

SNMP 와 Web 기반의 Load Cluster 관리 시스템의 설계 및 구현

김 명섭, 홍 원기
포항공과대학교 컴퓨터공학과
{mount, jwkhong}@postech.ac.kr

요 약

인터넷 서비스 사이트에 대한 늘어나는 사용자 접속을 수용하기 위한 방법의 하나로 Load Balancing 을 이용한 Cluster System 의 형태로 서비스 사이트를 구축할 수 있다. Clustered Computer 들은 Client 의 입장에서는 단일 host 와 같은 Single System Image (SSI)로 보여지고, Cluster group 에 쉽게 구할 수 있는 값싼 장비들을 계속 추가할 수 있는 확장성이 좋은 장점을 가지고 있다. 그렇지만 구축과 관리의 면에서 보면 여러 host 들이 각자가 맡은 역할을 수행함으로써 하나의 서비스를 형성하기 때문에 Cluster 를 구성하는 어느 한 host 에 장애가 발생하면 전체의 서비스에 장애로 이어지는 문제를 안고 있다. 이를 해결하기 위하여 여러 가지 High Availability (HA) solution 들이 개발되고 있다. HA 는 안정적인 서비스를 위하여 꼭 필요한 기능이지만 근본적으로 서비스 수행과는 분리된 부가적인 기능이다. 따라서 HA 는 서비스 수행에 최대한 부담을 주지 않는 방법으로 그 기능을 수행해야 한다. 또한 구축된 Cluster 환경을 효율적으로 관리하는 것 또한 중요한 문제가 된다. 본 논문에서는 효율적으로 Load Cluster 환경을 구성하고 관리하는 시스템을 제안한다. SNMP 를 이용하여 각각의 Real Sever 의 역할을 할당하고, 그들의 동작상태를 monitoring 함으로써 관리를 위한 Load Balancer 의 부담을 분산시키고 HA 를 위한 Network 의 사용량을 줄였고, Web 을 이용한 관리자 Interface 는 관리자가 쉽게 각각 host 의 서비스 상태를 관찰하고, 구성을 조절할 수 있도록 하였다. 또한 Load Cluster Group 과 각 host 들의 정보관리를 효과적으로 하기 위하여 LDAP 을 사용하였다.

1. 서론

일반 사용자가 쉽게 인터넷을 사용할 수 있는 빠른 네트워크 환경과 다양한 응용프로그램들이 마련되면서 인터넷을 사용하는 사람들은 폭발적인 증가를 해왔다. 그리고 이러한 증가의 추세는 앞으로도 계속 이어질 것이고, 이에 못지않게 사용자의 요구는 다양하게 증가될 것이며 이에 따른 인터넷 서비스의 발전도 지금까지 그래왔듯이 지속적으로 이어질 것이다. 인터넷 사용자의 증가와 서비스 사이트에 대한 접속 수 증가에 대한 대책으로 성능이 좋은 장비로 사이트를 구축하는 방법도 있겠지만 가격과 확장성의 면을 고려하여 Cluster 환경[1, 2, 10]으로 서비스 사이트를 구축하는 것이 활발히 진행되어가고 있다. 인터넷 서비스에 대한 Cluster 환경 구축은 Round-Robin DNS[2, 3, 4, 5], Dispatcher Approach[10, 11, 12], Parallel filtering[7], L4 Switch[6, 7]등의 다양한 방법으로 이루어지고 있다. Cluster 환경은 사용자의 입장에서는 Cluster 에 포함된 host group 이 하나의 Single System Image (SSI)로 보여지게 되고 사용자는 단일 host 에 접근하는 것과 똑같은 방식으로 Cluster system 에 접근하고, 그 서비스를 이용하게 된다. 이러한 Cluster 환경은 저렴한 가격의 host 들을 사용함으로써 가격적인 면이나 확장성의 장점으로 그 적용은 계속 증가 할 것이다.

Cluster 환경은 여러 host 들이 모여서 하나의 service 를 하기 때문에 안정적인 서비스 보장이 상당히 중요하다. 하나의 host 장애가 전체 cluster 의 장애가 되지 않고 서비스를 원활히 계속 할 수 있도록 하는 High Availability (HA)[8, 9]가 중요하다. HA 는 cluster 를 구성하는 단위 host 들의 집합에서 각각의 host 의 주어진 역할에 대한 수행상태를 모니터링하고 그 역할 수행에 문

제가 발생하였을 경우 적절히 대처함으로써 서비스를 원활히 수행해 나갈 수 있는 환경을 제공함을 뜻한다. 지금까지 여러 가지 방법의 HA 에 관한 연구 결과들이 발표되고 system 들이 개발되어 왔다. 그렇지만 지금까지의 HA 는 구축된 Load Cluster System 이 기능적인 측면은 배제한 채, 부가적인 역할인 monitoring 기능과 configuration control 기능만을 담당하여왔다. 즉 monitoring 된 각 node 의 정보가 작업 분배에는 전혀 고려가 되지 않았다. 또한 HA 를 위한 system 이 어느 특정 node 에서만 동작하여 그 node 의 부하를 증가시키는 경우가 많았고[7], 여러 node 들에 분배가 되었다 하더라도 manager 와 agent 간에 통신에 자체 개발된 프로토콜을 사용하여 확장성이나 이식성에 문제를 안고 있었다. HA 를 위한 system 은 각각의 node 들을 정기적으로 monitoring 해야 하기 때문에 그 과정에서 network 의 부하를 많이 소비하는 것 또한 개선의 여지를 많이 안고 있다. 또한 Load Cluster 를 구축하고, HA system 을 구축하는 작업들에 있어서 관리자의 수작업이 각각의 node 에서 이루어져야 하기 때문에 Load Cluster 를 구축하는데도 어려움이 있었고, 현재의 작업 분배 상태를 보여주거나, 각 node 의 load 를 관찰하는 것이 어려웠다.

본 논문에서는 Dispatcher 기반의 Load Cluster (LC) system 을 통합 관리할 수 있는 관리 시스템을 제안한다. 네트워크와 system 관리의 표준으로 자리잡고 있는 SNMP 를 이용함으로써 관리 네트워크와 system 관리 표준을 따르고, HA 를 위한 기능들을 그 역할에 따라서 여러 host 들에 분산함으로써 어느 한 host 에 지나치게 걸렸던 부하를 분산시키고, monitoring 을 위한 네트워크의 사용량을 줄이는 방법을 제시한다. 또한 HA system 의 관리 정보를 LC system 에서 사용하게 함으로써 좀 더 효율적인 LC system 의 사용에 대한 방법을 제시한다. 그리

고 Web 기반의 LC system 관리 방식을 제시함으로써 관리자가 LC system 을 구축하고 관리하는데 효율성과 편의성의 증대를 위한 방안을 제시한다.

논문의 구성은 다음과 같다. 제 2 장에서는 LC system 의 구축과 HA 를 위한 여러 가지 관련기술과 연구들을 알아보고 그들이 가지는 한계점과 본 논문에서 해결하려고 하는 부분에 대하여 짚어본다. 제 3 장에서는 LC 관리 시스템이 가져야 하는 요구 사항들을 제시한다. 제 4 장에서는 SNMP 와 Web 기반의 관리시스템의 구조를 설명한다. 제 5 장에서는 논문에서 제시된 시스템의 구현에 관하여 기술 한다. 마지막으로 제 6 장에서는 결론과 앞으로의 연구방향을 제시함으로써 본 논문을 마무리한다.

2. 관련연구

Cluster 의 구축과 관리는 Cluster 가 어떤 목적으로 구성이 되어 있는가에 따라서 다양한 방법으로 이루어질 수 있다. 이 장에서는 여러 가지 Cluster 환경에서 구축된 관리 시스템들을 살펴보고 그들의 장단점을 분석해 봄으로써 본 논문에서 해결하고자 하는 문제를 제시해 본다.

2.1 Linux HA

Linux Virtual Server (LVS) system[8, 9, 10, 11, 12]에 장애극복과 확장성을 제공하기 위한 연구로 여러 가지 다양한 고가용성(HA)을 위한 방법과 tool 들을 제공하고 있다. 그림 1 은 그 중 가장 대표적인 HA 방법을 나타낸 것이다.

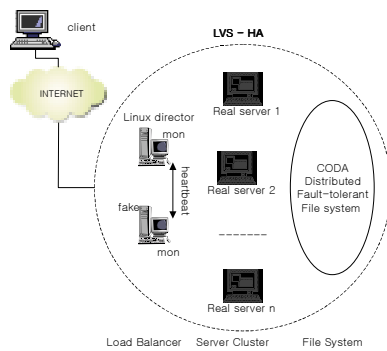


그림 1. Linux HA

전체적인 구조를 간단히 살펴보면, Real Server 를 monitoring 하는 MON daemon[13]과 Load Balancer 의 Backup Server 를 위하여 heartbeat protocol[9]과 fak[14]로 구성되어 있다. 그리고 file system 의 안정성을 위하여 CODA file system[15]을 제안하고 있다.

MON system[13]은 LVS 에서 Real Server 들의 상태와 서비스를 관찰하고 이들 중에 장애가 발생하였을 경우 Load Balancer 의 LVS 구성을 바꾸거나, 관리자에게 알람을 보내는 Real Server 의 관리자 역할을 한다. 이렇게 되면 Cluster 환경에서 Load Balancer 가 single point of failure 가 되는데 이를 극복하기 위하여 Load Balancer 의 Backup Server 를 두고 Backup Server 는 Load Balancer 의 동작 상태를 heartbeat protocol 을 이용하여 체크 하다가 장애가 생겼을 경우 역할을 대신하게 된다.

이 시스템의 경우 HA 를 위한 관리기능에만 중심을 두었고, 관리자의 Configuration 편의를 위한 고려는 전혀 없다. 그리고 MON 시스템은 그 목적이 일반 host 의 자원관리를 목적으로 개발된 것이기 때문에 Cluster 의 특수한 환경의 요구를 모두 수용하지는 못한다. 또한 Real Server 를 외부에서만 관리하기 때문에 Real Server 의 정확한 상태와 부하를 측정하지 못하고, Real Server 의 수가 많을 경우에 관리를 위한 network 사용이 커지게 된다. MON system 은 Load Balancer 에 동작하기 때문에 Load Balancer 에 과부하를 일으키는 요인이 될 수 있다.

2.2 Cluster Environment Observer (CEO)

Cluster 환경은 UNIX 나 Windows 등 다양한 platform 에서 구축 되는데 CEO[16]에서는 이들의 서로 다른 platform 에 구축된 cluster node 들을 단일한 interface 를 통하여 관리할 수 있는 환경을 제공한다. CEO에서는 Cluster 내의 host 들을 모니터링하는 system 으로 PARMON[17, 18]을 구축하였다. PARMON 은 High Performance Cluster 에서 각 단위 node 들을 monitoring 하는 tool 로서 client/server 의 구성을 갖는다. PARMON server 는 monitoring 되어야 하는 모든 node 에 설치가 되고 PARMON client 에서 Message Passing 방법을 이용하여 cluster node 의 cpu usage, memory utilization 등을 monitoring 하게 된다. 이 시스템은 monitoring 을 위한 통신방법으로 자체 개발한 것을 사용하고 있다. JAVA application 으로 구성이 되어 있어 platform 독립성, 이식성이 좋은 장점은 가지고 있으나, PARMON client 가 설치된 host 에서만 관리가 가능하다는 단점이 있다.

2.3 JobCoNTrol

JobCoNTrol[19]은 NT cluster 를 관리하는 system 으로 관리자는 remote 로 cluster 내의 각 node 의 구성요소를 check 하고, 현재의 CPU 및 Memory 사용량을 monitoring 할 수 있으며 remote 로 프로그램을 실행할 수 있는 기능을 제공한다. 각각의 NT node 에는 manager 와의 통신과 node control 을 위한 agent 가 동작하며 이들을 관리하는 manager 는 Windows application 으로 작성이 되어 있다. 이 시스템은 manager 와 agent 사이의 통신을 관리목적에 맞게 최적화된 자체 프로토콜을 사용한 장점은 있지만, agent control 을 manager program 을 통해서만 해야 하기 때문에 관리 방법에 있어서 이동성, 이식성에 많은 제약이 있다.

2.4 NCSA Symera

Symera[20]는 Microsoft 의 Distributed Component Object Model (DCOM)[21]기반의 NT 기반의 Cluster node 관리 system 이다. 이 시스템은 각 node 의 resource 관리, job scheduling, fault tolerant 기능과 manager 와 통신할 수 있는 library 를 제공함으로써 개발자가 새로운 management object 를 만들어 시스템에 통합할 수 있는 방법을 제공한다. 이 시스템은 manager 와 agent 간에 DCOM 이라는 표준 통신 방법을 사용하였고, manager 시스템에 대한 확장성과 유연성을 제공한다는 면에서 큰 장점이 있다. 그렇지만 DCOM 의 사용은 시스템의 동작 플랫폼을 Windows OS 에 한정시켜 이식성이 떨어뜨

리는 문제를 안고 있다.

2.5 Pilot Project

Pilot Project[22]는 핀란드에서 개발된 과학 계산 응용프로그램 수행을 위하여 PC 기반으로 만들어진 high performance Cluster 를 관리하기 위한 Cluster Management System (CMS)을 제공한다. 이 CMS 는 각 Cluster 상의 node 들의 memory 사용량, cpu load 를 rsh, rup 등의 system 명령어를 통하여 알아내고, 그 정보를 기반으로 Job Queue 에 있는 program 수행을 위한 node 를 결정하게 된다. 사용자의 편의를 위하여 GUI 방식의 interface 를 제공하고 있다.

표 1 은 각 시스템의 특성을 정리한 것이다. 관리 시스템은 주로 High Performance cluster (HPC)에 대해서 주로 만들어져 있고, Load Balance Cluster (LBC)에 대해서는 많은 연구가 없다. 또한 관리 구조가 중앙 집중식, 분산식 등의 다양한 방법으로 이루지고 있다. 관리대상과 manager 사이의 통신방법은 주로 표준 protocol 을 따르지 않고 자체 개발한 protocol 을 사용한다.

표 1. Cluster 관리 시스템 비교

	Linux HA	CEO	JobCoNTrol	NCSA Symera	Pilot Project
Management Type	Centralized	Distributed	Distributed	Distributed	Centralized
Communication Method	Proprietary	Proprietary	Proprietary	DCOM	rsh, rup
Manager Interface	CLI	GUI	GUI	GUI	GUI
Managed target	Linux	UNIX Windows	Windows NT	Windows	Linux
Cluster Type	LBC	HPC	HPC	HPC	HPC
Cluster Configuration	X	X	X	X	X
Fault Tolerant	O	X	X	O	X
Resource Management	X	O	O	O	O
Portability	X	O	X	X	O

LBC 에 대해서는 HPC 만큼의 관리 시스템이 개발되어 있지 않다. 또한 관리자 Interface 도 역시 관리자가 직접 각각의 node 에 접속하여 설정을 해 줘야 하고, 동작의 상태를 보여주는 기능 또한 부족하다. 본 논문에서는 Load Cluster 를 구성 및 HA 를 통합 관리할 수 있는 system 을 제안한다.

3. Load Cluster 관리 시스템의 요구사항

본 논문에서는 Load Cluster system 의 여러 가지 방법들 중에서 Dispatcher Approach 에 대한 관리 시스템을 제시한다. Dispatcher Approach 는 Cluster Group 중 하나의 host 가 Load Balancer 의 역할을 하는 것으로 client 로부터의 모든 request 들을 받아서 미리 정해진 load balancing algorithm 에 의해 Real Server 들에게 분배하는 역할을 한다. Dispatcher 방식은 그림 1 과 같이 Load Balancer, Real Server 그리고 HA[8, 9]를 위한 Backup Server 로 구성된다.

Real Server 는 Client 로부터의 Request 를 받아서 실제 서비스를 수행하는 역할을 하고, Backup Server 는 Load Balancer 에서 기능 장애가 생겼을 경우 그 역할을

대체하는 기능을 수행한다. 이러한 Dispatcher 형태의 Load Cluster 를 효과적으로 관리하기 위해 Load Cluster Management System (LCMS)이 가져야 하는 기능들을 살펴보자.

3.1 LC 구성을 위한 관리 기능

이 기능들은 Load Cluster 의 구성에 관한 부분으로 LCMS 는 LC group 에 들어갈 수 있는 여러 host 들의 정보를 관리하고 이들 host 들의 조합으로 Load Cluster 를 구성하고 그 구성정보를 저장하며, 구성을 해지하는 역할을 수행해야 한다. 즉 LC group 에 소속된 host 들에게 Load Balancer (LB), Backup Server(BS), Real Server(RS)의 역할을 부여할 수 있는 관리자 Interface 가 제공되어야 하며, 각각의 host 의 정보를 저장하기 위한 정보저장고 그 정보를 관리자 임의로 살펴보고, 변경할 수 있는 기능이 있어야 한다. 또한 host group 이 여러 LC 를 구축할 수 있도록 구축된 LC group 들의 정보를 효과적으로 저장할 수 있는 방법이 마련되어야 한다. 또한 각각의 host 들에 역할이 주어졌을 경우 그 역할에 맞게 자동으로 기능을 시작 할 수 있는 방법이 마련되어야 하고, manager 와 host 사이에 효율적인 통신 방법도 마련되어야 한다. 또한 관리자의 인증을 위한 방법도 마련되어야 할 것이다.

3.2 HA 를 위한 관리 기능

이 부분은 위의 LC 구성 관리에 의해 구성된 Load Cluster 의 동작을 체크하고, 각 host 들의 동작 상태에 대한 log 기록, 허가 받지않은 client 의 접속여부 감시 등의 단순한 monitoring 기능에서 Cluster 내의 host 에서 장애가 발생했을 경우 이루어져야 하는 control 기능까지 폭 넓은 장애관리 기능을 담당한다. 이는 Load Cluster 를 구성하는 Real Server 들에 대한 동작 관리, 그리고 Load Balancer 의 동작관리의 두 부분으로 나누어 볼 수 있다. 또한 HA 의 기능은 Load Cluster 의 주 기능에 부담을 주지 않는 방법으로 이루어 져야 한다.

3.3 관리 효율성을 위한 관리 기능

이 기능은 관리자의 편의를 위해 LCMS 가 가져야 하는 기능들이다. 관리자 Interface 가 GUI 형태로 제공되어야 할 것이며, 관리자가 LCMS 에 접근하여 현재의 LC group 의 구성상태와 LC group 내 각 node 들의 동작 상태를 monitoring 하고 control 할 수 있는 쉬운 방법이 제공되어야 한다. 현재까지 제시된 Load Cluster 에 대한 관리 기능들은 Load Cluster 의 구성, 동작 monitoring, HA system 의 구성들에 대한 작업들이 관리자가 직접 그 host 에 접근하여 작업을 해야만 했다. Cluster group 내의 host 들이 증가할수록 이런 관리의 편의성을 위한 기능들에 대한 요구는 더욱 증가할 것이다. 언제 어디서나 관리자는 Internet 의 연결만을 통해서 Load Cluster 에 대한 관리를 할 수 있어야 할 것이다.

4. 관리 시스템의 설계

본 논문에서 제안하는 Load Cluster 관리 시스템의 구조는 아래의 그림 2 와 같다. 시스템 설계는 3 장에서 제시한 관리 시스템의 요구사항을 최대한 반영하고, 구현의 용이성과 다른 형태의 Load Cluster System 으로의

이식성을 고려하여 이루어졌다. 제시하는 시스템은 SNMP[23]를 기반의 관리 시스템으로 관리자 Interface 로 JAVA 와 Web 을 이용하여 관리의 편의성을 도모하였다.

관리 시스템의 구조는 크게 세 타입으로 Load cluster 관리 구조, Real Server 관리 구조, Load Balancer 관리 구조로 구성된다. LC 관리 구조는 가용한 host 들의 정보를 바탕으로 Load Cluster 를 구성/재구성 및 현재 구성된 Load Cluster 에 대한 정보관리 등의 역할을 담당하는 부분이다. 이를 위하여 Cluster group 에 참여할 수 있는 host 들에 대한 정보 관리를 위해 host Information Repository 를 두어 Load Cluster 를 구성할 수 있는 host 들의 정보와 구성된 Load Cluster Group 의 정보를 관리한다. LC Manager 는 HOST Information Repository 의 data 들을 관리자가 변경할 수 있는 API 를 제공한다. 이 API 는 관리자가 Web Browser 를 통하여 제어할 수 있도록 CGI 의 형태로 제공된다. 따라서 관리자는 Web Browser 를 통하여 현재의 Load Cluster 구축 상태를 파악할 수 있고, 가용한 host 들을 선택하여 하나 이상의 Load Cluster Group 을 구축할 수 있게 된다.

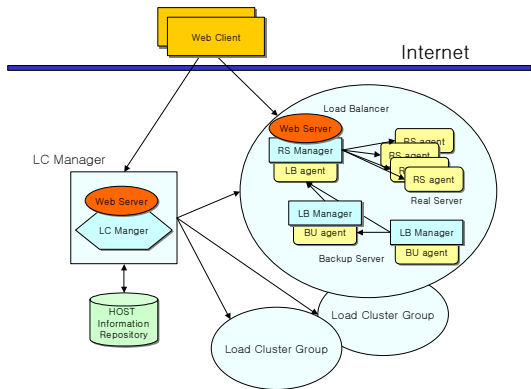


그림 2. LCMS architecture

또한 LC Manager 는 Load Cluster Group 을 형성하기 위하여 각각의 host 들에게 역할을 부여하는 기능을 수행한다. 각 Host 들은 그 부여된 역할에 따라서 하나의 SNMP agent 가 LB agent, BS agent, RS agent 의 다른 역할을 수행하게 된다. 즉 Load Balancer 의 역할이 부여된 host 의 SNMP Agent 는 LB agent 로서의 역할을 수행하게 되고, 그 host 가 Load Balancer 의 역할을 수행할 수 있도록 system 을 구성하고 LC manager 에게 Load Balancer 로써 필요한 정보들을 제공하게 된다. 또한 host 가 Real Server 의 역할을 부여 받았을 경우에는 RS agent 가 그 host 에 동작하여 Real Server 로써 host 를 구성하고, RS agent 는 LC Manager 와 뒤에서 설명할 RS Manager 에게 필요한 정보를 제공하는 역할을 수행한다. Backup Server 를 구성하기 위해 BS agent 가 동작하여 Load Balancer 에 대한 backup server 의 역할을 수행한다. Backup Server 는 primary backup server, secondary backup server 등 cascade 형태로 여러 host 를 사용할 수 있다. 그리고 Real Server 들에 대한 관리는 RS Manager 에 의해 이루어지는데 Real Server 에서 동작하는 RS agent 를 통해서 수행된다.

RS 관리 구조는 Load Balancer 에서 동작하는 RS Manger 와 Real Server 에서 동작하는 RS agent 로 구성된다. 이 관리 구조는 Client 로부터의 request 에 대한 각

Real Server 로의 분배가 각 Real Server 에서 잘 처리가 되고 있는지를 monitoring 하고, 어떤 Real Server 에 장애가 발생했을 경우 Load Balancer 에서 그 Real Server 로 request packet 을 forwarding 하지 않도록 Configuration 을 바꾸는 역할을 한다. 또한 각 Real Server 의 현재 load 를 감지하여 Load Balancer 의 Scheduling Algorithm 을 제어하는 역할을 수행한다.

LB 관리구조는 LB Manager 에 의한 Load Balancer 의 동작을 monitoring 하고, Load Balancer 에 장애가 생겼을 경우 Load Balancer 의 역할을 대신 수행하고, 다시 Load Balancer 의 기능이 복귀되었을 경우 그 기능을 Load Balancer 에게 돌려주는 역할을 수행한다. 이 구조는 하나의 Load Balancer 를 관리하기 위하여 Primary, Secondary Backup Server 를 둘 수 있는데, Secondary Backup Server 는 Load Balancer 와 Primary Backup Server 를 동시에 관리하게 된다.

관리 구조를 세 가지 타입으로 구분한 것은 관리 내용과 관리 대상을 기준으로 만들어졌다. 이러한 분리는 각각의 관리자를 서로 다른 host 에 분산시킬 수 있는 장점을 가지고 있다. 다음은 각각의 관리구조에 대하여 구체적인 사항들을 살펴본다.

4.1 Load Cluster 관리 구조

아래의 그림 3 에서 보는 바와 같이 LC 관리 구조는 LC Manager, Cluster Information DB, 그리고 각각의 역할에 맞는 관리대상 Host Agent 로 구성된다.

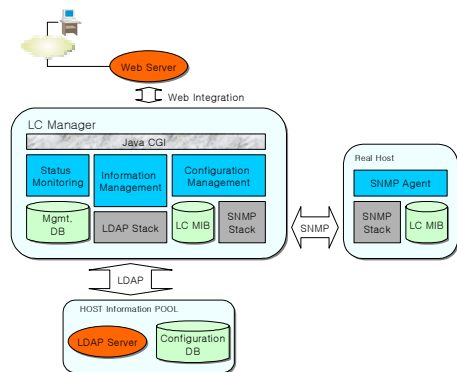


그림 3. Load Cluster 관리 구조

Cluster Information DB 는 LDAP[24]을 이용하여 정보들을 저장한다. 이는 LDAP 이 directory 구조로써 Cluster 구축이 여러 개의 Cluster 들이 peer 를 형성하고, 하나의 Cluster 에 LB, BS 가 있으며, LB 의 관리 대상으로 RS 가 존재하는 구조로 일반 Table 형식의 DB 보다는 LDAP 의 directory hierarchical 한 구조가 더 적합하기 때문이다. 또한 관리자 인증을 위한 보안 기능을 사용하기에 적합한 Database 구조이다. Load Cluster 를 위하여 LDAP 에서 저장하는 정보의 구조는 아래의 표 2 와 같다. 저장되는 정보는 일반 host 들에 대한 정보. Load Cluster 구성에 관한 정보 그리고 관리자에 관한 정보의 세가지로 나뉘어진다. 각각의 정보는 LDAP[24]에서 Tree 의 형태로 구성될 것이다.

LC Manager 는 관리자에게 Load Cluster 관리를 위한 API 제공을 위해 Web 과 JAVA 를 이용한다. Web 은 기존의 표준화된 HTTP protocol 을 이용함으로써 개발에

있어서 표준과 용이성을 부여할 수 있고, 관리자가 인터넷에 연결된 어느 컴퓨터에서나 관리를 할 수 있는 관리자 편의성을 부여할 수 있다.

표 2. Load Cluster 관리 정보

HOST Info.	Load Cluster Info.	Manager Info.
Host name	Load Cluster ID	Name
CPU type	Load Balancer IP	ID
Memory size	Backup Server IP's	Password
NIC type	Real Server IP's	
IP address	Scheduling Algorithm	
Weight	Service type	
Role status		
Role potential		

또한 HTTPS 나 SSL 과 같은 공신력 있는 network 보안방법을 사용할 수 있어 관리자 API 로 근래 많이 사용하고 있는 방법이다. JAVA 는 이식성과 개발의 용이성의 측면에서 볼 때 최상의 선택이다. 또한 LC Manager 는 Cluster Information DB 관리하는 모듈, 각각의 관리대상 agent 에 역할을 부여하고, 제어하는 module, 그리고 각 agent 의 상태를 monitoring 하는 module 이 있다. 관리자가 LC Manager 에 대하여 할 수 있는 기능은 LC Information 관리, LC configuration/reconfiguration, LC monitoring 의 세가지 기능으로 나누어 볼 수 있다.

관리대상 agent 는 각각의 역할에 맞게 host 의 상태를 관리하는 역할을 한다. LC Manger 와 agent 사이는 SNMP 를 이용하여 관리를 한다. SNMP 의 Manager 와 Agent 의 관리구조를 택함으로써 Manager 와 Agent 사이의 communication 과 관리정보의 작성에 대한 표준을 정할 수 있다. 이를 위하여 Load Cluster Management System MIB (LCMS-MIB)을 정의하였다.

```

Table entry of Load Balancer MIB
RealServerEntry ::= SEQUENCE {
    rsIndex          INTEGER,
    rsIpAddress     IpAddress,
    rsWeight        INTEGER,
    rsStatus        INTEGER
}
    
```

LCMS-MIB 은 host 의 역할을 구분하는 type 을 위한 type field 가 있고 이 field 의 값에 따라 그 역할이 결정되게 된다. 위의 표는 host 가 Load Balancer 로 동작할 경우 관리할 Real Server 들에 대한 정보를 table 의 형태로 나타낸 MIB 이다.

4.2 Real Server 관리 구조

기존의 많은 Load Cluster RS 관리 구조는 Load Balancer 에 관리 daemon 이 관리에 대한 모든 기능을 수행하였기 때문에 Load Balancer 에 부담을 많이 주었고, 또 관리를 위해 사용하는 network bandwidth 도 많고 그 관리 정보도 단순한 service availability check 정도였다. Dispatcher 방식의 Load Clustering 은 모든 Request 가 Load Balancer 로 들어가기 때문에 Real Server 보다는 LB 에 performance bottleneck 이 된다. 본 논문에서 제시하는 관리구조는 SNMP 기반의 Manager/Agent 의 관리구조를 택함으로써 Load Balancer 에 부담을 줄일 수 있다. 아래의 그림 4 는 RS 관리구조를 나타낸 것이다.

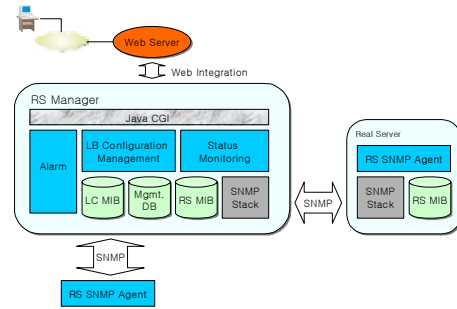


그림 4. Real Server 관리 구조

RS Manager 는 LC Manager 에 의해 구성된 Load Cluster 내의 Real Server 들을 관리한다. 관리의 내용은 Real Server 의 상태와 등록된 서비스의 동작여부를 주기적으로 check 하는 기능, monitoring 에 대한 logging 기능, 관리자에게 현재의 상태와 과거의 상태 보고 하는 기능, 그리고 Real Server 가 장애가 생겼을 경우 Load Cluster Group 에서 Real Server 를 제거하고, 다시 그 서버가 재동작할 경우에 Load Cluster Group 에 자동 등록하는 기능을 수행한다. 관리자와의 communication 을 위하여 JAVA 와 Web 을 이용한 CGI 를 제공하며, 이 CGI 는 RS core 와의 SNMP 통신으로 관리자에게 필요한 정보를 제공한다.

RS agent 는 제공하는 service 의 현재 상태와 관리 정보를 RS manager 에게 reporting 하는 기능을 수행한다. 다음은 RS manager 의 입장에서 RS 를 관리하기 위해 RS agent 가 제공해야 할 MIB 정보들 중 제공하는 Service 들에 대한 MIB table 을 나타낸 것이다.

```

Table entry of Load Balancer MIB
ServiceEntry ::= SEQUENCE {
    serviceIndex      INTEGER,
    serviceName      DisplayString,
    servicePort       INTEGER,
    serviceStatus     INTEGER,
    serviceCurrnetConnection INTEGER
}
    
```

이러한 SNMP 기반의 RS 관리 구조는 관리 기능을 manager 와 agent 로 구분하여 하나의 RS 관리구조가 가지는 역할을 분산함으로써 Load Balancer 로의 부담을 줄여줄 수 있다. 또한 SNMP protocol 의 사용은 기존의 방법보다 network bandwidth 소비를 줄이고, monitoring 속도를 향상시킬 수 있다. 또한 각각의 Real Server 에 Agent 를 두어 현재의 service connection 수, network bandwidth usage, CPU usage 등의 정보를 얻을 수 있으므로 해서 Real Server 에 대한 구체적인 정보를 얻을 수 있다.

4.3 Load Balancer 관리 구조

LB 관리 구조는 Load Balancer 의 역할 수행을 monitoring 하고 Load Balancer 에 장애가 발생했을 경우 자신이 Load Balancer 의 역할을 대신할 수 있도록 설정하는 기능을 수행한다. LB manager 는 backup server 에서 동작하고, primary backup server 는 Load balancer 의 동작을 secondary backup server 는 primary backup server 와

secondary backup server 의 동작을 모두 관찰하는 역할을 수행한다.

5. 구현

Dispatcher 방법의 Load Cluster 를 구성하기 위하여 Linux 환경의 Linux Virtual Server (LVS) [10, 11, 12]를 이용하였다. LVS 에서 제공하는 3 가지 LVS 구성 방법 중 LVS-DR[12]방법을 이용하여 구성하였으며 load balancer 의 scheduling algorithm 은 Round Robin (RR), Weighted RR (WRR), Least Connection (LC), Weighted LC (WLC)[12]의 방법을 이용하였다. 그리고 Virtual Server 에서 제공하는 서비스로 인터넷 스트리밍 서비스를 LVS 상에 구축하여 제공하였다. 구축된 스트리밍 서비스를 위한 LVS 시스템의 구조는 그림 5 와 같다.

Streaming Service 방법은 Web Server 를 이용한 방법과 Streaming Server Application 을 이용한 방법의 두 가지가 있는데, LVS System 은 두 가지의 방법으로 모두 구현 되었다. 또한 지원하는 format 또한 현재 인터넷 환경에서 많이 사용되고 있는 Windows media format[26], Real media format[27], Quicktime movie format[28]을 모두 지원하도록 하였다. 또한 Client 역시 Web browser 를 이용하는 방법과 직접 player 를 이용하는 두 가지 방법 모두를 지원하도록 하였다.

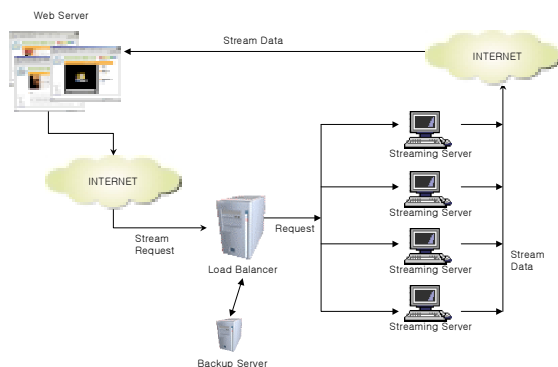


그림 5. Streaming Service 를 위한 LVS

Load Cluster Management system 의 구현은 다음과 같다. 우선 host 의 SNMP agent 로 NET SNMP[29]를 이용하였으며, 관리를 위해 추가해야 할 MIB 의 구성은 UCD-SNMP extension 을 이용하였으며 back-end processor 는 C/C++로 구현을 하였다. LC Manager 와 RC Manager 는 JAVA application 을 이용하였으며, Manager 와 SNMP agent 와의 communication 을 위하여 Advenet Net[30]에서 개발한 JAVA class API 를 이용하였다. LDAP 을 위해서 openLDAP 1.2[25]을 이용하였다. 관리자 와 LC/RC Manager 의 communication 을 위하여 Java script, applet, JSP, Servlet 을 사용하였으며 JSP engine 으로 tomcat[31]을, web server 로 Apache Web Server 를 사용하여 구현하고 있다. Load Balancing 을 위한 LVS system 은 구현이 완료된 상태이고, Load Cluster Management System 은 구현 중에 있다.

6. 결론 및 향후 과제

인터넷에 대한 서비스가 증가하고, 인터넷 사용자의 수가 증가함에 따라서 기존의 단일 서버의 용량을

넘어서는 처리를 Cluster 환경을 구축함으로써 보다 많은 수의 client 처리를 수용할 수 있게 하고 있다. 이렇게 여러 대의 host 들이 모여서 만들어진 Cluster 환경은 어느 한 host 가 장애가 발생했을 경우 cluster 의 서비스에 영향을 미치지 않도록 하는 HA 기능 또한 상당히 중요하다. 본 논문에서는 Cluster 환경의 구축과 HA 를 위한 SNMP 와 Web 기반의 관리 시스템을 제안 하였다. 이 시스템은 기존의 HA 시스템이 가지는 LB 로의 과부하현상을 LB 와 RS 들에게 분산함으로써 LB 의 부담을 줄이고, network 의 bandwidth 부하도 역시 줄일 수 있다. 또한 RS 의 서비스 정보와 system load 를 좀 더 정확하게 얻을 수 있으므로 해서 관리에 도움을 줄 수 있다. 또한 관리자가 인터넷 access 를 통한 관리를 할 수 있는 환경을 제공함으로써 관리의 편의성을 제공하였다. LC 관리구조, RS 관리구조, LB 관리구조의 분리는 기능별로 관리 시스템을 3분화 함으로써 다수의 Load Cluster 를 관리할 수 있는 환경을 마련하였다.

본 논문의 system 은 설계 후 구현단계에 있다. 제시된 관리 구조는 Dispatcher 방식을 Load Cluster 에 대해서만 고려를 하고 있다. 또한 구현 역시 LVS-DR 방식에 대해서만 고려한다. 이러한 관리 구조를 Round-Robin DNS 나 parallel filtering 방식의 Load Cluster 에 대하여 적용하는 방법도 연구가 되어야 할 것이다. 또한 RS agent 로부터 monitoring 된 RS load 에 대한 실시간 정보들을 Load Balancer 의 request scheduling 에 적용하는 방법에 대하여도 지속적으로 연구되어야 할 것이다.

[참고 문헌]

- [1] Jonathan Cragle, "Load Balancing Web Servers", *Windows2000 Magazine*, June 1998.
- [2] T. Brisco, "Dns support for load balancing", RFC 1794, April 1995, <http://www.ietf.org/rfc/rfc1794.txt>.
- [3] T. T. Kwan, R. E. McGrath, and D. A. Reed, "Ncsa's world wide web server: Design and performance", *IEEE Computer*, November 1995, vol28, no 11, pp.68-74.
- [4] E. D. Katz, E. D. Katz, R. McGrath, "A scalable http server: The ncsa prototype", *Computer Networks and ISDN Systems*, Nov. 1994, vol. 27, no 2, pp.155-163.
- [5] CISCO Systems, "Cisco IOS Server Load Balancing and the Catalyst 6000 Family of Switches", 1999, http://cisco.com/warp/public/cc/pd/si/casi/ca6000/tech/ios6k_wp.pdf.
- [6] Foundry Networks, "Cutting Through Layer 4 Hype", white paper, http://www.foundrynet.com/whitepaper_layer4.html.
- [7] Mohit Aron, Darren Sanders, Peter Druschel, Willy Zwaenepoel, "Scalable Content-aware Request Distribution in Cluster-based Network Servers", *USENIX Annual Technical Conference*, June 2000.
- [8] Tewari R, Dias D, Mukherjee R, Vin HM "High Availability for Clustered Multimedia Servers", *Proceedings of International Conference on Data Engineering*, February 1996, IEEE Press, pp.345-354.
- [9] A. Robertson, "High-Availability Linux Project", May 1998 - now, <http://www.linux-ha.org/>.
- [10] Wensong Zhang, "Linux Virtual Server Project", May 1998 - now, <http://www.linuxvirtualserver.org>.
- [11] Wensong Zhang, "Linux Virtual Server for Saclable

Network Services", *Ottawa Linux Symposium 2000*, July 2000.

- [12] Joseph Mack, "LVS-HOWTO", April 2001, <http://www.linuxvirtualserver.org/Joseph.Mack/HOWTO/LVS-HOWTO.html>.
- [13] Jim Trocki, "Service Monitoring Daemon - MON", Transmeta Corporation, September 1997 - now, <http://www.kernel.org/software/mon/>.
- [14] S. Horman, "Creating redundant linux servers", *The 4th Annual LinuxExpo Conference*, May 1998, <http://vergenet.net/linux/fake>.
- [15] CMU CODA team, "The coda project", 1987 - now, <http://www.coda.cs.cmu.edu/>.
- [16] Rajkumar Buyya, Toni Cortes, Oriol Teixio, "Cluster Environment Observer", Monash University, Australia, <http://www.csse.monash.edu.au/~rajkumar/ClusterObserver/>.
- [17] Rajkumar Buyya, Krishna Mohan, Bindu Gopal, "PARMON: A Comprehensive Cluster Monitoring System", *Proceedings of the International Conference on High Performance Computing on Hewlett-Packard Systems*, ETH Zurich, Switzerland, 1998.
- [18] Rajkumar Buyya, Krishna Mohan, Bindu Gopal, "PARMON: A Portable and Scalable Monitoring System for Clusters", *International Journal on Software: Practice & Experience (SPE)*, John Wiley & Sons, USA, June 2000.
- [19] Markus Fischer, "A cluster management software for Windows 9X, NT, 2000", 1999, <http://mufasa.informatik.uni-mannheim.de/lra/persons/markus/jobcontrol.htm>.
- [20] The Symera Team, "NCSA Symera", 1997-1998, <http://symera.ncsa.uiuc.edu/>.
- [21] Microsoft, "Distributed Component Object Model - DCOM", <http://www.microsoft.com/com/tech/DCOM.asp>.
- [22] Center for Science Computing, "PILOT PROJECT FOR PC CLUSTERS", 1999-2000, <http://www.csc.fi/metacomputer/pckluster/plans/aboaproject.html>.
- [23] William Stallings, "SNMP, SNMPv2, and RMON 1 and 2", Addison Wesley, 1999, pp163-203.
- [24] W. Yeong, T. Howes, S. Kille, "Lightweight Directory Access Protocol", IETF RFC1777, March 1995, <http://www.umich.edu/~dirsvcs/ldap/doc/rfc/rfc1777.txt>.
- [25] Kurt Zeilenga, "Open LDAP Project", OpenLDAP Foundation, <http://www.openldap.org>.
- [26] Microsoft, "Microsoft Windows Media", <http://www.microsoft.com/windows/windowsmedia/en/default.asp>.
- [27] Real Networks, "Real media technology", <http://www.realnworks.com/>.
- [28] Apple, "QuickTime", <http://www.apple.com/quicktime/>.
- [29] Net SNMP Project Team, "Net-SNMP", Sourceforge, <http://net-snmp.sourceforge.net/>.
- [30] AdventNet, "AdventNet SNMP API 3.1", <http://www.adventnet.com/products/snmpbeans/index.html>.
- [31] Jakarta Project, "Tomcat", <http://jakarta.apache.org/tomcat/index.html>.