

Roll-Forward Recovery 를 적용한 고가용 웹서버 구현

김용희⁰ 이재규 박희상 송대기 이철훈
충남대학교 컴퓨터공학과
{yonghee, jklee, hspark, dksong, chlee}@ce.cnu.ac.kr

Implementation of High Available Web-Servers using Roll-Forward Recovery

Yong-Hee Kim⁰, Jae-Gyu Lee, Hi-Sang Park, Dae-Ki Song, and Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National Univ.

요 약

본 논문은 Recovery 기법 중의 하나인 Roll-Forward 를 적용한 고가용 웹 서버의 설계 및 구현에 대해서 그 방안을 제시한다. Roll-Forward Recovery 는 복구에 드는 시간을 감소시킬 수 있다는 장점이 있다. 즉, 일반적인 중복 시스템에서 결함이 발생하면 시스템의 상태는 가장 최근의 checkpoint 지점으로 Rollback 되어야 하기 때문에 복구에 많은 시간을 소비하게 되지만, Roll-Forward 를 적용함으로써 복구에 드는 시간을 감소시킬 수 있다. 단일 서버시스템 또는 Rollback Recovery 를 적용한 시스템의 결함발생시 서비스를 제공할 수 없는 상태가 되거나 복구에 많은 시간을 소비한다. 이에 비해 Roll-Forward 를 적용한 중복 서버시스템을 구성하여 동작 운용중인 서버의 결함발생시 즉시 다른 서버가 결과를 보여줌으로써 고가용 웹 서비스를 제공할 수 있다. 본 논문에서 제시하는 Roll-Forward 를 적용한 고가용 웹 서버의 동작원리는 마스터서버에 들어오는 요청을 백업서버에서도 똑같이 수행하며, 마스터 서버가 동작 중 결함이 발생하였을 경우 백업서버가 그 결과를 보여줌으로써 고가용의 서비스가 제공되는 웹 서버를 구현하고자 한다.

1. 서 론

오늘날 인터넷 사용의 일반화로 그 응용범위가 확대되고 있는 추세이다. 특히, IT 환경에서는 60%이상의 서버시스템들이 Mission Critical 또는 Business Critical 한 작업에 투입되고 있으며, 네트워크 망을 통한 제어시스템의 활용이 날로 증대되고 있다. 이러한 시스템에서는 웹서버의 안정성 및 가용성 문제가 가장 중요한 문제로 지적되고 있다. 기존의 고가·고성능 컴퓨터를 기반으로 한 단일 서버방식에 의해서는 고비용 및 유지 보수의 어려움과 함께 고가용성 및 호환성 부족 등의 문제점 등이 있다.

컴퓨터 시스템의 구조가 사용자의 편리성과 고성능의 처리능력에 대한 욕구를 충족시키고자 점차로 복잡해지고 다양해짐에 따라 컴퓨터 시스템의 동작에 결함이 발생할 가능성도 더욱 커졌을 뿐만 아니라, 결함이 발생하였을 경우 신속한 조치에 대한 사용자의 기대 또한 커짐에 따라 짧은 시간 내에 시스템의 결함 부위를 찾아내고 필요한 조치를 취하기 위한 별도의 기능이 강력히 요구되어지고 있다. 시스템이 요구하는 고가용성을 부가하기 위하여서는 결함의 발생이 확인되는 즉시 시스템의 정지가 없도록 빠른 복구를 할 수 있는 기법을 개발하는 것이 매우 필요하다.

이에 따라 본 논문에서는 리눅스를 기반으로 하는 고가용성 및 고신뢰도의 보장을 위한 Heartbeat 및 Fake 기법과 Rollback 과는 달리 시스템의 복구시간이 적게 소요되는 Roll-Forward Recovery 를 적용하여 고가용 웹서버를 구현하고자 한다. 무정지, 고가용의 웹서버를 만들기 위한 솔루션으로써 가상 IP 를 사용하여 외부 사용자에게는 하나의 IP 만을 보여

* 본 연구는 BK21 충남대학교 정보통신인력양성사업단의 지원을 받았음.

주게 된다. 내부적으로는 두개의 서버가 Heartbeat 를 통하여 생존여부를 확인하면서 가상 IP 를 통하여 사용자의 요청이 들어오면 두 개의 서버가 모두 가상 IP 를 인식하여 요청을 처리하며, 하나의 서버에서 발생할 수 있는 결함으로 인한 최악의 상태에 대비하고자 하는 데 그 목적이 있다고 하겠다.

본 논문은 2 장에서는 고가용 웹 서버를 구현하기 위해 필요한 요소로 Linux 에서 제공되는 Open Source 인 Heartbeat 과 Fake, 그리고 서버동작방식인 Roll-Forward 및 Rollback 에 대한 관련 연구를 기술한다. 3 장에서는 Roll-Forward 를 적용한 웹서버 구성과 동작원리에 대해 설명하고, 4 장에서는 실험환경과 결과 및 문제점들에 대해 기술한다. 마지막으로 5 장에서는 결론 및 향후 연구과제를 기술한다.

2. 관련 연구

2.1 Heartbeat 및 Fake

Heartbeat 은 리눅스에서 제공하는 고가용성 소프트웨어 솔루션으로, 이더넷 카드 또는 시리얼 케이블로 연결된 호스트들이 Heartbeat 이라는 프로토콜을 사용하여 주기적으로 메시지를 서로 주고 받으며 시스템의 상태를 파악한다. 일정 시간동안 Heartbeat 메시지를 받지 못하면 상대 호스트에 결함이 발생했다고 판단하고 가상 IP 를 포함한 Resource 를 인계받게 된다. 웹 서버를 구성하고 있는 마스터/백업 서버는 이 Heartbeat 를 통하여 정상동작의 여부를 파악하도록 구현되어 있으며, Roll-Forward 를 적용한 Active/Active 모드로 실행이 가능하다[1][2].

Fake 는 랜상에서의 백업서버로 스워칭하기 위해 설계된 것으로 특히, Unsheduled 또는 Scheduled Down Time

동안에 Mail, 웹, 프록시서버들을 백업하기 위해 설계되었다. 마스터서버의 결함이 발생할 경우 ARP Spoofing 기법을 사용하여 마스터서버의 가상 IP 주소를 백업 서버로 이전하게 되며, 백업서버에서도 Fake 를 가동시키거나 중지시킬 수 있다.

2.2 Rollback 및 Roll-Forward

Rollback 은 일시적이고 간헐적인 결함에 대한 Fault Tolerance 를 제공하는 비용대 효과적인 방법으로, 결함이 발생하기 전의 수행상태로 시스템을 복구하기 위해서 가장 최근 checkpoint 지점에서 다시 수행하게 하는 기법이다[3]. Rollback Recovery 를 하기 위해서는 시스템이 정상가동시에 항상 시스템의 상태를 주기적으로 저장해 두었다가 결함이 발생하면 바로 직전에 저장된 상태로 시스템을 복구한다. Roll-Forward Recovery 는 이미 언급한 Roll-Back 의 단점들을 보완하며, 제한된 Deadline 내에 작업을 빨리 복구하여 서비스를 제공하는 가용성을 증가시킬 수 있다[4][5].

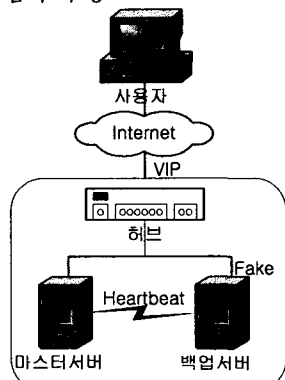
[표 1] Rollback 과 Roll-Forward 의 특징

	Rollback Recovery	Roll-Forward Recovery
장점	OS 또는 응용프로그램에 의해 복구되며, H/W로는 결함만 탐지하면 되므로 OMR 구조만으로 충분하다.	시스템의 결함발생 경우 Roll-back 를 하기 위한 복구시간이 적게 소요되므로 사용자에 대한 고품질이 제공된다.
	비용면에서 유리하다.	Checkpoint가 필요없으므로 마스터서버의 부하가 적다.
단점	표준 OS를 바로 사용하지 못하므로 독자적인 OS를 사용해야 한다.	마스터/백업서버 시스템이 모두 동작해야 하므로 시스템 측면에서 비효율적이다.
	마스터서버의 정상가동시에도 상태를 저장해야 하므로 상당한 성능저하가 생긴다.	마스터서버가 웹상에서의 트래픽으로 인한 결함이라면 백업서버도 같은 결함이 생길 수 있다.

이러한 Roll-Forward 의 단점을 고려하고라도 마스터/백업서버가 모두 동작함으로써 결함 발생시 복구하기 위해 소요되는 시간을 감소시킬 수 있으며, 또한 웹서비스를 제공받고자 하는 사용자에게는 고품질의 서비스를 제공할 수 있게 되는 것이다.

3. Roll-Forward 를 적용한 Web server 구성

3.1 서버시스템의 구성



[그림 1] 서버시스템의 구성도

3.2 서버의 동작방식

시스템이 가동되면 자동 또는 수동적으로 Heartbeat 데몬을 가동시키게 된다.

첫째, 마스터/백업서버의 Heartbeat 데몬이 가동되면서 현재 자신의 위치를 설정파일에 입력하여 인식한다.

둘째, 현재 자신이 마스터 서버로 설정되어 있다면 Heartbeat 는 가상 IP 를 자신의 eth0:0 로 취한다.

셋째, Heartbeat 데몬이 마스터/백업서버가 서로에게 살아있다는 메시지를 전송하게 된다.

넷째, 백업서버에게는 Hub 를 통해 포워딩 될 수 있도록 추가로 iptables 로 다음과 같이 실행명령을 입력한다.

'iptables'는 커널에게 어떤 패킷을 필터링할 것인지 알려주며, 패킷 필터링 테이블에 필터링 규칙을 삽입하거나 삭제할 수 있는 도구이다[6][7].

```
/sbin/iptables
```

```
iptables -A PREROUTING -t nat -p tcp -d 168.188.46.174 --dport 80 -j DNAT -- to 168.188.46.180/80
```

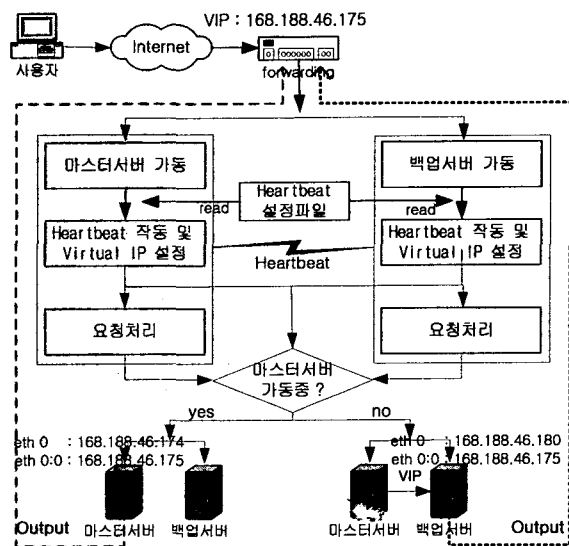
위와 같이 지정해 줌으로써 마스터서버의 80 Port 로 전달해 오는 TCP Packet 이 백업서버의 80 Port 로 포워딩되게 된다. 또한 매 부팅시마다 설정하지 않고 자동으로 실행될 수 있도록 iptables-save 를 해 두면 용이하다[8].

다섯째, 마스터서버가 Heartbeat 을 전달하지 않으면 백업서버가 Fake 를 이용하여 마스터서버와 Aliasing 되어 있던 IP 를 넘겨받게 되며, 마스터서버가 가지고 있던 자원들도 같이 넘겨 받게 된다.

Roll-Forward 를 적용하여 구현하고자 하는 웹서버의 동작 원리로는 Dummy Hub 를 통해 들어오는 사용자의 요청은 모두 마스터/백업서버에게 포워딩되고, 각자 같은 일을 수행하게 된다.

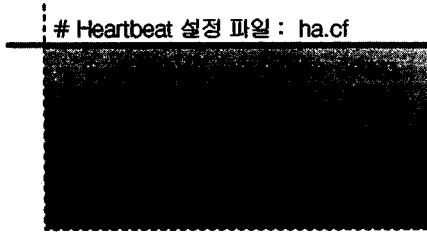
첫째, 마스터서버가 동작중일 경우, 마스터서버의 결과가 Hub 를 통해 사용자에게 보여지게 된다.

둘째, 마스터서버가 결함이 발생하게 되면, 백업서버는 Virtual IP 를 넘겨받게 되며, 백업서버의 결과가 Hub 를 통해 사용자에게 보여지게 된다.



[그림 2] 마스터/백업서버의 동작방식

Heartbeat 을 위한 주기능 설정 파일인 ha.cf 의 내용과 그 기능은 다음과 같다.

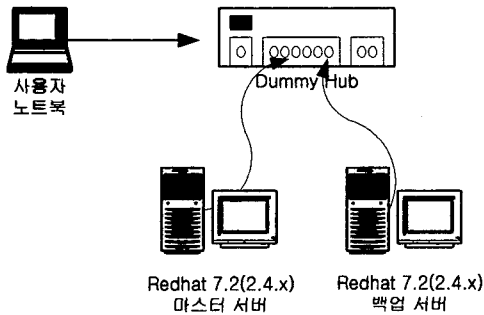


- keepalive : 각 Heartbeat 신호사이의 경과시간
- deadtime : 마스터서버의 결함발생 판단을 위한 시간
- serial : 물리적으로 연결된 시리얼 포트
- node : Heartbeat 으로 연결된 마스터/백업서버

위에서 언급한 Heartbeat 의 주기능을 설정한 파일의 내용에 의거하여 서버시스템의 위치를 인식하게 되며, 마스터/백업서버 시스템이 각자 시스템 자신의 상태를 서로에게 알리게 된다.

4. 실험 환경 및 결과

본 논문에서 연구를 위한 실험환경은 [그림 3]과 같이 마스터/백업서버 2 대와 Hub 1 대, 클라이언트 1 대로 구성된 Roll-Forward 를 적용한 고가용 웹서버이다.



[그림 3] 실험을 위한 고가용 웹서버

웹서버 프로그램은 아파치(1.x)이며, 고가용 웹서버를 구현하기 위한 실험의 각 시스템 사양은 아래 [표 2]와 같다.

[표 2] 웹서버 시스템 관련 실험환경

	마스터/백업서버	사용자
CPU	Pentium 200MHz	Pentium 4 1.7GHz
RAM	128MB	512MB
HDD	3.2GB	20GB
LAN card	10Mbps Ethernet	100/10Mbps Ethernet
OS	Redhat 7.2 (2.4.17)	Window XP

본 논문에서의 실험결과는 마스터서버와 백업서버를 서로 연결하고 있는 Heartbeat 의 기능을 확인하였지만, 사용자로부터 요청을 받아 마스터/백업서버가 각각 요청을 처리한 후 결과를 다시 되돌려 주는 실험은 아직 하지 못한 상태이다. 그러나 실험을 한 후 결과를 도출할 수 있을 것이라 사료되며, 논문의 서두에서 제시되었던 바와 같이 마스터/백업서버의 실행모드가 Active/Active 이며, Roll-Forward Recovery 방식을 적용한 고가용 웹서버를 구현함으로써 사용자에게 고가용의 웹 서비스를 제공할 수 있다.

한편, 마스터서버 및 백업서버의 실행모드가 Active/Active 이므로, 서비스를 제공받고자 하는 사용자로부터 들어오는 요청의 과중한 트래픽으로 인해 마스터서버가 결함이 발생하는 경우 백업서버에도 같은 현상이 일어날 수 있다는 것이다. 이에 대해 마스터서버와 백업서버가 요청을 받아들이는 데 있어서 시간적인 차이를 두거나, 또는 마스터서버의 부하를 분산하여 처리할 수 있는 Real Server 들 등으로써 위에서 언급한 문제점에 대한 대안으로 제시될 수 있다고 하겠다.

5. 결론 및 향후 연구과제

본 논문에서는 이미 언급된 바와 같이 Roll-Forward Recovery 를 적용한 고가용 웹서버를 구현하고자 그 방안을 제시하였다. 웹서버에 Roll-Forward Recovery 기법을 적용함으로써 마스터서버의 결함발생시 복구하는데 소요되는 시간을 최소화할 수 있으며, Active/Active 모드로 실행함으로써 서비스를 제공받고자 하는 사용자에게 대해 고가용의 서비스를 제공할 수 있다[5].

향후 연구과제로는 웹서버로서 그 기능을 다할 수 있도록 실험을 통하여 실험결과를 도출하는 것이며, 현재는 Dummy Hub 를 통하여 들어오는 패킷을 마스터/백업서버에 포워딩함으로써 동시에 같은 작업을 처리할 수 있도록 하였는데, 앞으로는 Hub 를 통하지 않고 마스터서버에 들어오는 요청을 백업서버에 어떻게 포워딩하며, 내용의 일관성을 위해 어떤 처리가 필요할 것인지, 사용자에게 결과는 어떻게 보낼 것인지에 대해 연구할 계획이다.

참고문헌

- [1] High-Availability Linux Project, " <http://www.linux-ha.org>"
- [2] LinuxOne High Availability Web Server, " <http://alpha-cluster.linuxone.co.kr>"
- [3] D.K. Pradhan and N.H. Vaidya, " Fault Tolerant Computer System Design ", pp.160-181, 1996
- [4] D.K. Pradhan and N.H. Vaidya, " Roll-Forward checkpointing scheme : Concurrent retry with nondedicated spares ", in IEEE Workshop on Fault Tolerant Parallel and Distributed Systems, pp. 166-174, July 1992.
- [5] D.K. Pradhan and N.H. Vaidya, " Fault Tolerant Computer System Design ", pp.181-192, 1996
- [6] 리눅스 2.4 패킷 필터링 하우투, " <http://www.kldp.org>"
- [7] 리눅스 2.4 NAT HowTo : NAT 조절하기, " <http://www.kldp.org>"
- [8] Linux Kernel 2.4 네트워크 트래픽 분산, " <http://www.kldp.org>"