

# 클러스터 시스템 개요

권 오 영 (e-mail: oyeon@naver.ac.kr)  
한국기술교육대학교 정보기술대학원 교수

I. 서 론

II. 클러스터 시스템 구조

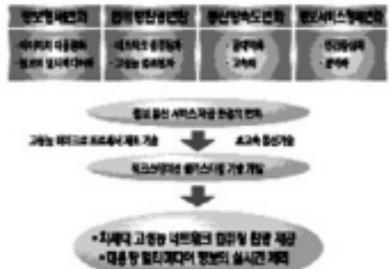
III. 클러스터 내부 통신 메커니즘

IV. 클러스터 기반 저장 시스템

V. 결 론

## I. 서 론

1990년대에 들어오면서 인터넷의 활성화로 인하여 컴퓨팅 환경이 기존의 단순 텍스트 디아트의 고속 처리에서 다양한 매체를 고속 네트워크를 통하여 보다 인간에게 친숙한 형태로 제공되며 방향으로 변화되고 있다. 여기에 더불어 마이크로 프로세서의 급속한 성능 향상과 시스템 비스의 속도에 친숙한 고속 네트워크 장비들의 출현으로 일반 POI 차기형 워크스테이션들을 고속 네트워크로 연결하여 고성능 컴퓨팅을 제공하는 클러스터링 시스템에 대한 관심이 급증하고 있다. 그림 1은 이러한 컴퓨팅 환경 및 제조 기술의 발전방향



■그림 1 컴퓨팅 환경 및 제조 기술의 발전 방향

을 보여주고 있다.

기존의 고성능 컴퓨터인 핵심형 슈퍼컴퓨터나 대규모 병렬컴퓨터는 높은 가격, 유지보수의 어려움, 요구되는 성능과 원을 수 있는 성능 간의 차이 등 여러 문제

점을 안고 있다. 반면에 클러스터 시스템은 슈퍼컴퓨터나 병렬컴퓨터 보다 확장성 속도에서 우수하고 단위 시스템의 성능 향상과 분산생활처리 기술의 발달로 고속 컴퓨팅 및 멀티미디어 데이터 처리를 효과적으로 수행할 수 있게 되었다. 즉, 클러스터 컴퓨팅은 민족화하는 컴퓨팅 환경에 잘 적용할 수 있는 시스템으로, 원하는 물을 분야에 적합하게 commodity component들을 구입하여 시스템을 구성할 수 있다. U. C. Berkeley의 NOW (Networks Of Workstations) 프로젝트는 클러스터링 시스템을 기반으로 고성능 컴퓨터를 구축하고자 한 대표적인 연구의 하나다. 클러스터링 시스템은 효율적인 가격 대 성능 비, 음을 도케인에 따라 시스템을 갖출 수 있는 융통성 (elasticity), 시스템 모듈화로 시 이론 복구할 수 있는 가용성 (availability), 요구되는 성능에 맞추어 시스템을 자유롭게 꾸밀 수 있는 적합성 (affordability)과 확장성 (scalability) 등을 강점으로 갖는다.

본 논문에서는 클러스터 컴퓨팅을 구성하는 요소들에 대한 간단한 소개를 하고자 한다. 2장에서는 클러스터 시스템의 구조를 간단히 살펴보고, 각 구성요소의 역할을 간략히 기술한다. 3장에서는 클러스터 시스템의 가장 중요한 부분 중의 하나인 내부 통신 프로토콜에 대한 소개를 하고, 4장에서는 클러스터 시스템을 이용한 파일 시스템 지원방안에 대하여 기술한다.

## II. 클러스터 시스템 구조

일반적인 클러스터 시스템은 고속의 시스템 영역 네트워크를 통하여 각 노드들을 물리적으로 연결하고 있다. 그림 2는 클러스터 시스템의 구조를 잘 보여주고

## 일반적인 클러스터 시스템은 고속의 시스템 영역 네트워크를 통하여 각 노드들을 물리적으로 연결하고 있다.

있다. 일반적으로 각 노드는 파이어스(Linux)를 탑재한 PC들이 활용되고, 외부 네트워크 등 신 시스템 관리 및 모니터링을 위하여 이더넷(Ethernet)을 활용한다. 그 외에 클러스터 시스템의 사용 목적에 따라 필요한 시스템들이 클러스터에 추가된다.

클러스터에 사용되는 고속 네트워크로는 Myrinet, SCI, ATM 등이 있으며, 기존 통신 프로토콜을 바로 사용할 수 있다는 점에서 Fast Ethernet, Gigabit Ethernet 등이 이용되기도 한다. Myricom사의 Myrinet card는 사용자가 network card를 위한 firmware를 재구성할 수 있다는 장점을 가지고 있어서 현재 구성되고 있는 많은 클러스터 시스템들이 Myrinet을 활용하고 있다. 일반적으로는 Gigabit 이상의 전송속도를 갖는 장비들이 클러스터를 구성하는 데 사용되는 추세다. 최근에는 Gigabit per second 이상의 전송속도를 갖는 장비도 출시되고 있다. 결국 클러스터 시스템의 성능을 결정하는 가장 중요한 부분 중의 하나가 고속 네트워크 장비의 성능을 효과적으로 이용할 수 있는 통신 소프트웨어다. 통신 소프트웨어의 기원에 대한 내용은 3장에서 다루기로 한다.

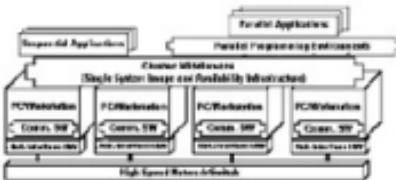
클러스터 미들웨어는 각 노드의 운영체제와 클러스터 응용의 중간에 위치하여 클러스터의 이용목적에 맞게 다양하게 구성될 수 있다. 미들웨어는 주로 단일 이미지 제공, 가상화 제공, 클러스터 파일 시스템 제공,

Sun의 Solaris MD나 NOW 팀이 개발한 Gridix는 분산체계 거울에서 클러스터 시스템의 단일 이미지 제공 기능을 지원한다. 소프트웨어 분산 컴퓨�헤리 시스템이나 클러스터 파일 시스템은 Run time system (library) 형태로 제공되고 있으며, 차원 관리나 작업 스케줄링을 위한 미들웨어로 LSF (Load Sharing Facility)가 많이 사용되고 있으며, PBS (Portable Batchting System)도 널리 이용되고 있다.

클러스터 시스템은 분산시스템의 특성을 갖고 있으므로, 다수의 노드를 효과적으로 관리하는 데 많은 부담이 발생한다. 이를 효과적으로 처리하고, 클러스터 시스템을 하나의 컴퓨팅 차원으로 통합 관리할 수 있는 클러스터 관리 소프트웨어는 클러스터 시스템의 중요한 구성을 묘사할 것이다. 현재 유통 수준에서 많은 인구가 이루어지고 있다.

클러스터의 행렬 프로그래밍 환경으로는 MPI (Message Passing Interface)가 가장 널리 활용되어, PVM (Parallel Virtual Machine)도 사용된다. 최근 클러스터를 위한 Software DSM 시스템도 활발히 연구가 되고 있다. OMPI의 컴파일 도구는 각 노드에서 프로그램 기반에 널리 이용되고 있다. TotalView와 같은 행렬 프로그램의 디버깅을 지원할 수 있는 디버깅 클러스터 시스템에는 존재하지 않는다. 하지만 클러스터를 행렬 응용에 널리 활용하기 위해서는 클러스터를 위한 행렬 디버깅도 행렬 프로그래밍 환경으로 제공되어야 할 것이다. 이론적인 행렬 프로그래밍 환경을 이용하여 Weather Forecasting, Quantum Chemistry, Ocean Modeling 등의 Grand Challenge 문제를 해결하려는 시도가 많이 이루어지고 있다.

클러스터 시스템을 슈퍼컴퓨터나 벤처 컴퓨터로 대안으로 활용하는 방안은 이외에 고성능 앱 서비스, 월 캐시, 월 단식 연산, 행렬 데이터베이스, 미션 등 핵심 연동되는 고성능 서비스로 활용하라는 노력이 활발히 이루어지고 있다. 또한 멀티미디어 서비스에 대한 요구가 꾸준히 증가하면서 클러스터 시스템을 VoD와 같은 주문형 멀티미디어 응용에 활용하는 사례들이 늘어나고 있다.



■그림 2 클러스터 시스템의 구성모습

차원관리, 작업 스케줄링, 클러스터 관리 등의 작업을 수행한다. 이러한 미들웨어의 기능은 운영체계 거울에 추가될 수도 있으며, 유통 수준에서 구현될 수도 있다.

## 클러스터 시스템을 슈퍼컴퓨터나 병렬컴퓨터의 대안으로 활용하는 방안 이외에 고성능 웹 서버, 웹 캐쉬, 웹 템색 엔진, 병렬 데이터베이스 머신 등 웹과 연동되는...

ETRI에서 개발한 MovieBo 시스템이 클러스터 시스템을 VoD에 응용한 대표적인 사례다.

### III. 클러스터 내부 통신 메커니즘

기존의 대용량 병렬 처리 시스템과 클러스터 시스템과의 큰 차이점 중의 하나가 노드간 통신 메카니즘이고, 이들 간의 성능차를 얼마나 줄이느냐가 전체 성능을 크게 좌우하게 된다. 클러스터링 시스템의 노드간 통신은 작은 치연시간(Latency)과 높은 대역폭(Bandwidth)을 지원해야 한다. 이와 같은 요구사항은 특히 용융 도메인의 통신 특성에 따라 중요성이 달라진다. 즉 멀티미디어 서비스에서 클러스터링 시스템을 활용할 경우에는 차수적인 높은 대역폭을 통신 메카니즘에서 지원해 주어야 하며, 병렬컴퓨팅 서비스의 경우에는 작은 치연시간에 더욱 비중을 두어야 한다. TCP/IP와 같은 날리 사용되는 표준 통신 프로토콜을 클러스터 내부 통신 메카니즘으로 이용할 수 있으나, TCP는 과도한 프로토콜 처리 오버헤드와 메시지의 메모리 복사 오버헤드로 인하여 작은 치연시간과 높은 대역폭을 얻기 힘든 통신 메카니즘이다. 거울을 무회할 수 있고, Myrinet, SCI 등의 고속 네트워크 장비를 효과적으로 이용할 수 있는 하위 수준의 강력 통신 프로토콜들이 클러스터 시스템 개발과 함께 활발히 연구되고 있다. 대표적인 클러스터 내부 통신 메커니즘으로 AM, FM, PM, BIP, U-net, CLCP 등이 존재하며, 최근에는 인텔의 주도로 VIA(Virtual Interface Architecture)라는 클러스터 내부 통신 프로토콜을 정의하고 있다.

AM(Active Message)은 Berkeley의 NOW 팀이 개발한 통신 소프트웨어로써 고속 데이터 전송을 위해서 버퍼링을 하지 않는 특징을 갖고 있다. 또한, 도착되는 메시지의 처리를 빼놓게 하기 위해서 메시지의 헤더에 메시지를 처리할 한글리의 주소를 삽입한다. AM은 NI(Network Interface) 카드 위에 물려갈 수도 있을 뿐만 아니라, 더 높은 계층의 위에 물려갈 수도 있고

(AM over U-Net), AM 위에 다른 네트워크 프로토콜이 물려갈 수도 있다<sup>1)</sup> 때문에 TCP/IP over AM 상당히 유통성 있는 프리미티브라고 할 수 있다. 최근 보고된 AM(over Myrinet)의 성능(SBus 기반은 1.0s Latency와 200 Mbps의 대역폭이다. PM(PCI Fast Message)은 Illinois주립대학에서 워크스테이션 클러스터링 및 병렬 컴퓨팅의 고속 통신을 위해서 개발한 통신 소프트웨어이다. PM은 비교적 작은 카드의 전송을 목적으로 만들어졌다. 작은 카드를 전송하는 데 있어서 DMA는 그 효과를 기대할 수가 없기 때문에, PM은 전송하는 쪽에서는 PIO(Programmed I/O)를 이용해서 카드를 네트워크 인터페이스의 메모리에 옮기능된다. 받는 쪽에서는 네트워크 인터페이스에서 호스트 메모리(Memory)로 DMA를 이용한다. 또한, 전송된 카드의 신뢰성을 위해서 버퍼에서 흐름 제어 기능을 수행한다. PM의 가장 큰 단점은 큰 카드의 전송에는 적합치 않다는 것이다. 최근에 보고된 PM(over Myrinet)의 성능(PCI slot은 9.43ns의 Latency와 200Mbps의 대역폭이다.

PM은 일본 RWC(Real-World Computing)에서 워크스테이션 클러스터링 프로젝트를 위해서 개발한 고속 통신 프리미티브다. PM은 기본적으로 Myrinet API에서 제공하는 구조를 그대로 이용하고 있으므로, 전송하는 쪽과 받는 쪽 모두 DMA를 사용한다. 그러나 Immediate Sending 방식과 NACK 흐름 제어 기능을 이용해서 신뢰성과 성능의 향상을 이루었다. 또한 동시에 통신할 수 있는 프로세스 수의 제한을 일으키기 위해서 채널 Context의 한정 방향을 사용하였다. PM은 Myrinet을 이용한 보고된 실험 결과에 의하면 941 Mbps(PCI slot, Pentium)의 성능을 보이고 있다.

BIP(Basic Interface for Peer-to-peer)는 메시지 전송에 의한 병렬 컴퓨팅을 대상으로 노드간 통신에 필요한 네트워크 인터페이스 구현을 목표로 한다. 즉 일계성이 부족한 병렬 음성들을 효과적으로 지원하기 위한 네트워크 프로토콜 구조를 구현한다. 이를 위해서

## 내포적인 클러스터 내부 통신 메커니즘으로 AM, FM, PM, BP, U-net, CLCP 등이 존재하며 최근에는 인텔의 주도로 VIA(Virtual Interface Architecture)라는 클러스터 내부 통

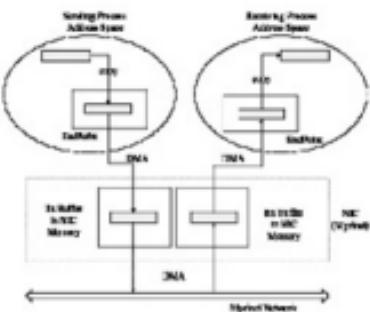
IP는 Zero Memory Copy Big Message Delivery Based on Pipeline, Small Message Delivery 등의 기술들을 구현하였다. 현재까지 보고된 통신 소프트웨어들에서는 IPoI 가장 우수한 전송성능을 보인다.

U-net은 Cornell 대학에서 개발한 통신 프라미티브 코서, 사용자 수준으로 네트워크를 추상화한 프라미티브이다. 사용자 프로세스는 U-Net에서 제공하는 인터페이스를 이용해서 캐시를 전송할 경우, 전송되는 캐시는 네트워크 인터페이스에서 재생하는 데이터 맵크 계층의 서비스만을 받고 직접 목적지의 사용자 프로세스에게 전달된다. 현재까지의 구현 사례를 보면, Cornell 대학에서 SBA-200 ATM Card를 이용해서 119.4 Mbytes/121 Mbytes (99%)의 대역폭을 보이고, UCSB대학에서 CIIA-200 ATM Card를 이용해서 140 Mbytes/140.4 Mbytes (100%)의 대역폭을 보이고 있다.

CLCP는 ETRI의 클러스터 시스템인 CROWN을 위하여 개발된 통신 메커니즘으로 U-net에 기반을 두고 있으며, Myrinet을 통신 하드웨어로 사용한다. 더욱이 멀티비디어 서비스를 위하여 U-net에서 제공하지 않는 인터럽트 방식의 메시지 전송을 지원한다. CLCP를 경유하는 메시지는 커널을 무화하는 통신 경로를 갖는다. 그림 3은 호스트 내에서 메시지의 흐름을 보여준다. 송신을 위해서 통신 프로세스는 먼저 송신하고자 하는 메시지를 자신의EndPoint의 송신버퍼에 복사한다. 복사된 메시지의 정보는EndPoint에 있는 전송큐(Tx Queue)에 기록된다. 복사 및 정보 기록 작업은 송신 API에서 수행된다. EndPoint에 메시지가 복사되며 사용자 수준의 송신은 마무리된다. 송신 API에서 제어가 리턴되면 통신 프로세스는 다른 작업을 수행한다. 메시지를 실제 네트워크로 전송하는 작업은 CLCP Firmware에 의해 수행된다. 모든 통신 프로세스의 전송 및 수신 큐는 물리적으로 NIC의 메모리에 존재하므로, CLCP Firmware는 이 큐들에 쉽게 접근할 수 있다. CLCP Firmware는 NIC에 존재하는 링링 큐를 포함한 모든 큐를 풀링하는 작업을 피풀이한다.

이제, 송신한 메시지에 대한 정보(DMA 주소, 크기, 차 날번호 등)를 보고 DMA를 통해 NIC의 Tx 버퍼로 복사한 후, 다시 네트워크 DMA를 통해 전송을 한다.

수신작업은 위해서 수신 프로세스는 자신의 End Point에 있는 수신 큐(Rx Queue)에 도착한 메시지가 복사된EndPoint의 수신 가능 버퍼 공간 정보를 미리



■그림 3 CLCP의 노드간 메세지 경로

기록해 둔다. 인터럽트를 이용한 수신을 위한 경우에 수신 프로세스는 select 시스템 호출을 하고 메시지의 도착을 기다린다. 수신을 위한 준비가 완료된 상태에서에서 메시지가 도착하면 Firmware는 도착된 메시지의 애더 정보를 보고 메시지를 수신할 목적 프로세스의 EndPoint를 찾아낸다. 그리고, EndPoint가 가지고 있는 수신 큐의 정보를 가지고 DMA를 물리 주소를 알아낸 후 EndPoint의 수신버퍼로 DMA를 한다. 메시지를 기다리던 수신 프로세스는 자신의 수신 큐를 풀링하여 메시지가 도착했음을 알고, 도착된 메시지를 자신의 다른 주소 공간으로 복사한 후 사용된 수신버퍼 공간을 다시 수신 가능 버퍼 공간으로 환원함으로 수신 작업이 끝난다. CLCP는 메시지 길이가 8K Byte일 경우 450 Mbytes의 대역폭을 보이며, 4 Bytes의 짧은 메

**클러스터 시스템은 각 노드들이 지리적으로 근접한 위치에 있다. 그러므로 각 노드의 지역 저장 장치들을 연동하여 대규모 스토리지 서버로 활용할 수 있다.**

시체에 대해서 100% 수준의 치안시간을 보인다.

#### IV. 클러스터 기반 자정시스템

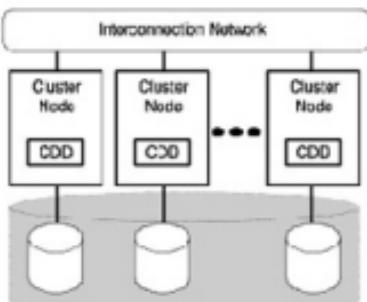
클러스터 시스템은 각 노드들이 지리적으로 근접한 위치에 있다. 그러므로, 각 노드의 지역 저장 장치들을 연동하여 대규모 스토리지 서버로 활용할 수 있다. 즉, 클러스터를 위한 분산 Software RAID를 구성하면, 대규모의 스토리지 장치를 얻을 수 있다. 또한 클러스터 시스템이 내포하고 있는 병렬성을 이용하여 위하여 클러스터를 위한 병렬 파일 시스템에 대한 연구도 활발히 이루어지고 있다.

##### 1. Trojans

Trojans 시스템은 USCO에서 개발되고 있는 16대의 PO를 Fast Ethernet으로 연결한 Linux 클러스터 시스템으로 디바이스 드라이버 수준에서 단일 입출력 공간을 제공하고 있다. 그림 4에서 보듯이, 각 노드에 CDD(Cooperative Disk Driver)가 기기별 전역 가상 공유 디스크를 제공하고 있다. CDD는 디바이스 드라이버로 구성되었으므로 사용자는 일반 디스크를 쓰는 것과 동일하게 CDD가 제공하는 디스크를 활용할 수 있다는 장점을 갖는다. 하지만 커널 의존성이 생기는 것이 단점이다.

##### 2. Galley

Galley는 현재 IBM의 RS/6000, SP-2 등 Parallel Supercomputer에서 수용이 되는 병렬 파일 시스템이다. 모듈들이 모아서 서비스 파일을 구성하여, 서비스 파일은 하나의 노드에 저장된다. 그리고, 노드들 사이에 저장된 서비스 파일들이 모아서 하나의 파일을 형성하는 3차원 파일 구조를 가지고 있는 것이 Galley의 특징이다. 또한 여러 가지 접근 형태를 지원하여 병렬화에 대한 세밀한 제어를 가능하게 한다. 시스템의 입출력 서버와의 연결은 메시지지상을 이용하여 클라이언트와 서버사이



■그림 4. Trojans의 CDD를 이용한 전역  
기상 디스크(global virtual disk)

이의 통신을 위하여는 헌디파이브웨어를 사용한다. 현재는 Linux 클러스터 시스템에서도 사용할 수 있다. 하지만 이전 트레이는 Sigena를 이용함으로 뛰어난 성능을 보이지는 않는다.

##### 3. Symphony

Symphony (7)는 Univ. of Texas에서 개발한 멀티 미디어 파일 시스템으로 사택스트 파일뿐만 아니라 멀티미디어 파일을 지원하는 통합된 파일시스템으로 Sun SPARC들을 연결한 클러스터 시스템을 이용하여 구현되었다. Symphony에서는 시스템의 복잡성을 단순화시키기 위해서 페미터 구조를 활용하였다. Symphony는 파일 서버 인터페이스, 데이터 타입 상세 레이어, 데이터 타입 드립 레이어로 구성된다. 디스크 서비스 시스템 (Disk Subsystem)에서는 서로 다른 데이터 타입들 간의 효율적인 데이터 저장과 삭제, 데이터 타입 모듈에 대한 복구, 폐쇄 데이터 관리, 디스크 스캐닝 등의 기능을 수행한다. 버퍼 서비스 시스템 (Buffer Subsystem)에서는 다양한 데이터 타입에 대한 캐싱 정책을 적용하기 위해서 데이터 타입에 대한 캐싱 정책을 가지고 있고 부하 정도에 따라서 동적으로

## 클라스터 시스템은 고성능 컴퓨팅을 요구하는 다양한 분야에 널리 활용되고 있으며 최근에는 웹의 사용이 증가되면서 확장성 있는 웹 서비스를 위한 시스템으로도..

개성 장점을 적음하기 위해 비파티 할당한다. 자원 관리(Resource Manager)에서는 실시간 응답을 보장하기 위해 충분한 자원 즉, 비파티 공간과 디스크 대역폭(Disk Bandwidth)을 보장해 주는 기능을 한다.

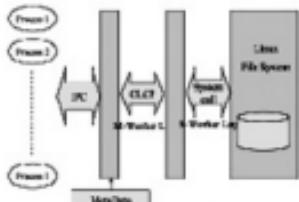
### 4. Crown 파일 시스템

CROWN 파일 시스템은 멀티미디어 서비스를 위하여 모디오(비)모디오와 같은 인속 메체 파일들을 저장하고 관리할 수 있는 사용자 수준의 파일 시스템으로 그림 5와 같이 구성된다. 하나의 파일은 CROWN 시스템을 구성하는 각각의 노드에 분산 저장되어, 파일 이름, 파일 크기, 파일의 분산정보 등과 같은 메타 데이터는 관리 서버에 저장된다.

CrownFS는 M-workers와 S-workers로 불리는 두 key로 구성되어 있다. M-Workers는 파일 시스템의 인터페이스 역할을 하는 모듈로서, 파일 접근에 대한 요구를 받고 그 파일의 메타 데이터 정보를 분석한 후 실제로 사용자가 접근을 원하는 부분을 소유하고 있는 S-Worker에게 파일을 일정으로 넘긴다. S-worker로부터 요구한 파일의 자료가 도착하면 이를 다시 요구한 프로세스에게 전달하는 역할도 한다. S-Worker는 실제 코킬 디스크를 관리하면서 M-Worker들로부터 모든 파일 요구를 받고 디스크에서 해당 파일을 읽은 후 다시 M-Worker로 전송하는 역할을 한다.

### V. 결 론

본 논문에서는 현재 병렬 컴퓨터의 대안으로 제시되는 클라스터 시스템을 구성하기 위하여 필요한 요소들에 대하여 간단히 살펴보았다. 우선 물리적으로 시스템들을 연결하기 위한 네트워크 장비가 필요하며, 대부분 Gigabit 이상의 전송속도를 갖는 고속 시스템 영역 네트워크 장비들을 이용한다. 물리적으로 연결된 시스템을 위한 고속 통신 소프트웨어가 클라스터 시스템의 성능을 좌우하는 기본 요소가 되며, 클라스터를 위



■그림 5. CrownFS의 구조

한 대부분의 미들웨어 및 응용들은 이 고속 통신 소프트웨어를 활용한다. 클라스터 미들웨어는 일반 사용자 가 클라스터 시스템을 단일 컴퓨팅 시스템으로 활용하기 위한 기본 구조를 제공하며, 일반 사용자는 MPI, PVM, 또는 DSM 라이브러리를 이용하여 병렬 응용을 작성한다.

클라스터 시스템은 병렬 컴퓨터의 대안으로 구성되기 시작하였다. 그러나만 클라스터 시스템은 일반 off-the-self commodity component들을 이용하여 사용자의 요구에 따라서 원하는 형태의 컴퓨팅 시스템을 구성할 수 있다는 장점을 가지고 있기 때문이다. 이러한 장점으로 인하여 클라스터 시스템은 고성능 컴퓨팅을 요구하는 다양한 분야에 널리 활용되고 있으며, 최근에는 웹의 사용이 증가되면서 확장성 있는 웹 서비스를 위한 시스템으로도 널리 활용되고 있다. 앞으로도 클라스터 시스템을 이용한 다양한 응용 개발 및 클라스터 미들웨어에 대한 개발이 더욱 많이 이루어져야 한다. 앞으로 클라스터 컴퓨팅에 대한 참고가 될 만한 사이트들의 주소를 제공함으로써 참고문헌을 대신하였다. 본 논문에 언급된 내용들에 대한 참고문헌은 이 사이트들을 참조하면 될 것이다.

- <http://www.dcs.monash.edu.au/~mfrunsas/index.html>
- <http://www.cs.vt.edu/~dmcng/index.html>
- <http://www.bewulf.org/>
- <http://linux-ha.org>