해주는 소프트웨어에 대해서 기술하였다. 그리고 끝으로 IV장에서 결론을 맺는다.

II. 클러스터링 기술

수년간 SMP(Symmetric Multiprocessing) 등의 고확장성 시스템 구조가 주류를 이루었으나 메모리 버스 대역폭 등의 제한적인 설계 구조로 SMP에서는 주로 32에서 64프로세서만을 효과적으로 지원할 수 있다. 그러나 이러한 단일 SMP로는 효율성을 낼 수 없는 대용량 어플리케이션들은 슈퍼컴퓨터 이상의 고가의 대형 컴퓨터가 필요하다. 이에 가격 대 성능비가 우수하고 고성능, 고가용성, 고확장성을 지원할 수 있는 클러스터 시스템이 대체할 수 있게 되었다. 클러스터 시스템은 고성능 프로세서와 초고속 네트워크를 이용한 범용 컴퓨터들을 클러스터링하는 것으로 다음과 같은 구조와 구성요소 및 기술들을 가지게 된다[2].

1. 클러스터 시스템의 구조

클러스터 시스템의 구조는 공유 디스크(shared disk)와 공유 디스크가 없는(shared nothing) 소프트웨어 모델이 일반적으로 사용된다 공유 디스크 모델에서는 클러스터내의 서버에서 수행되는 모든 소프트웨어가 클러스터 시스템에 연결된 디스크 등의 모든 자원을 사용할 수 있으나 주 메모리는 공유되지 않는다. 두 서버가 같은 데이터를 읽을 때, 데이터는 디스크로부터 두 번 읽히거나 한 시스템에서 다른 시스템으로 복사된다. SMP 시스템에서와 같이 어플리케이션은 공유 데이터 접근을 동기화와 순차화시켜야 한다. 동기화를 위하여 분산 잠금 관리자(Distributed Lock Manager: DLM)가 사용되며 DLM은 어플리케이션에게 클러스터내의 자원 참조 상태를 알려준다. 만일 두 대 이상의 서버가 동시에 단일 자원을 참조하려 할 때, DLM은 충돌을 감지하여 이를 해결한다. DLM의 조정은 여러 서버가 직렬 액세스하게 되어 부가적으로 통신 메시지를 발생시켜 성능을 저하시킬 우려가 있다. 이 문제를 해결하기 위하여 무공유 소프트웨어 모델이 사용된다.

무공유 소프트웨어 모델은 각 노드가 각각 자체 메모리와 디스크를 가지는 형태이며 클러스터내의 서버가 개별적으로 클러스터 자원의 일부를 소유한다. 서버 고장시 소유하고 있던 자원을 물려받을 다른 서버를 가변적으로 고려하지만, 평상시는 특정 자원을 한 시스템만이 소유하고 사용한다. 또한 클라이언트의 요청이 자원을 소유하고 있는 서버로 자동적으로 라우트 된다. 한 클라이언트가 여러 서버가 소유하고 있는 복수 개의 자원을 사용하려 할 때 한 서버가 호스트 역할을 하게 되며 호스트 서버는 클라이언트 요구를 분석하여 하부 요구를 적절한 서버에게 보낸다. 각각의 서버는 하부 요구를 처리하고 필요시 호스트 서버에 응답하며, 호스트 서버는 최종 결과를 모아 클라이언트에게 보낸다. 호스트에 보낸 요구는 복수 데이터 레코드 조회시에 복수 디스크 읽기 등 많은 시스템 동작이 필요로 하는 높은 수준의 기능으로 최종의 원하는 데이터를 찾을 때까지 트래픽을 발생시키지 않는다. 클러스터 서버들에 분산되어 있는 데이터베이스와 같은 어플리케이션을 잘 활용하면 전반적인 시스템 성능이 단일

서버를 사용하였을 때보다 선형적으로 증가할 수 있다. 이러한 클러스터 시스템이 가져야 할 구성 요소는 다음과 같다.

- 노드(node): 클러스터를 구성하는 기본 서버로서 SMP 또는 단일프로세서 시스템이 사용된다. 프로세서, 메모리 이외에 전용 디스크와 OS 이미지를 가지고 단일 시스템(stand-alone)으로도 동작할 수 있다. 인접 서버와 물리적으로 디스크를 공유할 수 있다.
- 연결망(interconnect): 노드와 노드를 연결하는 공유 매체로 노드간의 통신을 위해 존재한다. 노드간의 상태 정보나 필요한 데이터의 교환이 주로 이루어지며 클러스터의 성능 확장성을 제공하기 위해서는 고속의 연결망이 필요하다. Fast Ethernet, FDDI(Fiber Distributed Data Interface) 등과 같은 표준 네트워크 제품을 사용하거나 ServerNet과 같은 고성능의 네트워크 제품을 사용할 수 있다.
- 디스크 서브시스템(DISK subsystem): 대부분의 클러스터 응용분야에서는 대용량의 데이터를 사용한다. 이러한 데이터를 저장하는 디스크는 고성능 및 고신뢰성을 제공할 수 있어야 한다. 데이터 저장 디스크는 클러스터 구축 방식에 따라 각 노드가 전용(shared nothing)으로 또는 노드간에 공유(shared disk)하여 사용된다. 대부분 RAID(Redundant Array of Inexpensive Disk) 제품을 사용하며 노드간의 인터페이스는 SCSI(Small Computer System Interface)를 많이 이용한다. 고성능의 Fibre Channel 등도 많이 사용되고 있다.
- 클러스터용 운영체제: 클러스터내 각 노드에 별도의 이미지로 존재한다. 고확장성, 고가용성의 클러스터를 구축하기 위하여 내장된 클러스터링 소프트웨어와 함께 병렬성, 고속 I/O, 고속 통신을 지원할 수 있어야 한다. 내장된 클러스터링 소프트웨어에는 노드간 통신기능, 고장 복구(fail-over) 기능, 노드그룹 관리 기능 등이 있다. 이러한 기능들은 API(Application Programming Interface)로 제공되며 이를 이용하여 복잡한 클러스터 기능을 구현한 cluster-aware 소프트웨어를 개발할 수 있다.
- Cluster-Aware 소프트웨어: 클러스터 기반의 클라이언트 서버 환경에서 응용 프로그램 개발을 위해 필요한 미들웨어이다. 클러스터에서 제공되는 확장성, 가용성을 이용하여 응용 프로그램에 병렬성, 고장 복구 기능을 구현할 수 있다. 클러스터의 응용분야에 따라 다양한 cluster-aware 소프트웨어가 있을 수 있으며 대표적인 패키지로 Oracle Fail Safe, Oracle Parallel Server, Microsoft SQL Server, Microsoft Transaction Server, Microsoft Message Queue Server 등이 있다.
- 어플리케이션: 클러스터는 고확장성, 고가용성의 장점을 활용하여 데이터웨어하우스, 온라인 트랜잭션 처리 및 인터넷/인트라넷 정보 서버 등의 주요 업무에 사용될 수 있다.

2. 클러스터 활용 기술

클러스터에서 사용되는 기술은 통신 기술, 파일 시스템 기술 및 운영체제 기술 등이 있으며 여

기에서는 통신 기술 및 파일 시스템 기술에 대한 실례를 들었다.

가. 통신 기술

클러스터에 관한 연구의 대표적인 UC Berkeley의 NOW(Network of Workstation) 프로젝트에서는 원격 시스템의 메모리에서 데이터를 가져오는 것이 네트워크를 사용하지 않고 디스크에서 데이터를 가져오는 것보다 더 빠르다는 것을 보여주므로 최근의 고성능 네트워크의 역할이 그만큼 커졌음을 보여준다[3]. 이러한 ATM(Asynchronous Transfer Mode)과 Myrinet과 같은 고속의 네트워크의 발달로 클러스터 컴퓨터에서도 슈퍼 컴퓨터에 견줄 수 있을 만큼 성능 향상을 얻을 수 있게 되었다. 그러나 패킷 프로세싱의 오버헤드로 성능 저하를 가져오므로 이러한 네트워크 오버헤드를 극복하기 위한 고속 통신 기술에 대한 연구가 활발하게 진행되고 있다. 일리노이 대학의 Fast Messages, 코넬 대학의 U-Net, Active Messages 등이 고속 전송 시스템 소프트웨어이다.

- Illinois Fast Messages(FM)[4]

  FM은 워크스테이션을 네트워크로 연결한 NOW와 MPP 등에서 사용할 수 있도록 고성능 메시지 계층을 구현한 것이다. 네트워크의 하드웨어 성능을 우선적으로 고려하여 설계된 저수준의 메시지 계층으로 FM 2.1에서는 초당 41메가 바이트를 전송할 수 있다. 프로세서들간 네트워크를 공유하지 않는다.

- U-Net[5]

  코넬에서 개발되었으며 병렬 분산 컴퓨팅을 위해 사용자 수준의 네트워크 인터페이스를 정하고, 이를 가상화시켜 다수의 프로세스들간에 인터페이스를 공유할 수 있게 한다. U-Net TCP 는 기존의 통신 프로토콜인 표준 TCP 스택의 기능을 모두 지원하며 추가로 성능을 향상시키기 위하여 수정하여 구현한 것이다. 이는 네트워크의 전체 대역폭(bandwidth) 전송을 지원하지만 여전히 100 마이크로 세컨드 이상의 패킷 처리 오버헤드가 따른다.

- Active Message(AM)[6]

  굵은 단위(fine-grain)의 통신을 하기 위한 강력한 프리미티브로 고수준 통신 프로토콜 함수를 제공하는 라이브러리와 병렬 언어 컴파일러로부터 통신 코드 생성시에 기본 계층으로 사용되는 메시지 수들로 구성된다. 어플리케이션 프로그래머가 직접 이용하지는 않으며 NOW 프로젝트에서는 AM을 Fast Communication 계층(sockets, RPC(Remote Procedure Call)와 MPI(Message Passing Interface)), xFS 병렬 파일 시스템, Split-C와 Id 컴파일러, Scalapack 라이브러리 등에서 현재 이용하고 있다.

- Fast Socket(FS)[7]

  FS는 UC 버클리에서 개발한 통신 소프트웨어로 AM을 기반으로 하고 있으며 LAN용 고속 통신에 적합하다. 프로세싱 오버헤드 문제를 해결하기 위해서는 API를 변경하거나 네트워크 프로토콜을 변경하는 방법, 프로토콜의 구현 자체를 변경하거나 혹은 이들을 조합하여 해결하는 방법 등이 있을 수 있는데, FM은 이중 새로운 통신 프로토콜로 지역 네트워크를 위하여 낮은 오버헤드를 갖는 데이터의 송수신 패스를 가질 수 있도록 구현하였다. API는 경량의 프로토콜

을 사용하며 패킷의 도착지를 사용자 버퍼로 직접 전송하는 단순한 버퍼 관리 전략으로 체계적인 프로토콜 계층을 사용하지 않음으로써 성능을 향상시킨다.

나. 파일 시스템 기술

클러스터 파일 시스템은 이를 구성하는 여러 개의 프로세싱 노드들이 고속의 네트워크에 연결되어 있다는 점에서는 분산 파일 시스템과 유사하며, 동시에 여러 클라이언트에서의 요구를 처리한다는 점에서는 병렬 파일 시스템과 유사한 부분이 있다. 그러나 병렬 파일 시스템은 과학 계산을 위하여 설계되었고, 분산 파일 시스템은 파일 공유를 위하여 최적화되어 클러스터 파일 시스템의 대부분은 파일을 여러 서버에 분산하는 스트라이핑 기법을 적용해 병렬적인 입출력 효과를 얻음으로써 처리량을 개선하고자 하고 있다[8].

UC 버클리의 NOW 프로젝트에서 수행한 분산 파일 시스템인 xFS(Serverless Network File Service)에서는 모든 자료의 저장과 캐시의 관리를 하나의 중앙집중화된 파일 서버가 담당하는 서버 시스템의 한계를 극복하고자 하였다. 데이터 분산은 LFS(Log structure File System)를 사용한 소프트웨어 RAID를 구현하여 하도록 하였으며, 서버와 클라이언트의 구별을 없애 어느 노드에서도 데이터를 저장하고 캐싱하며 관리할 수 있게 한다. 파일의 메타 데이터를 분산 저장함으로써 서버에서 생기는 병목 현상을 방지한다.

일종의 분산 파일 시스템인 Swift는 기존의 RAID 시스템을 소프트웨어적으로 구현한 것으로 여러 파일 서버들에 파일들을 스트라이핑 시키므로 입출력 성능을 향상시키며 패리티 정보를 추가적으로 저장하여 신뢰성이 높은 파일 서비스를 제공하게 된다.

클러스터 시스템을 위한 파일 시스템은 효율적인 데이터 관리를 위하여 대규모 데이터를 여러 노드에 분산 저장하고 일관성 있게 접근할 수 있는 메커니즘을 제공하여야 한다.

III. 클러스터 소프트웨어

일반적인 클러스터 소프트웨어는 단일 시스템 이미지를 갖도록 처리해주는 가상 공유 메모리 형태의 소프트웨어와 클러스터를 관리해주는 소프트웨어가 있다. 본 장에서는 주로 가상 공유 메모리 형태의 클러스터 환경 기반을 위한 소프트웨어와 클러스터를 관리해주는 소프트웨어에 대해서 기술하였다.

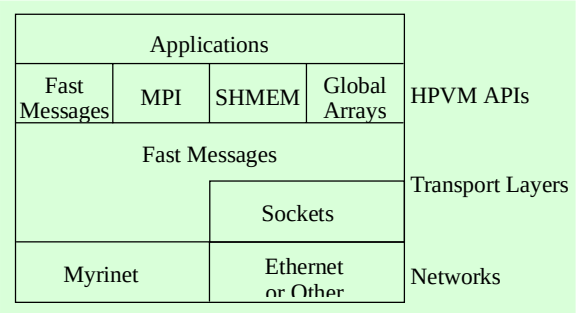
1. 클러스터 컴퓨팅 환경 기반을 위한 소프트웨어

가. HPVM[9]

최근 컴퓨터와 네트워크 하드웨어의 성능이 높아짐으로 그에 따른 소프트웨어의 지원이 필요하게 되었다. 고수준 API를 가지고 고성능 통신 프로그램을 하게 되고, 스케줄링과 자원 관리가 필요하며, 이질적인 머신을 관리하기 위한 소프트웨어가 필요하다. HPVM(High Performance Virtual Machines)은 연결된 컴퓨터에서 분산된 자원을 사용하여 고성능 계산을 목적으로 일리노이 대학에서 수행하고 있는 프로젝트이다. HPVM 어플리

케이션들은 전형적인 슈퍼컴퓨팅 어플리케이션들과 새로운 고성능 분산 어플리케이션들을 지원한다. HPVM은 Windows NT나 리눅스 상에서의 펜티엄 프로 컴퓨터들을 미리넷 네트워크로 연결한 클러스터링을 지원하게 된다.

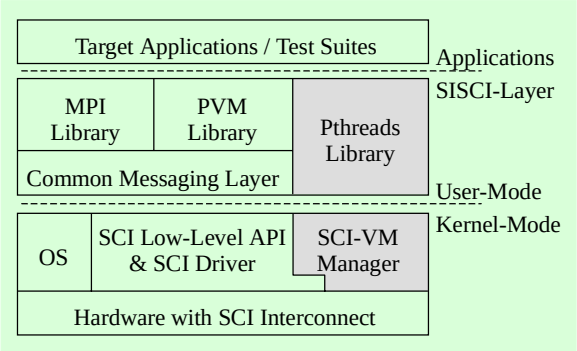
HPVM의 성능은 IBM SP2 와 Cray T3D와 같은 MPP(Massively Parallel Processor) 시스템들과 비교할 만하다. Fast Messages, MPI, SHMEM, Global Arrays와 같은 인터페이스를 지원하며 직접적인 미리넷 통신 및 소켓을 통한 10/100Mbps Ethernet 등 고성능 연결망을 지원한다. 프로그래밍 인터페이스는 (그림 1)과 같으며 바이너리를 무료로 사용해볼 수 있도록 배포하고 있다.



(그림 1) HPVM Programming Interface

나. SMiLE[10]

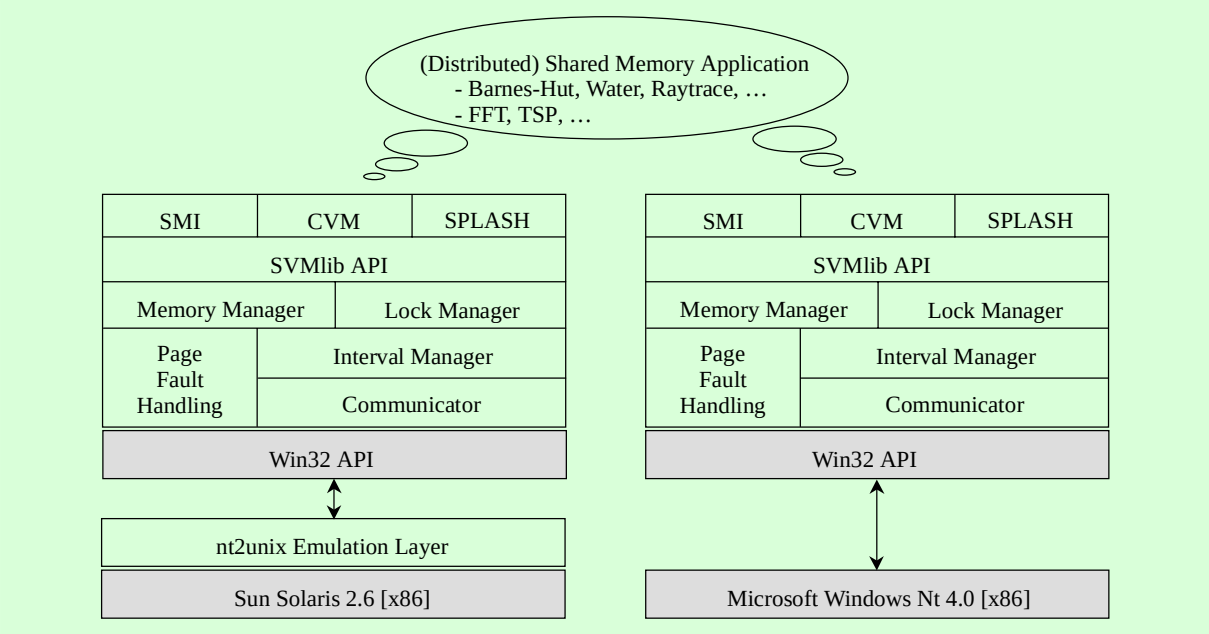
SMiLE(Shared Memory in a LAN-like Environment)은 SCI(Scalable Coherent Interface)기반의 클러스터링 병렬 시스템을 지원하는 소프트웨어 구조인 SISCI(Standard Software Infrastructure for the SCI-based Parallel Systems)을 위한 클러스터링 소프트웨어이다. SCI는 클러스터에서 원격 메모리를 투명하게 접근할 수 있도록 하드웨어적인 공유 메모리를 지원한다. 노드간의 통신에 있어서도 메시지를 주고받지 않고 원하는 노드의 원하는 메모리 주소에 직접 자료를 읽고 쓸 수 있는 공유 메모리 방식을 지원한다. 그러나 전역 공유 메모리처럼 사용할 수는 없으며, 그러기 위해서는 전형적인 소프트웨어 DSM(Distributed Shared Memory) 메커니즘이 필요하다. SMiLE은 LAN 환경에서 메모리를 공유할 수 있게 하는 소프트웨어로 SCI의 가상 메모리 환경을 지원한다. DSM 메커니즘을 따르지만 , 소프트웨어 DSM 시스템과는 달리 시스템내에서 데이터가 이전되거나 중복되지 않는다. SISCI의 구조는 (그림 2)와 같다. Windows NT 용으로 개발되었으며 PVM(Parallel Virtual Machine) 및 Pthreads 라이브러리 개발도 포함되어 있다.



(그림 2) SISCI 구조

다. SVMlib[11]

RWTH Aachen에서 개발한 Windows NT 상에서의 사용자 수준에서 사용하는 공유 가상 메모리 서브시스템으로 일반적인 DSM이다. nt2unix라는 자체 Win32 에뮬레이션 계층으로 컴파일하면



(그림 3) SVMlib Interface

UNIX에서도 사용 가능하다. 분산 동기화 알고리즘을 사용하여 병렬 어플리케이션에서의 문제를 해결하였다. TCP/IP와 같은 프로토콜을 지원하며 돌핀에서 구현한 SCI의 고성능 메시지 통신도 지원할 계획이다. SVMlib(Shared Virtual Memory Library)는 다른 공유 메모리 시스템과의 호환성을 위하여 4가지의 추가적인 API를 지원한다. SVMlib는 바이너리를 무료로 사용해볼 수 있도록 하고 있으며 시스템 인터페이스 계층은 (그림 3)과 같다.

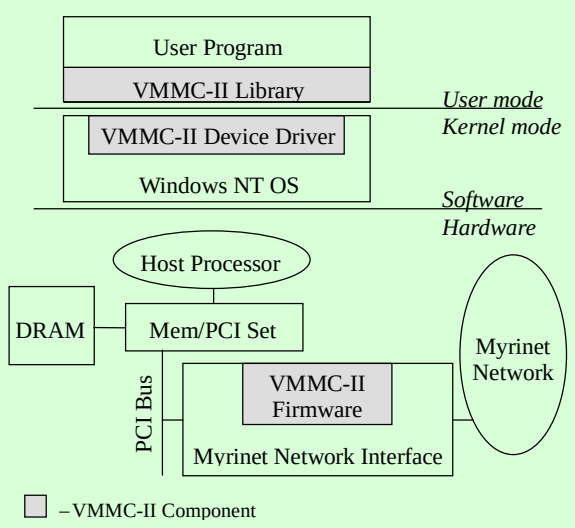
- SMI(Shared Memory Interface): RWTH Aachen에서 개발한 non-CC NUMA(Cache Coherent Non-Uniform Memory Access) 클러스터용
- CVM(The Coherent Virtual Machine): 메릴랜드 대학에서 개발한 멀티쓰레드 소프트웨어 DSM 시스템
- SPLASH(Stanford Parallel Application for Shared Memory): 스탠포드 대학에서 개발
- JIAJIA: Computing Technology, CAS에서 개발한 소프트웨어 DSM 시스템

라. VMMC[12]

VMMC(Virtual-Memory-Mapped Communication) 시스템은 프린스톤 대학의 SHRIMP (Scalable High-Performance Really Inexpensive Multi-Processor) 과제에서 가상 메모리 맵 통신을 지원하는 소프트웨어이다. 리눅스용과 Windows NT 용으로 개발되었으며 Windows NT 용인 VMMC-II/NT는 미리넷 네트워크 인터페이스를 지원하며 바이너리로 사용해볼 수 있다. VMMC

모델은 각각의 가상 메모리 버퍼간에 직접 데이터를 전송할 수 있다. 즉, 사용자 프로세스 가상 메모리 버퍼와 네트워크간에 데이터를 DMA(Direct Memory Access) 할 수 있는 네트워크 인터페이스를 가진다.

VMMC-II 구조는 (그림 4)와 같다.



(그림 4) VMMC-II 구조

2. 클러스터 관리 소프트웨어(CMS)

CMS(Cluster Management Software)란 클러스터 시스템에 주어진 작업을 관리하기 위한 소프트웨어이다. 대부분의 CMS는 부하 균형 및, 체크포인팅 지원, 프로세스 이전, 작업 모니터링 및 재스케줄링, 작업 수행 및 중지, 재개시할 수 있게 한다[13]. 체크 포인팅 지원을 하게 되면 작업의 수행중에 상태를 정기적으로 저장하게 되므로 시스템 고장시 마지막 체크 포인트된 위치에서 새롭게 시작할 수 있다. 대표적인 상업용 및 연구용 CMS는 〈표 1〉과 같으며 이중 몇 가지만 소개하고 URL을 참조할 수 있게 하였다[14-32].

가. Codine[14]

이질적인 네트워크 환경을 지원하는 소프트웨어로서 벡터 또는 병렬 컴퓨터와 같은 대용량의 클러스터 시스템을 지원한다. GUI(Graphic User Interface) 기반의 관리 도구로 다양한 구조에서의 배치 큐잉 프레임워크를 지원하며 동적, 정적인 부하 균형 및 상호 작용 작업과 병렬 작업 등을 제공한다. 대표적인 기능은 다음과 같다.

- 배치(batch), 상호 작용(interactive), 병렬(Express, p4, PVM) 작업 지원
- 다중 큐 지원
- 체크포인팅(checkpointing) 지원
- 정적 부하 균형 지원
- 체크포인팅과 작업 이전으로 동적 부하 균형 지원
- 통계적 산출(utilization statistic) 지원
- 관리자와 사용자에게 X11 Motif GUI 및 명령어와 스크립트 인터페이스 지원
- POSIX 환경 지원
- DCE 기술 지원

나. Connect: QUEUE[15]

UNIX 지원형으로 세 개의 큐 타입을 제공한다. 지능형 배치 작업 스케줄링 시스템은 물리적으로 사용하는 메모리의 퍼센티지와 CPU 이용률 및 제한된 큐 작업을 기반으로 작업의 부하 균형을 관리한다.

- 배치 큐(batch queue): 스케줄링 작업과 수행중인 작업에 대해 퍼센티지 제공

〈표 1〉 클러스터 관리 소프트웨어

소프트웨어	벤더 및 연구소	URL	구분
Codine	GENIAS GmbH, Germany	http://www.genias.de/genias/english/codine/Welcome.html	상업용
Connect:Queue	Sterling Corp., USA	http://www.sterling.com/	상업용
CS1/JP1	Hitachi Inc, USA	http://www.zoosoft.com/jp1/sysmanhu.html	상업용
Load Balancer	Unison Software, USA	http://www.unison.com/main-menu/products/operations/loadbalancer/LoadBalancer.html	상업용
LoadLeveler	IBM Corp., USA	http://www.rs6000.ibm.com/	상업용
LSF	Platform Computing, Canada	http://www.platform.com/products/overview.html	상업용
NQE-Network Queuing Envn.	Craysoft Corp., USA	http://www.cray.com/PUBLIC/product-info/sw/nqe/nqe30.html	상업용
Task Broker	Hewlett-Packard Corp.	http://www.hp.com:80/wsg/ssa/task.html	상업용
Batch	UCSF, USA		연구용
CCS	Paderborn, Germany	http://www.uni-paderborn.de/pcpc/ccs/	연구용
Condor	Wisconsin State University, USA	http://www.cs.wisc.edu/condor/	연구용
DJM	MSCI, USA	http://www.msc.edu/msc/docs/djm/	연구용
DQS 3.x	SCRI, FSU, USA	http://www.scri.fsu.edu/~pasko/dqs.html	연구용
EASY	ANL, USA	http://info.mcs.anl.gov/Projects/sp/scheduler/scheduler.html	연구용
far	University of Liverpool, UK	http://www.liv.ac.uk/HPC/farHomepage.html	연구용
Generic NQS	University of Sheffield, UK	http://www.shef.ac.uk/uni/projects/nqs/	연구용
MDQS	ABRL, USA	ftp://ftp.arl.mil/arch/	연구용
PBS	NASA Ames Research Center, USA	http://www.nas.nasa.gov/NAS/Projects/pbs/	연구용
PRM	ISI, UC, USA	http://nii-server.isi.edu/gost-group/products/prm/	연구용
QBATCH	Vita Ltd, USA	http://gatekeeper.dec.com/pub/usenet/comp.sources.misc/volume25/QBATCH/	연구용

- 디바이스 큐(device queue): 물리적 디바이스를 접근하는 배치 작업
- 파이프 큐(pipe queue): 원격지에 디바이스나 파이프 큐들을 전송하기 위해 제공

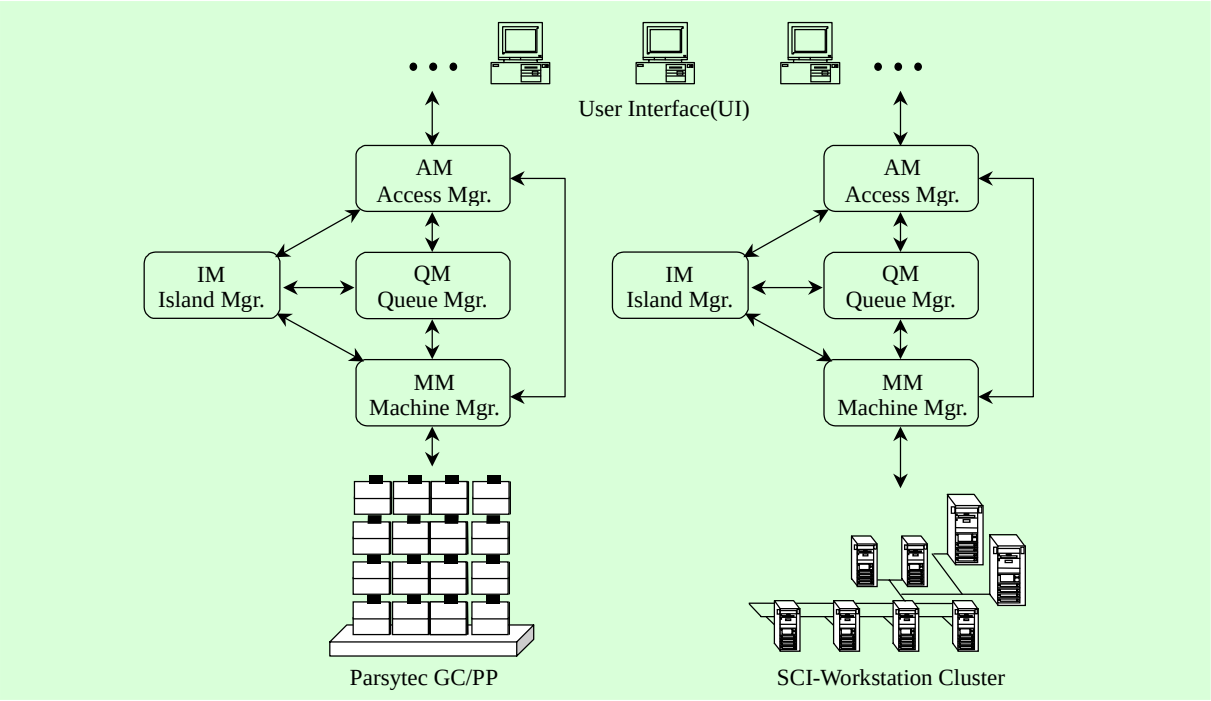
다. LoadLeveler[18]

사용자는 간단한 LoadLevel 명령어를 사용하거나 GUI를 이용하여 작업을 할 수 있으며 분산된 작업 스케줄링, 순차 및 병렬 작업 지원, 이질적인 환경을 지원한다. 작업을 수행하거나 제거할 수 있으며 작업 상태를 모니터링하고 우선 순위를 지정하거나 변경할 수 있다.

- 상호 작용적인 세션 지원: 원격 TCP/IP 허용
- 독립적인 제어: 자원의 허용 가능성에 따라 사용량을 사용자가 결정
- 중앙 집중적인 관리: 시스템 관리자는 클러스터 상에 수행중인 모든 작업을 관리

라. LSF(Load Sharing Facility)[19]

이질적인 UNIX 환경에서 분산된 부하 공유와 배치 큐잉을 지원하는 소프트웨어 패키지이며 모든 하드웨어와 소프트웨어 자원을 단일 이미지로 제공한다. 배치 작업, 상호 작용적인 작업 및 병렬 작업을 지원하고 관리한다.



(그림 5) CCS 6 Component

- 배치 작업 큐와 스케줄링 제공
- 상호작용 작업을 네트워크를 통해 분산
- 부하 균형
- 네트워크로 연결된 이질적인 자원을 투명하게 접근할 수 있도록 제공
- 부하 모니터링
- 제공되는 API로 새로운 분산 어플리케이션 작성
- 작업 분석 도구 제공

마. CCS[22]

Paderborn SCI Compute Cluster의 클러스터 처리용 자원 관리 소프트웨어로 CCS(Computing Center Software)를 사용한다. Paderborn SCI Compute Cluster는 독일에서 SCI 기술을 이용 및 지원하는 여러 업계 및 학교에서 공동으로 개발하고 있으며, 32 Siemens-Nixdorf Celsius 2000E, 프로세서는 64 Intel Pentium II, 300 MHz, 전체 성능은 19.2 GFlop/s, 분산 메인 메모리 8 GByte SDRAM, 노드는 32 Dolphin PCI/SCI 인터페이스, 500 MByte/s SCI-Link 대역폭 및 통신 네트워크가 4 × 8 이차원 torus의 기술 요소를 갖는 64-processor SCI Cluster PSC와 이질적인 24-processor SCI Cluster 및 리눅스용 SCI 드라이버 등을 제공한다. CCS는 작업의 분산과 관리를 담당하는 자원 관리 소프트웨어로 하드웨어와 독립적인 스케줄링을 하며 자원을 배분하고 데몬의 고장시 자동적인 복구 및 fault tolerance 등을 지

원한다. User Interface(UI), Access Manager(AM), Queue Manager(QM), Machine Manager(MM), Island Manager(IM), Operator Shell(OS) 등의 6 모듈로 구분되며 그 관계는 (그림 5)와 같다.

바. DJM(Distributed Job Manager)[24]

MPP 시스템에 더욱 효율적으로 사용할 수 있게 설계되어 쉬는 프로세싱 엘리먼트들(PEs)을 최소화 시키는 작업 수행 알고리즘을 사용한다. 실시간 어플리케이션이나 이질적인 어플리케이션 및 데몬 프로세스를 위하여 MPP를 준비해 둘 수 있다.

IV. 맺음말

본 고에서는 클러스터 시스템을 구축하기 위하여 필요한 전반적인 클러스터 컴퓨팅 기술에 대하여 알아보았다. 최근 클러스터 시스템은 그 응용 분야가 과학 계산용 뿐만 아니라, 멀티미디어, 대용량 데이터 베이스 및 병렬 처리 등 다양해짐에 따라 여러 분야에서 각광받고 있으며, 다음 세대를 이끌어 갈 컴퓨터 환경으로 가능성을 보이고 있다. 이러한 클러스터 시스템에서 클러스터 환경 기반을 위한 소프트웨어와 클러스터를 관리해주는 소프트웨어는 필수적이며 상업용 및 연구용으로 기술 개발되고 제공된다. 이들 기술들을 기반으로 보다 효율적이고 성능이 우수한 클러스터 시스템을 구축할 수 있을 것이다.

참 고 문 헌

[1] "High Performance Communication," http://www-csag.cs.uiuc.edu/projects/communication.html

[2] "Cluster-Ready Technology," http://www.amdahl.com/doc/products/clusterwp.html

[3] M. Dahlin *et al.* "Cooperative Caching: Using Remote Client Memory to Improve File System Performance," *In Proceedings of the First Symp. on Operating Systems Design and Implementation*, 1994.

[4] "Illinois Fast Messages(FM)," http://www-csag.cs.uiuc.edu/projects/comm/fm.html

[5] T. V. Eicken, A. Basu, V. Buch and W. Vogels. "U-Net: A User-Level Network Interface for Parallel and Distributed Computing," *In Proceedings of the Fifteenth SOSP*, Copper Mountain, CO, Dec. 1995. pp. 40−53,

[6] "Active Messages," http://now.cs.berkeley.edu/AM/active_messages.html

[7] S.H. Rodrigues, T.E. Anderson and D.E. Culler, "High-Performance Local Area Communication with Fast Sockets," *USENIX '97*, 1997.

[8] 유 혁, "클러스터 파일 시스템," 병렬처리시스템 연구회지 제 8권 1호, 1997. 2, pp. 22−31.

[9] S. Pakin. *et al.* "High-Performance Virtual Machines," Aug. 18, 1997, http://www-csag.cs.uiuc.edu/projects/hpvm/doc/hpvmdoc_toc.html

[10] "SMiLE," http://wwwbode.informatik.tu-muenchen.de/Par/arch/smile/

[11] "SVMlib," http://www.lfbs.rwth-aachen.de/~karsten/projects/SVMlib/index.html

[12] "VMMC," http://www.cs.princeton.edu/shrimp/ntvmmc/

[13] NHSE Review, "Cluster Management Software," 1996, http://nhse.cs.rice.edu/NHSEreview/CMS/

[14] http://www.genias.de/genias/english/codine/Welcome.html

[15] http://www.sterling.com/

[16] http://www.zoosoft.com/jp1/sysmanhu.html

[17] http://www.unison.com/main-menu/products/operations/loadbalancer/LoadBalancer.html

[18] http://www.rs6000.ibm.com/

[19] http://www.platform.com/products/overview.html

[20] http://www.cray.com/PUBLIC/product-info/sw/nqe/nqe30.html

[21] http://www.hp.com:80/wsg/ssa/task.html

[22] http://www.uni-paderborn.de/pcpc/ccs/

[23] http://www.cs.wisc.edu/condor/
[24] http://www.msc.edu/msc/docs/djm/
[25] http://www.scri.fsu.edu/~pasko/dqs.html
[26] http://info.mcs.anl.gov/Projects/sp/scheduler/scheduler.html
[27] http://www.liv.ac.uk/HPC/farHomepage.html
[28] http://www.shef.ac.uk/uni/projects/nqs/
[29] ftp://ftp.arl.mil/arch/
[30] http://www.nas.nasa.gov/NAS/Projects/pbs/
[31] http://nii-server.isi.edu/gost-group/products/prm/
[32] http://gatekeeper.dec.com/pub/usenet/comp.sources.misc/volume25/QBATCH/