

# 이중 채널 이더넷 기반 가상서버를 위한 분산 고장 감내 미들웨어의 구현

함명호<sup>0</sup> 김진용 최보곤 신현식  
서울대학교 컴퓨터공학부

(mhham<sup>0</sup>, sugar, bgchoi)<sup>0</sup>@cslab.snu.ac.kr, [shinhs@snu.ac.kr](mailto:shinhs@snu.ac.kr)

## Implementation of Distributed Fault-Tolerant Middleware for Dual Channel Ethernet based Virtual Server

Myung-Ho Ham<sup>0</sup> Jin-Yong Kim Bo-Gon Choi Heon-Shik Shin  
School of Computer Science and Engineering, Seoul National University

### 요 약

긴박한 임무 상황(mission critical)의 시스템 뿐만 아니라, 웹 서버등의 고가용성, 고신뢰성을 위한 가상 서버의 구축도 관심이 되고 있다. 본 연구에서는 가상서버의 고가용성, 고신뢰성을 보다 향상시키기 위해 가상서버 클러스터의 각 구성 노드들을 이중의 네트워크 채널로 중복 시켜 네트워크 고장에 대한 신뢰성을 향상시켰다. 관리자 노드와 작업노드 풀로 구성되는 시스템의 각 노드와 이중 채널로 구성된 네트워크에 대한 결함검출과 결함복구를 위한 분산 결함 허용 미들웨어를 구현하였고, 적응형 고장 감내(Adaptive Fault Tolerance) 기법을 사용하여 다양한 임무 상황에서의 자원 효율성을 향상시켰다.

### 1. 서 론

항공 관제 시스템, 비행 제어 시스템, 무기체제 시스템 등의 긴박한 임무상황(mission critical)인 특수한 상황에서는 시스템의 고장이 인명피해나 막대한 경제적 손실을 초래하기 때문에 고가용성, 고신뢰성의 결함 허용 시스템을 구축하여 사용하여 왔다. 하지만 인터넷의 폭발적인 성장으로 인하여 웹 서버의 과부하가 치명적인 문제가 되었고, 인터넷상 사업(e-business)의 중요성 증가로 인하여 고가용성, 고신뢰성 서버 구축도 피할 수 없게 되었다[1].

컴퓨터의 성능을 향상시키기 위해서 SMP(Symmetric Multiprocessing), MPP(Massive Parallel Processing), NUMA(Non-Uniform Memory Access)등 여러가지의 병렬 처리 기술들이 연구되었다. 1980년대부터 컴퓨터 클러스터링이 개발되기 시작했는데 이 기술의 여러가지 장점으로 인하여 많은 시스템들이 클러스터링되어 사용되고 있다. 클러스터링의 장점으로는 PC나 워크스테이션 등 모든 종류의 컴퓨터를 사용할 수 있다는 점, 가격대 성능면에서 기존의 대형 컴퓨터를 능가한다는 점, 업그레이드와 확장성이 우수하다는 점등을 들 수 있다. 특히 오픈 소스인 리눅스를 이용한 클러스터링이 주목을 받고 있는데 이는 가격대비 성능이 탁월하고 유닉스 시스템으로의 이식 용이, 리눅스를 지원하는 다양한 응용 프로그램, 그리고 소스가 완전히 공개돼 있어 다른 운영체제에 비해 각종 클러스터링 기법을 구현하기가 용이하다는 장점을 갖고 있기 때문이다.

가상서버(Virtual Server)란 클라이언트 입장에서 하나의 서버로 보이는 서버들의 집합으로 느슨하게 연관된(loosely coupled) 독립된 서버들의 클러스터에 구축된 확장 가능하고 가용성이 높은 서버이다. 이 클러스터 구조는 클러스터 밖의 클라이언트에 투명하다. 따라서 클라이언트 응용 프로그램은 마치 클러스터가 하나의 고성능, 고가용성의 서버인 것처럼 클러스터와 상호 작용한다[1].

리눅스 가상 서버(Linux Virtual Server)와 울트라몽키(UltraMonkey) 프로젝트에서는 한 개 이상의 리눅스 가상 서버와 다수 개의 리얼 서버(Real Server)들로 구성된다. 가상 서버는 리얼 서버 풀(Real Server Pool)을 관리하며, 서비스 요청이 왔을 때, 이를 선택된 리얼 서버에 전달하는 작업을 수행한다[그림1].

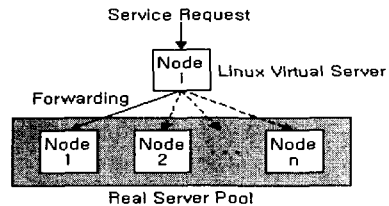


그림 1 리눅스 가상서버

이러한 가상 서버 프로젝트에서 노드의 고장이나 네트워크의 고장에 대한 감내기법은 해당 노드를 리얼 서버 풀(Real Server Pool)에서 제거하여 스케줄링 대상에서 제외하는 것이다. 그러므로 한 개의 네트워크 인터페이스

카드(NIC)를 가진 노드의 네트워크 고장은 노드의 고장과 동일한 결과를 갖게된다. 따라서 보다 엄격한 상황하에서의 임무를 고려할 때 네트워크에 대한 고장감내 연구의 필요성이 있다.

근래에 이중 채널을 이용하여 고장감내 미들웨어를 설계한 연구가 있었다. 이 연구에서는 상용제품 COTS(Commercial-Off-The-Shelf)의 네트워크 카드를 중복하여 네트워크 계층 2의 위치에서 미들웨어를 구성하였다[2]. 그러나 클러스터링된 가상서버를 구축하여 서비스를 제공하기 위한 완전한 시스템이 아닌 기초적인 연구를 수행하였다.

본 연구에서는 보다 높은 수준의 신뢰성과 가용성을 만족시키는 가상서버 구현을 위해 이중채널을 기반으로 한 분산 고장 감내 미들웨어의 설계와 구현을 하였다. 특히 이전 연구[3]에서 수행한 고장감내 미들웨어를 위한 기초적인 연구를 확장하여 서비스 제공을 위한 완전한 시스템 구축과 작업 노드의 확장성을 제공하였다. 어플리케이션 레벨에서의 미들웨어 구축으로 다양한 OS에 포팅이 용이하며, 다양한 고장감내 기법이 적용가능 하다. 더욱이 적응형 고장 감내(Adaptive Fault Tolerance) 기법을 적용하여 다양한 임무 상황에서 적합한 고장 감내 기법을 적용함으로써 자원의 효율적인 사용을 극대화하였다.

**2. 이중 채널 기반 가상 서버의 구조**

높은 가용성의 시스템은 잦은 동작의 이상뿐만 아니라 얼마나 신속하게 복구되느냐에 의존적이다. 실제 서비스를 제공하는 노드에서 동작의 이상이 발견되더라도 여분의 노드들이 대신 그 서비스를 수행한다면 가용 상태가 계속 유지된다. 두 개의 채널을 가진 노드들은 고장을 감내할 수 있게 구성된 네트워크를 통해 통신한다. 노드들 사이의 신뢰성 높은 제어 명령이나 주기적인 데이터를 교환하기 위해 지역적인 제어 네트워크의 구성이 필요한데[5], 그림 2의 네트워크 구조는 (1) 허브 결합, (2) 네트워크 링크 결합, (3) 네트워크 인터페이스 카드의 결합을 감내하기 위해 고안된 구조이다.

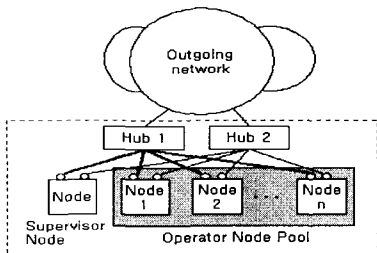


그림 2 이중 채널 기반 가상 서버의 구조

본 시스템은 기능별로 관리자 노드와 작업 노드로

나뉘어 진다. 관리자 노드는 작업 노드 풀을 관리하며 외부로부터 전달되는 요청을 선택된 작업 노드에게 전달하는 역할을 한다. 작업 노드풀은 현재 서비스 가능한 노드들이며 같은 동작을 수행하는 프로세스들이 준비된다. 실행 결과는 관리자 노드에게 반환된다.

**3. 이중 채널 기반 가상 서버를 위한 고장 감내 미들웨어**

**3.1. 고장 감내 미들웨어의 구조**

본 연구에서는 가상 서버의 높은 신뢰성을 위하여 이중 채널을 이용한 시스템을 구성하였다. 또한 다양한 임무상황에서 최상의 신뢰성과 자원 효율성을 보장하기 위해 적응형 고장 감내(Adaptive Fault Tolerance) 기법을 도입하였다. 그림 3은 본 시스템의 고장 감내 미들웨어의 논리적인 구조를 나타낸 것이다.

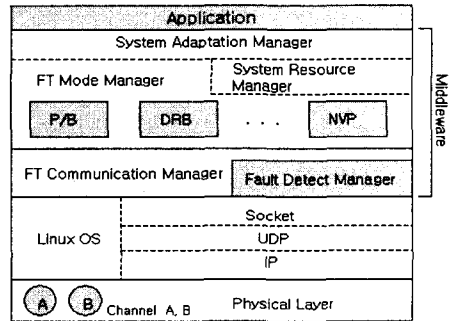


그림 3 고장 감내 미들웨어의 구조

고장 검출 관리자(Fault Detect Manager)는 UDP 하트 비트(Heartbeat) 메시지를 사용하여 노드들의 상태를 검사하고 AST(Active Stations Table)를 유지한다.

고장 감내 통신 관리자(FT Communication Manager)는 네트워크 채널의 고장이거나 노드의 고장시에도 관리자 노드와 작업 노드간의 신뢰성 있는 통신을 보장한다.

시스템 자원 관리자(System Resource Manager)는 관리자 노드와 작업 노드 풀에 대한 갱신된 정보를 관리하며, 상위 애플리케이션 계층에 정보를 제공한다.

고장 감내 기법 관리자(FT Mode Manager)는 다양한 고장 감내 정책들을 수행하게 해준다.

시스템 적응 관리자(System Adaptation Manager)는 클라이언트의 작업요청을 처리할 적절한 고장 감내 기법을 결정한다.

**3.2. 고장 감내 미들웨어의 구현**

가상서버(Virtual Server)에서 클라이언트로부터 받은 패킷을 작업노드로 전달하는 하는 방법은 크게 다익렉트 라우팅(Direct Routing), NAT(Network Address Translation), IP tunneling 등으로 나뉘어 진다. 그러나 이들 기법은 커널의 수정이나 패킷 전달을 위한 별도의 프로토콜이 필요하다. 본 연구에서 구현한 고장감내 미들웨어는 서로 다른 운영체제로의 이식성, 다양한

스케줄링 기법과 고장 감내 기법의 확장성을 위하여 커널 계층이 아닌 어플리케이션 계층에서 미들웨어를 구현하였다.

이중 채널을 위한 고장 검출은 고장 검출 관리자에 의해 이루어진다. 고장 검출 관리자는 하트비트(Heartbeat) 프로토콜을 사용하여 네트워크 고장이나 노드 고장을 검출한다. 이때 이중 채널로 구성된 각 노드들의 네트워크 고장검출을 위한 하트비트 메시지의 개수는  $2n$ 개이다( $n$ 개의 노드). 여기서 메시지 누락(Message omission) 문제를 고려하면  $k$ 개의 메시지 누락을 견디기 위해,  $k+1$ 개의 메시지를 보내야 한다[6]. 네트워크 결함이 발생한 경우 채널을 스탠바이(standby) 채널로 전환을 해야 한다. 본 시스템에서는 이 경우 모든 노드들의 네트워크 채널을 스탠바이(standby) 채널로 전환하는 방법을 사용하였다.

관리자 노드와 작업노드 풀(pool) 사이의 신뢰성 있는 통신을 지원하는 고장 감내 통신 관리자는 노드들의 현재 상태와 네트워크의 상태를 알고 있어야 한다. 그러므로 고장 검출 관리자와 고장 감내 통신 관리자는 AST(Active Station Table) 정보를 공유해야만 한다[그림 4]. 고장 검출 관리자는 고장 검출시 고장 감내 통신 관리자에게 시그널을 보내 고장이 발생하였음을 실시간으로 알린다.

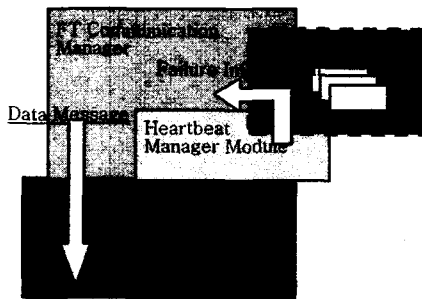


그림 4 공유메모리를 통한 FT communication manager 와 heartbeat manager의 시스템 정보 공유

고장 감내 기법 관리자는 다양한 고장 감내 기법들을 수행하게 한다. 대표적인 기법으로는 P/B, DRB, NVP 등이 있다[7]. 시스템 적용 관리자는 이런 기법들중 현재 상황에 가장 최적화된 기법을 적용하게 한다. 현재의 상황을 결정하기 위한 인자로는 사용가능한 리소스, 최근의 고장 정보, 환경적 정보, 클라이언트의 요구 정보 등이 있다[4].

그림 5는 P/B 기법을 적용한 가상서버의 동작 상태를 나타낸다. 클라이언트는 관리자 노드의 가상 IP를 사용해 서비스 요청을 한다. 관리자 노드는 고장 감내 기법중 P/B 기법을 적용해 primary 노드를 선택해 클라이언트가 요청한 프로그램을 실행하고 그 결과는 관리자 노드에게 반환된다. 마지막으로 관리자 노드는 결과값을 클라이언트에게 전송한다. 만약 primary 노드가 고장난 경우 관리자 노드는 backup 노드를 선택하여 같은 일을 수행한다. 만약 노드의 고장이 아니고 네트워크의 고장인 경우 primary 노드의 standby 네트워크 채널을 사용해 요청된 작업을 수행한다. 작업 노드들은 실시간 확장성을 갖는다. 작업 노드를 가상 서버에 추가시 작업 노드와 관리자 노드는 초기화를

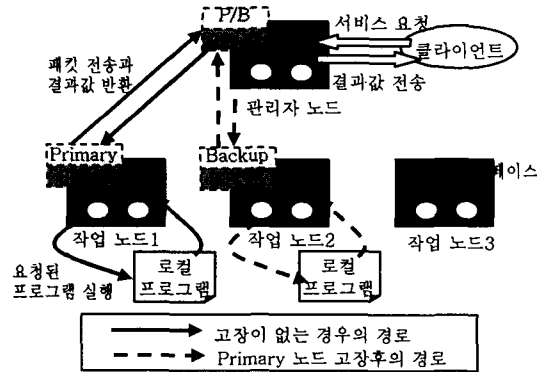


그림 5 PB 기법을 이용한 가상서버의 동작

위한 메시지를 주고 받음으로써 작업 노드를 작업 노드 풀에 등록하게 된다. 또한 사용자 응용 프로그램을 위한 RPC-like 함수를 사용하여 가상 서버의 특정 응용 프로그램을 실행시키기 위한 투명성(transparency)을 제공하였다.

#### 4. 결과 및 향후 연구

본 연구에서는 이중 채널 기반 가상 서버를 위한 분산 고장 감내 미들웨어를 구현하여 가상 서버의 신뢰성과 자원 효율성을 향상시켰다. 이전 연구[3]에서 수행한 고장 감내 미들웨어를 위한 기초적인 연구를 확장하여 서비스 제공을 위한 완전한 시스템 구축과 작업 노드의 확장성을 제공하였고, 기존의 가상 서버들이 갖지 못한 다양한 고장 감내 기법들을 적용한 적용형 고장 감내(Adaptive Fault Tolerance) 기법을 사용하여 신뢰성뿐만 아니라 시스템 자원의 효율성도 높였다. 앞으로의 연구는 채널을  $n$ 개로 확장시키는 문제와 미들웨어를 커널 계층에서 구현하였을때와의 성능 비교 평가, 가상서버의 효율적인 서비스 제공을 위한 클라이언트와의 프로토콜 개선 문제 등을 수행할 것이다.

#### 참고문헌

- [1] The Linux Virtual Server Project, <http://www.linuxvirtualserver.org>
- [2] Huang, J.; Song, S.; Li, L.; Kappler, P.; Freimark, R.; Gustin, J.; Kozlik, T., An open solution to fault-tolerant Ethernet: design, prototyping, and evaluation, Performance, Computing and Communications Conference, 1999 IEEE International, 1999 Page(s): 461 -468
- [3] 최보근;김진용;함명호;신현식, 이중 채널 이더넷을 이용한 분산 결함 허용 시스템, 한국정보과학회 추계 학술발표 논문집, 2002
- [4] Hecht, M.; Hecht, H.; Shokri, E., Adaptive fault tolerance for spacecraft, Aerospace Conference Proceedings, 2000 IEEE, Volume: 5, 2000 Page(s): 521 -533 vol.5
- [5] Jae Min Lee; Wook Hyun Kwon; Young Shin Kim; Hong-Ju Moon, Physical layer redundancy method for fault-tolerant networks, Factory Communication Systems, 2000. Proceedings, 2000 IEEE International Workshop on, 2000 Page(s): 157 -163
- [6] Sape Mullender, "Distributed Systems, 2nd Ed.", Addison-Wesley, 1993
- [7] Dhiraj K. Pradhan, "Fault-Tolerant Computer System Design", Prentice Hall PTR