비대칭 Heartbeat을 적용한 리눅스 기반 고가용 클러스터의 구현

임은지[0] 안창원 정성인
한국전자통신연구원
(ejlim, ahn, sijung)@etri.re.kr

Implementation of the High Availability Cluster based on unsymmetrical heartbeat

Eun-Ji Lim[0] Chang-Won Ahn Sung-In Jung
Electronics and Telecommunications Research Institute

요 약

인터넷의 사용자가 급증하여 고가용성과 확장성을 지닌 고성능의 인터넷 서버들이 요구된다. 클러스터 시스템은 이러한 요구사항을 만족시킬 수 있는 서버이다. 본 연구에서는 N-node heartbeat 을 구현하였고, 이것을 기반으로 하는 고가용 부하분산 클러스터, PersistentCluster를 구현하였다. PersistentCluster는 로드밸런서가 사용자의 요구를 서버들에게 분산시켜주는 LVS 시스템에서 로드밸런서가 고장나면 나머지 서버 중에 하나가 그 역할을 인계 받아 계속 수행하는 고가용성 클러스터링 솔루션이다. PersistentCluster는 로드밸런서만 heartbeat 메시지를 전송하는 비대칭 heartbeat을 채택하여 시스템의 메시지 전송 및 처리 오버헤드를 감소시켰다. 비대칭 heartbeat을 적용할 경우에 나타나는 각 노드의 부하 감소량을 실측하여 비대칭 heartbeat 의 성능을 평가하였다.

1. 서 론

인터넷의 사용자수가 폭발적으로 증가하여 서버의 부하는 더욱 가중되었다. 인터넷 서버는 보다 많은 사용자에게 서비스를 제공할 수 있는 고성능과 지속적인 서비스를 제공할 수 있는 고가용성을 동시에 갖추어야 한다. 이러한 요구사항을 만족하는 것으로서, 서버의 확장성과 가용성, 그리고 가격대 성능비의 면에서 우수한 클러스터 서버가 활용되고 있다[2].

Linux Virtual Server (LVS)[1,4]는 로드밸런서와 서비스 노드들로 구성된 클러스터 서버이다. 로드밸런서는 사용자의 서비스 요청을 받아서 서비스 노드들에게 분산시킨다. LVS는 로드밸런서가 고장이 날 경우에 전체 시스템이 서비스를 제공할 수 없게 된다는 단점을 가지고 있다. 그래서 일반적으로 로드밸런서의 백업서버를 두어서 로드밸런서가 다운될 경우 백업서버가 로드밸런서의 역할을 수행하도록 한다. 그러나, 로드밸런서를 위한 전용의 백업서버를 둔다는 것은 시스템 자원의 낭비가 될 수 있고, 백업서버까지 고장나는 경우에는 서비스를 제공할 수 없다는 문제점이 있다.

본 연구에서는 N개의 노드들이 서로 감시하며 한 노드가 고장날 경우 나머지 중에 하나가 그 역할을 인계받는 N-node heartbeat을 구현하였다. 또한, N-node heartbeat을 기반으로 구현된 PersistentCluster는 확장성과 고가용성을 동시에 갖춘 고성능의 인터넷 서버를 구성하는데 이용될 수 있는 클러스터링 솔루션이다. PersistentCluster에서 로드밸런서가 고장난 경우 나머지 서버 중에 하나가 로드밸런서가 될 수 있다. 또한, 본 연구에서는 클러스터의 모든 노드가 heartbeat 메시지를 전송하는 경우에 발생할 수 있는 메시지 전송 및 처리 오버헤드를 감소시키기 위하여 로드밸런서만 heartbeat 메시지를 전송하는 비대칭 heartbeat 기능을 구현하였다. 모든 노드가 heartbeat 메시지를 전송하는 경우와 로드밸런서만 heartbeat 메시지를 전송하는 경우에 각 노드들의 부하를 측정하여 비대칭 heartbeat의 성능을 측정하였다.

본 논문은 다음과 같이 구성된다. 2장에서 관련연구를 살펴보고, 3장에서는 PersistentCluster의 구성을 기술한다. 4장에서는 PersistentCluster의 failover 방식과 비대칭 heartbeat의 동작, 성능평가를 보여준다. 마지막 5장에서 결론 및 향후 연구 과제를 기술하겠다.

2. 관련연구

2.1 Linux Virtual Server (LVS)

LVS[1,4]는 로드밸런서와 서버로 구성되었고, 로드밸런서는 들어오는 사용자의 요청을 서로 다른 서버로 전달한다. LVS는 사용자에게 하나의 서비스 IP를 가진 단일 서버로 보이기 때문에 클라이언트에는 어떠한 수정도 필요하지 않다. 서비스 노드를 추가, 제거 할 수 있으며, 노드를 감시하여 고장 날 경우에 클러스터를 재구성 할 수 있다. 따라서 확장성과 고가용성을 동시에 갖추었고 볼 수 있다. 그러나, LVS는 로드밸런서가 병목 현상을 일으킬 수 있으며, 로드밸런서가 고장 날 경우에 전체 시스템의 서비스 중단을 초

래 할 수 있다는 단점이 있다.

2.2 Heartbeat

Heartbeat은 Linux HA 공개 프로젝트[3]에서 제공하는 고가용성 소프트웨어 솔루션이다. 이더넷 카드 혹은 시리얼케이블로 연결된 호스트들이 heartbeat 이라는 특정한 메시지를 서로 주고 받으며 상태를 파악한다. 일정 시간 동안 heartbeat 메시지를 받지 못하면 상대 호스트에 결함이 발생했다고 판단하고 가상 IP를 포함한 리소스를 인계받는다. 현재의 heartbeat은 2개의 노드로 정상 동작하도록 구현되어 있으며, active-standby, active-active 모드로 실행이 가능하다.

3. PersistentCluster의 구성

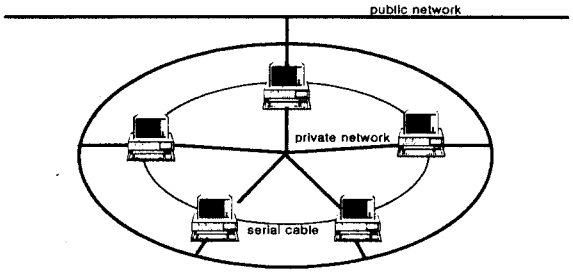
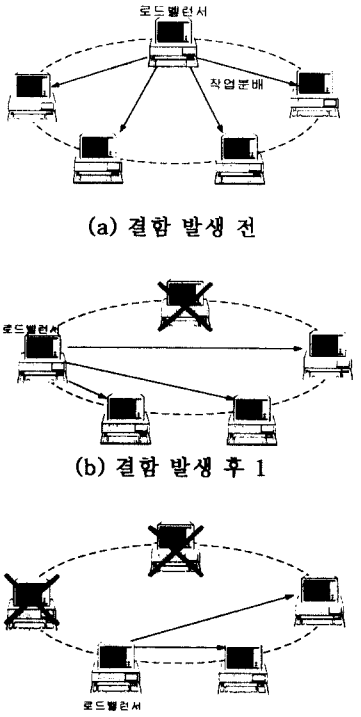


그림 1 PersistentCluster의 구성

PersistentCluster의 네트워크 구성은 그림 1과 같다. 각 노드에는 2개의 이더넷 카드가 장착되어 있어서, 하나는 클라이언트에 대한 서비스를 제공하기 위한 외부 네트워크로 연결되고, 나머지 하나는 heartbeat 전용의 내부 네트워크로 연결된다. 또한, 이더넷 카드나 내부 네트워크 상의 문제가 발생할 경우에도 heartbeat 메시지를 전송 가능하게 하기 위하여 각 노드를 시리얼 케이블을 이용하여 원형으로 연결한다. 각 노드들은 주기적으로 내부네트워크와 시리얼 케이블을 통하여 heartbeat 메시지를 전송하고, 서로의 상태를 파악한다. 만일, 특정 노드로부터 일정시간동안 메시지가 도착하지 않으면 그 노드에 결함이 발생했다고 판단한다.

4. PersistentCluster의 동작

LVS에서는 로드벨런서가 다운될 경우 전체 시스템이 서비스 중단 상태에 빠진다. PersistentCluster에서는 로드벨런서가 다운될 경우 선출 정책에 의하여 나머지 노드 중에 하나가 새로운 로드벨런서로 결정된다. 그림 2는 노드의 순서에 의해서 다음 로드벨런서가 결정되는 정적인 선출 정책을 사용했을 경우의 예를 나타낸 것이다.



(a) 결함 발생 전

(b) 결함 발생 후 1

(c) 결함 발생 후 2

그림 2 PersistentCluster의 failover

로드벨런서는 클러스터의 구성 정보를 유지하여야 한다. PersistentCluster에서는 모든 노드가 로드벨런서가 될 후보 노드이기 때문에, 로드벨런서는 클러스터의 구성 정보에 변화가 있을 때마다 그 정보를 나머지 노드들에게 전달한다. 그래서 모든 노드는 클러스터의 구성 정보를 유지하게 되고, 다음에 새로운 로드벨런서로 선출되었을 때 정상 동작을 할 수 있다.

4.1 비대칭 heartbeat

N개의 노드로 구성된 클러스터에서 모든 노드가 주기적으로 heartbeat 메시지를 나머지 노드에게 전송한다면, 노드 수가 증가함에 따라 각 노드가 받는 heartbeat 메시지의 양도 증가할 것이고, heartbeat 메시지를 처리하는 것은 노드에 상당량의 부하를 초래할 것이다. 따라서, PersistentCluster에서는 클러스터에 속한 노드 중에서 로드벨런서만 다른 노드로 주기적인 heartbeat 메시지를 전송하는 비대칭 heartbeat 기능을 구현하였다.

비대칭 heartbeat을 적용할 경우에 각 노드들은 로드벨런서의 상태만 파악 할 수 있고 나머지 다른 노드의 상태를 알 수 없다. 그래서, 로드벨런서가 다운 된 후에 나머지 노드들의 상태를 서로 파악하고 살아있는 노드 중에서 새로운 로드벨런서를 선출하기 위하여 메시지를 교환한다. 비대칭 heartbeat을 사용할 경우에 로드벨런서 failover 절차는 그림3과 같다.

로드벨런서가 다운되었음을 가장 먼저 발견한 노드는 DEAD_NOTI 메시지를 모든 노드에게 전송한다. 그리고, 자신의

상태 메시지를 전송하고, timer를 설정하여 일정 시간동안 다른 노드의 상태 메시지를 받는다. DEAD_NOTI 메시지를 받은 노드는 자신의 상태 메시지를 다른 모든 노드에게 전송하고, 역시 timer를 설정하여 다른 노드의 상태 메시지를 받는다. Timer가 종료되면 상태 메시지를 보내오지 않은 노드를 DEAD 상태로 표시하고, 살아있는 노드 중에서 선출 정책에 따라 새로운 로드벨런서를 결정한다.

```
If (do not receive the heartbeat message from load balancer for a
fixed time interval) {
    send DEAD_NOTI_MSG to all nodes
    send LOCAL_STATUS_MSG to all nodes
    set timer
}
else if (receive DEAD_NOTI_MSG from other node) {
    send LOCAL_STATUS_MSG to all nodes
    set timer
}
if (timer expired) {
    while (all nodes)
        check node status
    determine new load balancer according to voting algorithm
    if (current node == new load balancer)
        takeover load_balancer
}
```

그림 3 로드벨런서의 failover

4.2. 비대칭 heartbeat의 성능 평가

비대칭 heartbeat 방식을 적용할 경우에, 로드벨런서 측에서는 다른 노드로부터 메시지를 받아서 처리하지 않기 때문에 메시지 처리에 따른 부하가 감소한다. 또한, 나머지 노드들은 heartbeat 메시지를 전송하지 않을 뿐만 아니라, 로드벨런서를 제외한 다른 노드로부터의 heartbeat 메시지를 받아서 처리하지 않기 때문에 노드의 부하를 더욱 감소시킬 수 있다.

표1은 비대칭 heartbeat 방식에 따라 통신할 경우와 그렇지 않고 모든 노드가 heartbeat 메시지를 전송하는 경우에 클러스터 노드들의 CPU 부하를 측정한 결과이다. 클러스터의 모든 노드들은 어떠한 작업도 수행하지 않고 오직 heartbeat 통신만을 수행하며, 모든 메시지는 UDP broadcast로 전송된다. heartbeat 프로세스의 초당 CPU time을 총 2시간 동안 측정하여 평균을 계산하였다.

표1의 all-node hb는 모든 노드가 주기적으로 heartbeat 메시지를 전송하는 경우로서, 클러스터의 노드 수가 증가할수록 각 노드의 부하가 증가함을 보여준다. 이것은 총 노드 수가 N일 때 각 노드가 (N-1)개의 노드로부터 주기적으로 전송되는 메시지를 처리하기 때문이다. 따라서, 메시지의 크기가 크고 노드 수가 증가할수록 각 노드의 부하는 더 증가할 것임을 알 수 있다. 비대칭-hb lb는 비대칭 heartbeat을 적용한 경우에 load balancer 의 부하량으로서, 전체 노드 수에 무관하게 메시지를 전송하는 데에는 거의 부하가 없음을 알 수 있다. 그리고 비대칭-heartbeat rs 는 비대칭 heartbeat일 때 서비스 노드의 부하량으로서, 총 노드 수와 무관하게 비슷한 값이 나타나며 전체적으로 작은 값을 보여준다.

표 1 노드의 CPU 부하 (μsec/sec)

총 노드수	2-node	3-node	4-node	5-node
all-node hb	444.44	858.33	1279.17	1879.17
비대칭-hb lb	0.00	0.00	0.00	0.00
비대칭-hb rs	344.44	344.44	400.00	331.94

결과적으로, 클러스터의 노드수가 증가하면 heartbeat 메시지 전송 및 처리에 따른 부하가 노드에 상당한 부담을 가져올 수 있으나, 비대칭 heartbeat 방식을 사용하여 이러한 문제를 해결할 수 있음을 알 수 있다.

5. 결론 및 향후 연구 과제

본 연구에서는 고가용성과 확장성을 갖춘 클러스터링 솔루션 PersistentCluster를 구현하였다. 이것은 N-node heartbeat과 LVS를 기반으로 구현된 것으로서, 하나의 로드벨런서와 여러 서비스 노드로 구성되며 로드벨런서에 결함이 발생한 경우에 살아있는 서비스 노드 중에 하나가 새로운 로드벨런서로 결정되고 로드벨런서의 기능을 인계 받아 수행하게 된다.

Heartbeat 통신을 기반으로 하는 고가용성 클러스터 시스템에서 노드 수가 많아질 경우에 heartbeat 통신 자체가 노드에 상당한 부하를 초래할 수 있다는 점을 고려하여, PersistentCluster에서는 비대칭 heartbeat 통신 방식을 채택하였다. 즉, 클러스터에 속한 노드 중에 로드벨런서만 주기적으로 heartbeat 메시지를 전송한다. 이렇게 하여 클러스터에 속한 모든 노드의 메시지 처리 부하를 감소시킬 수 있다. 성능 평가에서는 실측을 통하여 이것을 증명하였다.

비대칭 heartbeat을 적용한 경우에는 로드벨런서에 결함이 발생한 후 새로운 로드벨런서가 결정되기까지 시간이 더 길어질 수 있다. 즉, 서비스 노드 간의 상태 파악을 위한 시간이 필요하기 때문이다. 이것은 비대칭 heartbeat의 단점이라 할 수 있다. 그러나, 비대칭 heartbeat을 채택하지 않았을 때 발생 가능한 노드의 결함이나 노드의 성능저하는 클러스터 시스템의 전체 성능에 큰 영향을 미치므로 방지할 필요가 있다고 판단된다.

현재 PersistentCluster에서 로드벨런서 선출 정책은 노드의 순서를 기반으로 하는 정적인 방식을 사용하고 있다. 향후, 노드의 부하 정보나 성능과 같은 변수를 고려하는 동적 선출 정책을 채택할 계획에 있으며, 다양한 환경에서의 성능 평가를 실시할 계획에 있다.

6. 참고 문헌

[1] Wensong Zhang, "Linux Virtual Server for Scalable Network Services", in the Ottawa Linux Symposium 2000.

[2] Gregory F. Pfister, In search of Clusters, Prentice Hall PTR, 1998.

[3] High-availability Linux Project. http://linux-ha.org

[4] W. Zhang and et al. Linux Virtual Server project. http://www.LinuxVirtualServer.org/