Red Hat 클러스터 관리자

Red Hat 클러스터 관리자 설치와 관리 가이드

Red Hat, Inc.

Red Hat 클러스터 관리자 : Red Hat 클러스터 관리자 설치와 관리 가이드
저작권 ‘
‘
2002 Red Hat, Inc.

차례

감사의 글

Red Hat 클러스터 관리자 소프트웨어는 Mission Critical Linux, Inc에서 개발한 오픈 소스 kimberlite http://oss.missioncriticallinux.com/kimberlite/ 클러스터 프로젝트에 기반하여 만들어 졌습니다.

Red Hat 개발자들은 Kimberlite를 기본으로 만들어진 소프트웨어에 보다 많은 개선점과 기능 수정을 추가하였습니다. 다음 목록은 이러한 개선점을 간략하게 설명하고 있습니다.

- 최종 사용자가 쉽고 편하게 사용할 수 있도록 Red Hat 설치 프로그램에 패키지를 통합하였습니다.
- 고성능 NFS 서비스를 위한 지원 추가.
- 고성능 Samba 서비스를 위한 지원 추가.
- 데이터 무결성을 지키기 위한 감시 타이머 사용 지원 추가.
- 자동으로 실패한 응용 프로그램을 재시작시키는 서비스 모니터링 기능 추가.
- 추가 클러스터 작업을 쉽게 사용할 수 있도록 서비스 관리자 프로그램 재작성.
- 그래픽 모니터링 도구인 Red Hat 클러스터 관리자 GUI 추가.
- 다양한 버그 수정.

Red Hat 클러스터 관리자 소프트웨어는 Linux-HA 프로젝트 http://www.linux-ha.org/stonith/의 STONITH 준수 전원 스위치 모듈을 통합합니다.

1장 .

Red Hat 클러스터 관리자에 오신 것을 환영합니다.

Red Hat 클러스터 관리자은 기술의 조합체로써, 시스템 상의 문제 발생시 프로그램들의 유용성을 지속하고 데이터의 무결성을 제공하는 능력을 갖추고 있습니다. 중복 하드웨어와 공유 저장 공간, 전원 관리, 그리고 견고한 클러스터 통신과 프로그램 복구 메커니즘을 사용하여, 기업체 시장의 필요를 충족시킵니다.

특히 데이터베이스 프로그램, 네트워크 파일 서버 그리고 동적인 내용을 가진 World Wide Web (Web) 서버에 알맞습니다. 또한 리눅스 가상 서버 (LVS) 프로젝트에 기반한 Piranha 부하분산 클러스터 소프트웨어와 함께 사용하여, 데이터 무결성과 프로그램의 최대 능력 발휘를 필요로 하는 고성능 전자상거래 사이트에 이용할 수 있습니다. 보다 많은 정보는 B.5 절을 참조하시기 바랍니다.

1.1. 클러스터 개요

클러스터를 설정하기 위해서 서버 관리자는 클러스터 시스템들 (자주 구성원 시스템으로 불립니다)을 클러스터 하드웨어에 연결 하고 시스템을 클러스터 환경에 맞추어 설정합니다. 클러스터의 근본이 되는것은 고급 호스트 구성원 알고리즘입니다. 이 알고리즘은, 다음과 같은 노드 내부 통신을 사용하여, 데이터의 무결성을 항상 지켜줍니다:

- 시스템 상태 정보를 저장한 공유 저장 디스크 상의 *Quorum* 파티션.
- *heartbeat* 채널을 위한 클러스터 시스템 간의 이더넷과 벙렬 연결.

클러스터에서 프로그램이나 데이터를 고성능화 하시려면, 서버 관리자는 응용 프로그램이나 공유 저장 디스크와 같은 분리된 서비스 설정과 자원인 클러스터 서비스를 설정하셔야 합니다. 서비스에는 외부 클라이언트에서 서비스로 윤활한 사용을 위해 IP 주소가 주어집니다. 예를 들면, 서버 관리자는 클라이언트에게 고성능 데이터베이스 응용 프로그램 데이터를 사용할 수 있는 서비스를 제공해 주는 클러스터 서비스를 설정할 수 있습니다.

양쪽 클러스터 시스템은 어떠한 서비스도 실행 가능하며, 공유 저장 디스크에 있는 서비스 데이터를 사용할 수 있습니다. 그러나, 각 서비스는 데이터의 무결성을 위해 한번에 한 클러스터 시스템에서만 실행될 수 있습니다. 서버 관리자는 양 클러스터 시스템이 다른 서비스를 실행하는 *active-active* 설정을 구성하거나 일차 클러스터 시스템이 모든 서비스를 실행하고 백업 클러스터 시스템은 일차 시스템이 실패할 경우에 작업을 이어받아 실행하는 방식인 *hot-standby* 설정을 구성할 수도 있습니다.

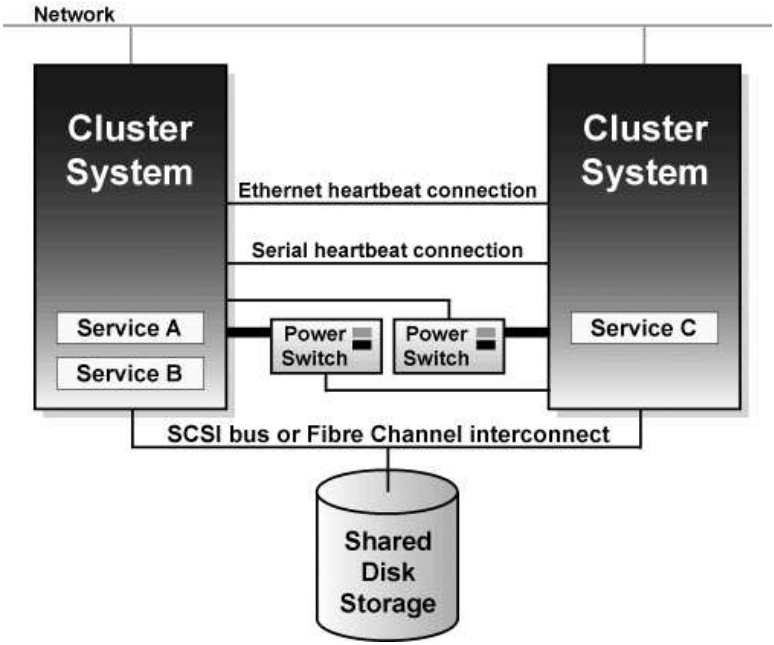

그림 1-1. 클러스터의 예

그림 1-1에서는 클러스터 중 active-active 설정의 예를 보여주고 있습니다.

만일 하드웨어나 소프트웨어에 문제가 생길 경우, 클러스터는 자동적으로 문제가 생긴 시스템의 서비스들을 다른 작동중인 클러스터 시스템에서 실행되도록 합니다. 이 오류 복구 기능을 통해 데이터의 문제를 최소화시키며, 사용자의 불편을 또한 최소화시킬 수 있습니다. 문제를 일으킨 서비스가 재시작되면, 클러스터는 자동적으로 양 시스템에서 서비스들을 재배치시킬 수 있습니다.

추가로 클러스터 관리자는 깨끗하게 서비스를 현재 실행중인 클러스터 시스템에서 정지하고 다른 시스템에서 재시작할 수 있습니다. 이 서비스 재배치기능을 통해 관리자는 클러스터 시스템 관리를 할때 프로그램이나, 데이터를 손쉽게 관리할 수 있습니다.

1.2. 클러스터 기능

클러스터는 다음과 같은 기능을 갖추고 있습니다:

- 하드웨어 이중화 (No-single-point-of-failure) 설정

  클러스터는 이중 제어장치 RAID 어레이, 여러 개의 네트워크와 병렬 통신 채널, 그리고 중복 무정전 전원공급장치 UPS (Uninterruptible power supply)를 통해 하드웨어 하나의 문제로 시스템이 중지되거나 데이터를 잃는 일이 없도록 합니다.

  대안으로서 하드웨어 이중화 (no-single-point-of-failure) 클러스터 보다 낮은 성능을 제공하는 저가의 클러스터를 설정하실 수도 있습니다. 예를 들면 서버 관리자는 단독 제어장치 RAID 어레이와 단독 *heartbeat* 채널만 사용하여 클러스터를 설정 가능합니다.

주의

소프트웨어 RAID나 멀티 이니시에이터(multi-initiator) 병렬 SCSI와 같은 특정 저가 하드웨어는 공유 클러스터 디스크에 사용하기에 부적합 합니다. 보다 많은 정보를 원하신다면, 2.1 절을 참조하시기 바랍니다.

- 서비스 설정 기본틀

  클러스터는 관리자가 쉽게 각 서비스를 고성능으로 설정할수 있도록 되어 있습니다. 서비스를 만들려면, 관리자가 서비스에 필요한 자원과 서비스 관련 자료, 서비스 이름, 프로그램 시작과 정지 스크립트, 디스크 파티션들, 장착점, 그리고 클러스터 시스템중 어느 시스템에서 서비스를 실행 하고자 하는지만 지정해 주면 됩니다. 관리자가 서비스를 다한 후, 클러스터가 모든 시스템이 사용할 수 있는 공유 저장 디스크에 있는 클러스터 데이터베이스에 입력할 수 있습니다.

  클러스터는 데이터베이스 프로그램들이 쉽게 사용할 기본틀을 제공하고 있습니다. 예를 들어, 데이터베이스 서비스은 고성능 자료를 데이터베이스 프로그램에 제공하고 있습니다. 클러스터 시스템에서 실행되고 있는 프로그램들은 네트워크를 통해 웹 서버 같은 데이터베이스 클라이언트 시스템 자료를 제공합니다. 만일 서비스에 문제가 생길 경우 다른 클러스터 시스템이 이 일을 담당할 것이며, 프로그램은 역시 공유된 데이터베이스 데이터를 사용할 것입니다. 네트워크-이용 데이터베이스 서비스는 보통 IP 주소를 사용하나, 이 주소 또한 서비스에 문제가 있을때, 클라이언트의 이용에 문제가 없도록 서비스와 함께 같이 복구됩니다.

  클러스터 서비스 기본틀은 다른 프로그램에도 쉽게 사용될 수 있습니다.

- 데이터 무결성 보장

  데이터의 무결성을 보장하기 위해, 오직 한 클러스터 시스템이 서비스를 실행 할 수 있으며, 서비스 데이터를 이용할 수 있습니다. 클러스터 설정에 있는 전원 스위치를 사용해, 문제 복구시에 시스템이 재시작하기 전에 전원-사이클을 통해 다른 시스템이 먼저 시작 할수 있도록 하고 있습니다. 이것은 두 시스템이 동시에 같은 데이터를 이용해 데이터를 손상시키는 것을 막고 있습니다. 이것이 꼭 필요한 것은 아니나, 무결성은 전원 스위치를 통해 어떤 문제 상에서도 데이터의 무결성을 가지는 것이 중요합니다. 감시 타이머 (Watchdog timer)가 전원 조정의 한 방법으로 사용될 수도 있습니다.

- 클러스터 관리 사용자 인터페이스

  사용자 인터페이스는 클러스터 관리를 쉽게 만들며, 관리자가 쉽게 서비스를 만들고, 시작하고, 정지하고, 재배치시키며, 클러스터를 감시할 수 있게 합니다.

- 다중 클러스터 통신 방법

  다른 클러스터 시스템의 상태를 감시하려면, 각 클러스터 시스템이 원격 전원 스위치를, 만일 있다면, 감시하며, *heartbeat* 핑을 네트워크와 병렬 채널을 통해 보내 다른 클러스터 시스템의 상태를 확인합니다. 그 위에, 각 클러스터 시스템은 주기적으로 시간 도장과 클러스터 상태를 공유 저장 디스크에 있는 두 quorum 파티션에 적습니다. 시스템 상태에는 시스템이 활동적 클러스터 구성원인지를 포함하고 있습니다. 서비스 상태에는 서비스가 현재 실행되고 있는지, 그리고 어떤 클러스터 시스템에서 실행되고 있는지 여부가 포함됩니다. 각 클러스터 시스템은 다른 시스템의 상태가 최근 것인지 확인합니다.

  클러스터의 무결성을 위해서, 만일 시스템이 양 quorum 파티션에 시작 시간을 쓸 수가 없다면, 클러스터에 연결 할 수 없습니다. 추가로 만일 클러스터 시스템이 시간 도장을 업그레이드 하지않고, 그 시스템로의 *heartbeats*이 실패할 경우, 이 시스템은 클러스터에서 제외됩니다.

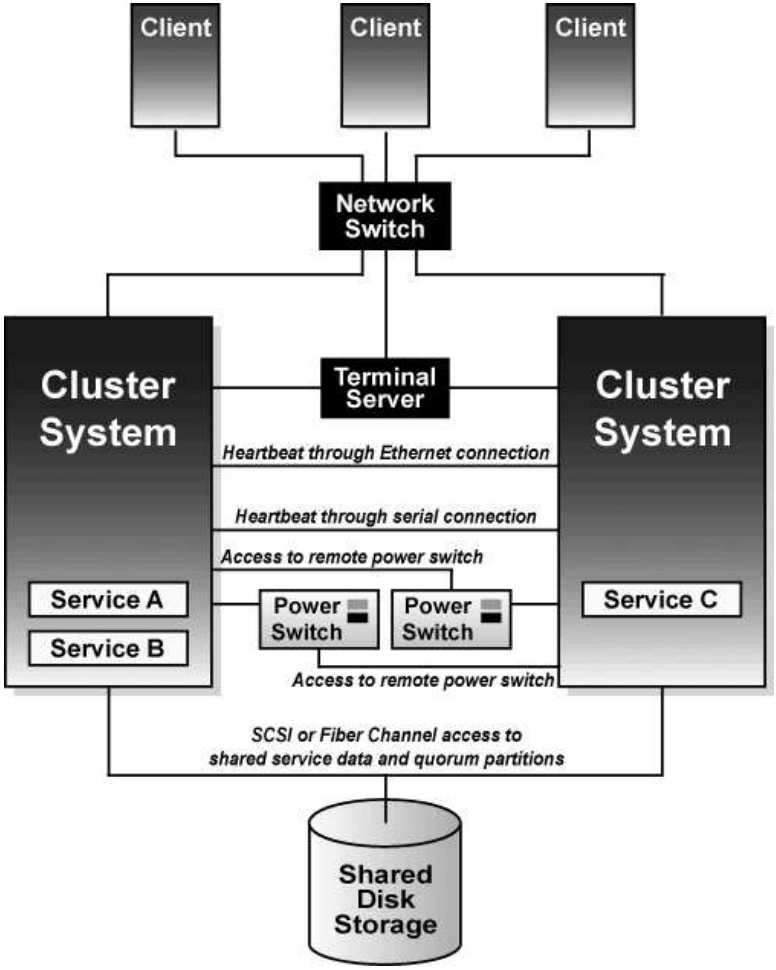

그림 1-2. 클러스터 통신 메커니즘

그림 1-2에는 어떻게 시스템들이 클러스터 설정안에서 통신하는지를 보여주고 있습니다. 기억할 것은 클러스터에서는 병렬 포트를 통해 사용되던 터미널 서버를 더 이상 사용하지 않는다는 것입니다.

- 서비스 문제 복구 기능

  만일 하드웨어나 소프트웨어에 문제가 생길 경우, 클러스터가 프로그램와 데이터의 무결성을 위해 알맞는 조치를 취할 것입니다. 예를 들어, 만일 클러스터 시스템이 완전히 다운될 경우, 다른 클러스터 시스템이 이 서비스들을 시작합니다. 이미 실행 중인 서비스에는 전혀 문제가 되지 않습니다.

  만일 실패한 시스템이 재시작하며, quorum 파티션에 쓸수 있게 되면, 클러스터에 다시 연결할 수 있으며, 다시 서비스들을 시작할 수 있습니다. 어떻게 서비스들이 설정 되었는지에 따라, 클러스터가 두 클러스터 시스템 안에서 서비스들을 알맞게 재배치 할 것입니다.

- 수동 서비스 재배치 기능

  자동 문제 복구에 덧붙여서, 클러스터는 관리자가 깨끗하게 한 클러스터 시스템에서 실행 중인 서비스를 멈추고 다른 시스템에서 재시작 할수 있게 합니다. 이것은 관리자가 데이터나 프로그램의 무결성에 신경쓰지 않고 클러스터 시스템에 미리 계획하고 관리를 할 수 있도록 도와줍니다.

- 이벤트 기록 기능

  문제가 발생시 서비스의 기능에 문제를 일으키기 전에 사전에 방지하고 고치는데 도움이 되도록, 클러스터 데몬은 기본 리눅스 syslog 하부히스템을 사용하여 메시지를 기록합니다. 서버 관리자는 메시지 기록에 사용될 보안 수준을 정할 수 있습니다.

- 응용 프로그램 모니터링

  클러스터 서비스는 프로그램의 상태를 감시할 수 있게 되어 있습니다. 이를 통해, 만일 응용 프로그램 관련 문제가 생길 경우, 클러스터는 자동적으로 프로그램을 재시작 할 수 있습니다. 응용 프로그램 문제의 경우 기본적으로 실행 중이던 구성원에서 재시작하려고 할것이며, 만일 이것에 실패하면, 다른 클러스터 구성원에서 시작합니다.

- 상태 모니터링 에이전트

  클러스터 상태 모니터링 에이전트는 중요한 클러스터와 응용프로그램 자료들을 수집합니다. 이 자료들은 로컬에서나 원격으로도 이용할 수 있습니다. 그래픽 사용자 인터페이스는 다른 시스템의 성능에 방해되지 않게 수집한 자료들을 보기 쉽게 표시합니다.

1.3. 이 책을 사용하는 방법

이 책에는 클러스터 하드웨어를 설정하는 방법과 리눅스 배포판과 클러스터 소프트웨어 설치에 대해 설명하고 있습니다. 이러한 작업들은 2 장와 3 장에 자세히 서술되어 있습니다.

클러스터 서비스의 설정과 관리에 대한 자료는 4 장을 참조하시기 바랍니다. 클러스터 관리에 대한 자료는 8 장에서 찾으실 수 있습니다.

부록 A에는 특정 하드웨어 장치를 어떻게 설정해야 하는지 또 공유 저장 장치의 설정에 대해 설명하고 있습니다. 부록 B에는 클러스터 소프트웨어에 대한 뒷배경과 다른 정보들이 있습니다.

2장 .

하드웨어 설치와 운영 체제 설정

하드웨서 설정과 리눅스 배포판을 설치하려면 다음과 같은 방법을 따르시기 바랍니다:

- 응용 프로그램과 사용자의 필요에 맞는 클러스터 하드웨어 설정을 선택합니다, 2.1 절를 참조하십시요.
- 설정하고, 클러스터 시스템들과 옵션인 콘솔 스위치와 네트워크 스위치 또는 hub, 2.2 절를 참조하십시요.
- 클러스터 시스템에 리눅스 배포판을 설치와 설정하십시요. 2.3 절를 참조 하십시요.
- 나머지 클러스터 하드웨어 부품을 설치하시고 클러스터 시스템에 연결하십시요. 2.4 절를 참조 하십시요.

하드웨어를 설정하고 리눅스 배포판을 설치한후, 클러스터 소프트웨어의 설치가 가능합니다.

2.1. 하드웨어 설정 선택하기

Red Hat 클러스터 관리자은 관리자에게 필요한 하드웨어를 이용해 프로그램과 사용자가 필요한 성능과 데이타 무결성을 갖춘 클러스터 설정을 할수 있게 도와줍니다. 클러스터 하드웨어는 클러스터 작동에 꼭 필요한 하드웨어만을 이용한 값싼 최소의 설정부터 충분한 Heartbeat 채널들, 하드웨어 RAID, 그리고 전원스위치들을 가진 고가의 설정까지 다양합니다.

하드웨어의 고장이 시스템의 다운타임의 근본 문제이므로, 설정과는 관계없이, 클러스터에는 고성능의 하드웨어를 사용하는 것을 권장합니다.

모든 클러스터 설정이 유용성을 제공하지만, 어떤 설정들은 거의 모든 문제에 대비하고 있습니다. 덧붙여서, 모든 클러스터 설정은 데이타 무결성을 제공 합니다, 하지만 어떤 설정은 모든 문제에서 데이타를 보호합니다. 그렇기 때문에, 관리자는 자신의 컴퓨터의 환경을 이해해야 하며, 각각 하드웨어의 유용성과 데이타 무결성 성능을 알고 필요한 하드웨어를 선택하여야 합니다.

클러스터 하드웨어 설정을 선택할때는, 다음을 유념해야 합니다:

프로그램과 사용자가 필요로 하는 성능

‘ 적당한, 메모리, CPU와 I/O 자원을 제공할 하드웨어 설정을 선택하십시요. 앞으로의 작업의 증가를 고려해서 설정을 선택하십시요.

가격 제한

‘ 하드웨어 설정은 예산에 맞추어 선택해야 합니다. 예를 들어, 다중 I/O 포트가 있는 시스템은 저가의 시스템보다 많이 비쌉니다.

유용성 필요

‘ 만일 컴퓨팅 환경이 고도의 유용성을 필요로 한다면, 예를 들어 제품 생산 환경, 그렇다면 클러스터 하드웨어 설정은 모든 시스템 문제, 디스크, 저장장치간의 연결, Heartbeat 채널, 그리고 전원 문제에도 걱정이 없어야 합니다. 하지만, 유용성의 끊김에도 괜찮은 환경, 예를들어 개발 환경같은 경우는 그와 같은 보호가 필요하지 않습니다. 높은 유용성을 위한 충분한 하드웨어에 관련한 자료를 원하시면 2.4.1 절, 2.4.3 절, 와 2.4.4 절를 참조하시기 바랍니다.

모든 문제가될 환경에서의 데이타 무결성 필요

‘ 클러스터 설정에서 전원 스위치의 사용은 모든 문제 환경에서 데이타의 보호를 보증합니다. 이런 장치들은 클러스터 시스템이 다른 클러스터 시스템이 오류 복구중 서비스를 재시작하기 전에 전원 사이클을 할수 있게 하여 줍니다. 전원 스위치는 반응을 하지 않던 (멈추었던) 시스템이 오류 복구후 서비스를 다시 시작하여 다시 반응을 할때 데이타에 문제가 생기지 않게 도와주며, 다른 클러스터 시스템으로 부터 I/O 명령을 받던 디스크에 다시 I/O를 발행합니다.

덧붙여서 , 만일 클러스터 시스템에서 quorum 데몬이 문제가 생길경우, 시스템이 더 이상 quorum 파티션을 감시할 수 없습니다. 만일 전원 스위치를 클러스터 상에서 사용하지 않을경우, 이런 에러 환경은 서비스가 하나 이상의 클러스터 시스템에서 작동되는 것을 초래할 수 있으며, 이를 통해 데이타상에 문제가 생길수 있습니다. 클러스터 상에서 전원 스위치 사용의 이익에 대해 좀 더 자세히 알고 싶으시면 2.4.2 절를 참조 하십시요. 제품 생산 환경과 같은 곳에서는 전원 스위치나 Watchdog timer를 설정상에 사용하는 것을 권장합니다.

2.1.1. 공유 저장 필요

클러스터의 작동은 신뢰성있고 잘 조화된 공유 저장장치의 사용에 좌우가 됩니다. 하드웨어에 문제가 생겼을 때, 다른 구성원에 피해를 주지 않고, 문제가 생긴 구성원을 공유 저장 장치의 사용을 막는것이 필요합니다. 저장 공유 장치는 클러스터의 구성에서 정말로 중요한 부분입니다.

많은 테스트들을 통해 안정성 있게 보통 SCSI 아답터를 사용해, multi-initiator 병렬 SCSI 설정에 데이타 양을 80MBytes/sec 으로 만든다는 것이 거의 불가능하지 않으면 힘들다는 것을 알게 됩니다. 더 심층적인 테스트들을 통해 이런 설정은 On-line 복구를 지원할 수 없다는 것을 알게 됩니다, 이유인 즉은 HBA 터미네이터들이 비활성화 되고 외부 터미네이터가 사용되었을때, 버스가 불안정적으로 작동하기 때문입니다. 이런 이유에서, 보편적인 아답터를 사용해 multi-initiator SCSI 설정을 하는 것은 지원하지 않습니다. Multi-port인 저장 장치에 연결된 Single-initiator 병렬 SCSI 버스나 광채널이 필요합니다.

Red Hat 클러스터 관리자 는 모든 클러스터 구성원이 공유 저장장치를 동시에 사용할수 있어야 합니다. 어떤 호스트 RAID 아답터는 공유 RAID 장치에 이런 기능을 사용할수 있도록합니다. 이런 제품들은 안정적인 작동을 위해 많은 테스트들이 행해져야 하며, 특히 만일 공유 RAID 장치가 병렬 SCSI 버스에 기반을 둔다면, 더 주의를 하여야 합니다. 이런 제품들은 간단하게 문제가 있는 시스템의 online 복구를 할수 없습니다. 현재 어떤 호스트 RAID 아답터들도 Red Hat Cluster Manager에 의해 인증되지 않았습니다. 자세한 현재 지원되는 하드웨어 관련 정보는 http://www.redhat.com을 참조하시기 바랍니다.

공유 저장 장치에 소프트웨어 RAID의 사용, 혹은 소프트웨어 Logical Volume Management (LVM)의 사용은 지원되지 않습니다. 이것은 이 기능들이 다중 호스트에서 공유 저장 장치를 사용하려는 요구를 제대로 처리하지 못하기 때문입니다. 소프트웨어 RAID나 LVM은 클러스터 구성원들중 공유하지 않을 저장 장치 (예를 들어, boot 그리고 시스템 파티션과 클러스터 서비스와 관련되지 않은 다른 파일 시스템) 에 사용될 수 있습니다.

2.1.2. 최소 하드웨어 필요

최소 하드웨어 필요에는 다음과 같은 클러스터의 작동에만 필요한 하드웨어 부품들만을 포함합니다:

- 2개의 서버들을 통해 클러스터 서비스 작동
- Heartbeat 채널과 클라이언트 네트워크 사용을 위한 이더넷
- 클러스터 Quorum 파트션과 서비스 데이타를 위한 공유 디스크 저장 장치

이런 유형의 하드웨어 설정은 2.1.5 절를 참조하시기 바랍니다.

최소 하드웨어 설정은 가장 경제적인 클러스터 설정입니다; 하지만, 이 설정에서는 여러 부분에서 문제가 발생할 수 있습니다. 예를 들어, 만일 RAID 컨트롤러에 문제가 생길 경우, 모든 클러스터 서비스가 사용 할 수 없게 될 것입니다. 또한 최소 하드웨어 설정을 사용할 경우, 소프트웨어 watchdog 타이머를 통해 데이타 무결성을 보장해야 합니다.

활용성을 높이기 위해서, 각 부품의 문제를 준비해야 하며, 또한 어떤 문제에서도 데이타 무결성을 보장하기 위해, 최소 설정을 좀 더 보강해야 합니다. 다음 표 2-1는 활용성을 개선시키며, 데이타 무결성을 보장하는 방법을 보여주고 있습니다:

문제	해결안
디스크 문제	다량의 디스크들에 데이타를 복제하도록 하드웨어 RAID를 설정합니다.

문제	해결안
RAID 컨트롤러 문제	벙럴 RAID 컨트롤러로 디스크의 데이타를 충분히 사용할수 있도록 합니다.
Heartbeat 채널 문제	클러스터 시스템안에 1:1 이더넷 혹은 벙럴 연결을 사용합니다.
전원 문제	충분한 UPS (Uninterruptible Power Supply)를 사용합니다.
모든 문제시 데이타의 문제	전원 스위치 혹은 하드웨어 기반의 watchdog 타이머를 사용합니다.

표 2-1. 활용성 개선및 데이타 무결성 보장하기

다음과 같은 부품으로 어떤 문제에도 데이타 무결성을 보장할 수 있는 '무적' 하드웨어 설정을 보장할 수 있습니다:

- 서버 2개로 클러스터 서비스를 동작중일 때
- 서로의 시스템간의 이더넷 연결을 통한 Heartbeat 채널과 클라이언트 네트워크 사용
- 이중-컨트롤러 RAID array를 통해 quorum 파티션과 서비스 데이타 복제
- 2개의 전원 스위치로 양 한쪽 클러스터 시스템에 문제가 생겼을 때 다른 클러스터 시스템에서 전원-사이클 가능케 함
- 충분한 Heartbeat 채널을 위해 클러스터 시스템끼리 1:1 이더넷 연결
- 충분한 Heartbeat 채널을 위해 클러스터 시스템끼리 1:1 벙럴 연결
- 고도의 전원 활용성을 위해 2개의 UPS 시스템

좀 더 자세한 하드웨어 설정을 위해서는 2.1.6 절 에 이런 유형의 하드웨어 예제가 나와 있습니다.

클러스터 하드웨어 설정에 다른 컴퓨터 사용에 보편적인 부가적인 하드웨어 부품을 추가할 수도 있습니다. 예를 들어, 클러스터 시스템에 네트워크 스위치 혹은 네트워크 헙을 추가하여, 클러스터 시스템에 네트워크 연결을 제공할 수도 있습니다. 클러스터 시스템에 또한 콘솔 스위치를 통해, 여러개의 시스템 관리를 실용적으로 할 수 있고, 개별적인 모니터, 마우스와 키보드의 필요를 줄일 수 있습니다.

콘솔 스위치의 한 종류로 터미널 서버가 있습니다. 이것은 벙럴 콘솔로 연결이 가능하며 한 곳에서 많은 시스템을 관리할 수 있습니다. 저렴한 가격의 방법중 하나로, 시스템 관리를 위해 GUI를 사용할 경우 용이한 *KVM* (키보드, 비디오, 마우스) 스위치가 있습니다.

클러스터 시스템을 선택할때, 주의 하실 것은 하드웨어 설정에 필요한 PCI 슬롯, 네트워크 슬롯, 그리고 벙럴 포트를 제공하는지 확인 하는 것입니다. 예를 들어, '무적' 설정은 여러개의 벙럴 그리고 네트워크 포트를 필요로 합니다. 더 좋게는, 최소한 벙럴 포트가 2개 이상인 클러스터 시스템을 선택하십시요. 더 자세한 내용은 2.2.1 절를 참조 하십시요.

2.1.3. 전원 컨트롤 유형 고르기

Red Hat 클러스터 관리자는 기본적인 전원 관리층과 장치기반의 모듈들로 이루어져 있으며, 이를 통해 다양한 전원 관리 유형을 폭넓게 사용할 수 있습니다. 클러스터에 사용할 알맞는 전원 콘트롤러를 선택할 때는 장치유형에 따른 문제들을 고려해야 합니다. 다음은 전원 스위치의 유형을 설명하며 그 뒤에 요약이 따릅니다. 데이타 무결성에 전원 스위치가 어떤 역할을 하는지 자세히 알기 위해서는 2.4.2 절를 참조 하십시요.

벙럴 연결과 네트워크 연결 전원 스위치는 서로 다른 장치로써, 한 클러스터 구성원이 다른 클러스터 구성원을 전원 사이클 할수 있도록 하여 줍니다. 이들은 벙럴 혹은 네트워크 케이블로 소프트웨어가 전기 콘센트에서 전원을 켜고 끌수 있도록 만드는 것입니다.

Watchdog timer는 문제가 생긴 시스템이 자신을 다른 시스템이 자신의 역할을 감당하기 전에, 다른 클러스터를 통해 전원 사이클 보다 괜찮은 방법으로 클러스터 상에서 제외 시킬수 있는 방법을 제공합니다. 보통 작동

모드에서는 지정된 시간이 있고 그 시간이 다 지나기전에 타이머를 재설정 해야합니다. 만일 클러스터 소프트웨어에 문제가 생겨 타이머를 재설정 하지 못할 경우, watchdog에서 현재 시스템이 멈추었든지 혹은 문제 발생으로 간주하게 됩니다. 문제가 없는 클러스터 구성원이 그 클러스터 구성원에 문제가 있다고 결정을 할때까지 어느 정도의 시간이 주어 집니다 (기본적으로 12초 입니다).즉 Watchdog 타이머의 시간이 이 시간보다 작아야 한다는 것입니다. 이를 통해, 문제가 없는 시스템이 문제가 있는 구성원의 서비스를 이양받기을 때에 완전하게 문제의 시스템이 클러스터내에 존재하지 않으며, 데이타 무결성에 대한 걱정을 줄일 수 있기 때문입니다. Watchdog의 지원은 리눅스 커널 단계에서 제공합니다. **Red Hat 클러스터 관리자**는 이 watchdog 기능을 보편적인 API들과 설정 방법을 따라 사용합니다.

Watchdog 타이머에는 두 종류가 있는데: 하드웨어 기반과 소프트웨어 기반입니다. 하드웨어 기반의 watchdog- 타이머는 Intel® i810 TCO 칩셋과 같은 시스템 보드 부품으로 이루어져 있습니다. 이런 구조는 주 시스템 보드의 CPU로 부터 고도의 독립성을 가지고 있습니다. 이 독립성은 시스템의 문제가 생겼을때 시스템의 재시작 스위치를 사용해 시스템을 재시작할수 있게 하여 탁월한 성능을 보여줍니다. 이런 watchdog 기능을 제공하는 PCI 확장 카드도 있습니다.

두번째 유형의 watchdog 타이머는 소프트웨어 기반입니다. 이 유형은 이 기능을 전담하는 하드웨어는 없습니다. 커널 쓰레드에서 가능케 하며, 매 정해진 시간 간격마다 작동하며, 타이머의 시간이 다 지나가면 시스템의 재시작을 시작합니다. 이 watchdog 타이머의 문제점은 어떤 문제, 예를 들어 interrupt가 막힌 상황에서 시스템이 멈추었을때, 커널 쓰레드가 불리울 수가 없게 되어 시스템 재시작을 할수 없는 경우가 발생합니다. 이로 인해, 이 같은 경우 이에 의지해 데이타 무결성을 보장할 수가 없습니다. 이는 또한 문제가 없는 클러스터 시스템이 멈추어있는 시스템의 서비스들을 이양받게 되면 어떤 경우에서는 이로인해 데이타에 또한 문제가 생길 수도 있습니다.

마지막으로, 관리자가 전원 콘트롤을 사용하지 않겠다 결정할 수도 있습니다. 만일 관리자가 "None" 유형을 선택할 경우, 기억할 것은 전원 사이클을 문제가 생긴 구성원에 할 수가 없게 된다는 것입니다. 또한 마찬가지로, 만일 문제가 생긴 구성원이 모든 문제시 재시작하지 않을 수도 있다는 것입니다. 클러스터에 전원 콘트롤을 사용하지 않는것은 실험단계에서는 유용하게 사용될수 있지만, 이것은 데이타 무결성이 중요하지 않을때나 사용하는 것으로, 제품 생산 환경에서는 사용않을 것을 권장합니다.

간단하게 어떤 유형의 전원 콘트롤을 사용할 지는 데이타 무결성의 필요성과 전원 스위치의 가격과 활용성을 비교해 보면 결정할 수 있습니다.

표 2-2 에 지원되는 전원 관리 모듈의 유형이 요약되어 있으며, 각각의 제품에 따라 좋은점과 나쁜점이 나열되어 있습니다.

유형	설명	좋은점	나쁜점
병렬 연결 전원 스위치	두개의 병렬 연결 전원 콘트롤이 클러스터에 사용 (구성원당 하나)	고도의 데이타 무결성 보장. 전원 콘트롤러 자체는 클러스터에 2개가 있기에 하나의 문제 발생점이 아님.	파워 콘트롤러 하드웨어와 케이블 구입 필요; 병렬 포트를 사용
네트워크 연결 전원 스위치	클러스터당 한개의 네트워크 연결 전원 콘트롤 필요	고도의 데이타 무결성 보장	전원 콘트롤 하드웨어 구입 필요. 전원 콘트롤러가 하나의 문제 발생점이 될수 있음 (하지만 보통 믿을수 있는 장치임.
하드웨어 Watchdog 타이머	고도의 데이타 무결성 보장	외부 전원 콘트롤러의 구입 방지	시스템에서 지원하는 watchdog 하드웨어 제공하지 않을수 있음
소프트웨어 Watchdog Timer	어느 정도의 데이타 무결성 보장	외부 전원 콘트롤러의 구입 방지; 어떤 시스템에서도 작동가능	어떤 문제 시나리오 따라, 소프트웨어 Watchdog 작동이 불가능 할 수 있음, 작은 취약점 제공

유형	설명	좋은점	나쁜점
No 전원 콘트롤러	No 전원 콘트롤러 기능이 사용됨	외부 전원 콘트롤러의 구입 방지; 어떤 시스템에서도 작동가능	어떤 문제 시나리오에서 데이타 문제에 노출되어 있음

표 2-2. 전원 스위치

2.1.4. 클러스터 하드웨어 표

다음과 같은 표를 클러스터 설정에 맞는 하드웨어 부품을 선택하는데 사용하십시요. 경우에 따라, 표에 제공된 제품들 중 일부는 클러스터 상에서 테스트가 되었지만, 다른 제품들도 클러스터에 사용할 수 있습니다.

시간이 지날면서 테스트된 클러스터 하드웨어 부품은 바뀝니다. 때문에, 밑에 있는 표가 부족 할 수 있습니다. 가장 최신의 지원되는 하드웨어 부품의 표를 위해서는 Red Hat 문서 사이트인 http://www.redhat.com/docs를 참조 하십시요.

하드웨어	양	설명	필요
클러스터 시스템	앙	Red Hat 클러스터 관리자는 IA-32 하드웨어 플랫폼을 지원합니다. 각 클러스터 시스템은 클러스터 하드웨어 설정에 알맞는 PCI 슬롯, 네트워크 슬롯, 그리고 벙럴 포트를 제공해야 합니다. 디스크 장치가 각 클러스터 시스템이 같게 지정 되어야 하기 때문에, 시스템에 symmetric I/O subsysterm을 갖는것을 권장합니다. 덧붙여서, 각 시스템은 최소한 450 MHz의 CPU 속도와 256 MB의 메모리가 필요합니다. 더 자세한 내용은 2.2.1 절를 참조 하십시요.	예

표 2-3. 클러스터 시스템 하드웨어 표

표 2-4 표에는 및가지의 다른 유형의 전원 스위치를 설명하고 있습니다. 다음에 나와 있는 전원 스위치는 클러스터당 하나만 필요합니다.

하드웨어	수	설명	필요
벙럴 전원 스위치	2개	전원 스위치는 각 클러스터 시스템이 다른 클러스터 시스템을 전원-사이클 할 수 있도록 해 줍니다. 클러스터 상에서의 전원 스위치 사용에 대해서는 2.4.2 절를 참조 하십시요. 주의할 것은, 클러스터에 벙럴이나 네트워크로 전원 스위치가 언걸이 가능하나 둘다를 사용할 수는 없습니다. 다음의 벙럴 전원 스위치는 완벅하게 테스트가 되었습니다: RPS-10 (미국내의 모델은 M/HD, 그리고 유럽내의 모델은 M/EC), http://www.wti.com/rps-10.htm에서 구입이 가능합니다. A.1.1 절를 참조 하십시요 다음과 같은 벙럴 전원 스위치는 잠복시간 (Latency)를 제공합니다. 이 스위치는 완벅하게 테스트되지 않았습니다: APC Serial On/Off Switch (partAP9211), http://www.apc.com	모든 문제 시나리오에 데이타 무걸성을 위해 강하게 추천합니다.

하드웨어	수	설명	필요
Null modem 케이블	2개	Null modem cables 은 클러스터 시스템의 병렬 포트와 병렬 전원 스위치를 연결하여 줍니다. 이 병렬 연결로 각 클러스터 시스템이 다른 시스템을 전원-사이클 할 수 있습니다. 멎멎 전원 스위치는 다른 케이블을 필요로 할 수 있습니다.	병렬 전원 스위치에만 필요
장착 브라켓	1개	멎멎 전원 스위치는 랙에 장착 할수 있도록 되어 있으며, 특별한 브라켓이 필요합니다.(e.g. RPS-10).	랙에 장착 가능한 전원 스위치에만 사용
네트워크 전원 스위치	하나	네트워크 연결 파워 스위치는 각 클러스터 구성원이 다른 구성원을 전원 사이클 할 수 있도록 하여 줍니다. 네트워크 연결 전원 스위치에 대한 정보나 이에 관련된 경고는 2.4.2 절를 참조 하십시요. 다음의 네트워크 연결 전원 스위치 완벽하게 테스트 되었습니다: · WTI NPS-115, 혹은 NPS-230, http://www.wti.com에서 구입이 가능합니다. 기억할 것은 nps 전원 스위치는 이중 redundant 파워 서플라이와 사용이 가능합니다. A.1.2 절를 참조 하십시요. · Baytech RPC-3 와 RPC-5, http://www.baytech.net APC 마스터 스위치를 위해 Latent 지원이 가능합니다 (AP9211, or AP9212), www.apc.com	모든 문제 상황에서의 데이타 무결성을 위해 강하게 권장.
Watchdog Timer	2개	Watchdog 타이머는 문제가 생긴 클러스터 구성원이 자신을 다른 시스템이 자신의 서비스를 이양 받기 전에 클러스터 상에서 제외 시킬수 있도록 합니다. 자세한 내용은 2.4.2 절를 참조 하십시요.	Watchdog 하드웨어가 제공되는 시스템의 데이타 무결성을 위해 권장합니다.

표 2-4. 전원 스위치 하드웨어 표

다음의 표는 관리자가 선택할수 있는 다양한 저장 장치를 제공합니다. 한 클러스터 상에 다음의 모든 부품이 필요치 않습니다.

하드웨어	양	설명	필요

하드웨어	양	설명	필요
외장 디스크 저장 인클로저	1개	광채널 혹은 single-initiator 벙럴 SCSI를 사용하여 클러스터를 single 혹은 dual 콘트롤러 RAID array에 연결합니다. single-initiator 버스를 사용하려면, RAID 콘트롤러가 다중 호스트 포트를 가지고 있어야하며, 호스트 포트에 연결된 logical 유닛에 동시 사용이 가능해야 합니다. Dual 콘트롤러 RAID array를 사용하려면, logical 유닛이 한 콘트롤러에서 다른 시스템으로 운영체제에 투명하게 문제 복구가 가능해야 합니다. 다음은 권장된 SCSI RAID array로 모든 포트에 있는 logical unit에 동시 사용을 가능하게 해줍니다 (이 표는 완전한 표가 아닙니다: 다만 테스트된 RAID 박스에 한해 유효합니다): · Winchester Systems FlashDisk RAID Disk Array, http://www.winsys.com에서 구입 가능 · Dot Hill's SANnet Storage Systems, http://www.dothill.com 에서 구입 가능 · Silicon Image CRD-7040 & CRA-7040, CRD -7220, CRD-7240 & CRA-7240, CRD-7400 & CRA-7400 콘트롤러 기반 RAID arrays. http://www.synetexinc.com 에서 구입 가능 장치 ID와 LUN의 대칭 확실히 하기 위해서 많은 Dual Redundant 기능을 가진 RAID array는 active/passive 모드가 지정되야 합니다. 자세한 것은 2.4.4 절를 참조하십시요.	네
호스트 버스 아답터	2개	공유 디스크 저장 장치를 연결하려면, 벙럴 SCSI 혹은 광채널 호스트 버스 아답터를 각 PCI 슬롯에 설치 해야 합니다. 벙럴 SCSI는, 저 전압 차동장치 (LVD) 호스트 버스 아답터를 사용하십시요. 아답터가 HD68 혹은 VHDCI 연결을 가지고 있습니다. 권장된 SCSI 호스트 버스 아답터들은 다음과 같습니다: · Adaptec 2940U2W, 29160, 29160LP, 39160, 그리고 3950U2 · Intel L440GX+에 장착된 Adaptec AIC-7896 · Qlogic QLA1080 그리고 QLA12160 · Tekram Ultra2 DC-390U2W · LSI Logic SYM22915 · 권장되는 광 채널 호스트 버스 아답터는 Qlogic QLA2200. 장치의 기능과 설정 관련 정보는 A.6 절를 참조 하십시요. 호스트-버스 기반의 RAID카드는 만일 multi-host를 확실히 지원한다면 지원합니다. 이 책이 출판된 때에는, 완벽하게 테스트된 호스트-버스 아답터 기반의 RAID카드가 없었습니다. 좀 더 최신 정보는 http://www.redhat.com를 참조 하십시요.	예
SCSI 케이블	2개	68핀을 가진 SCSI 케이블이 호스트 버스 아답터와 저장 장치 인클로저 포트를 연결하기 위해 필요합니다. 케이블은 HD68 혹은 VHDCI 연결을 가지고 있습니다. 아답터 유형에 따라 케이블이 바뀔수 있습니다.	벙럴 SCSI 설정만을 위함

하드웨어	양	설명	필요
SCSI 종료 시스템	2개	"out" 포트를 사용하는 RAID 저장 장치 인클로저 (예를 들어 FlashDisk RAID Disk Array)와 이것이 single-initiator SCSI 버스에 연결되었을때, 버스를 멈춤 시키기 위해 종료 시스템을 "out"포트에 연결하는 것이 필요합니다.	병렬 SCSO 설정 또한 종료할 필요가 있다면
광채널 헙 혹은 스위치	1개 혹은 2개	광 채널 헙 혹은 스위치 필요	몇몇의 광 채널 설정을 위해
광 채널 케이블	2개 혹은 6개	광 채널 케이블은 호스트 버스 아답터를 저장 장치 인클로저 포트와 연결시켜 주며, 광 채널 헙 혹은 광 채널 스위치로 연결시켜 줍니다. 만일 업 혹은 스위치가 사용되었을 경우, 헙이나 스위치를 저장 장치의 아답터에 연결하기 위해 케이블이 더 필요할 수도 있습니다.	광 채널 설정만을 위함

표 2-5. 공유 디스크 저장 하드웨어 표

하드웨어	양	설명	필요
네트워크 인터페이스	각 네트워크 연결에 한개	클러스터 시스템에서 각 네트워크 연결에 한개의 네트워크 인터페이스가 설치되어 있어야 합니다.	네
네트워크 스위치 혹은 헙	1개	네크워크 스위치 또는 헙을 통하여 여러개의 시스템을 네트워크에 설치할 수 있습니다.	아니요
네트워크 케이블	각 네트워크 인터페이스에 1개	보통 네트워크 케이블, 예를 들어 RJ45 연결이 있는 케이블, 네트워크 인터페이스를 네트워크 스위치 혹은 네트워크 헙에 연결.	네

표 2-6. 네트워크 하드웨어 표

하드웨어	양	설명	필요
네트워크 인터페이스	각 채널에 2개	각 이더넷 heartbeat 채널은 양 클러스터 시스템에 하나씩의 네트워크 인터페이스가 설치되어 있어야 합니다.	아니요
네크워크 크로스오버 케이블	각 채널에 하나	네트워크 크로스오버 케이블은 한 클러스터 시스템의 네트워크 인터페이스와 다른 클러스터 시스템상의 네트워크 인터페이스를 연결하여, 이더넷 Heartbeat채널을 만듭니다.	Redundant 이더넷 Heartbeat 채널에 1개

표 2-7. 1:1 이더넷 Heartbeat 채널 하드웨어 테이블

하드웨어	양	설명	필요

하드웨어	양	설명	필요
병렬 카드	병렬 채널당 2개	각 병렬 heartbeat 채널은 양쪽 클러스터 시스템에 1개씩의 병렬 포트가 필요합니다. 현재의 병렬포트의 한계를 증가 시키기 위해서는 다-포트 병렬 PCI 카드를 사용할 수 있습니다. 다음은 권장하는 다-포트 카드들 입니다: Vision Systems VScom 200H PCI card, 2개의 병렬포트 제공, http://www.vscom.de에서 구입가능 Cyclades-4YoPCI+ card, 4개의 병렬포트 제공, http://www.cyclades.com에서 구입 가능. 기억할 것은 병렬 Heartbeat 설정은 부수적인 것이기 때문에, 이 이유를 위해서 부가적인 하드웨어를 투자하는 것은 바람직 하지 못합니다. 만일 미래에 2개 이상의 구성원을 가지게 된다면, 병렬 Heartbeat채널의 지원은 반대합니다.	아니요
Null modem 케이블	각 채널에 하나	Null 모뎀 케이블은 한 클러스터 시스템의 병렬 포트와 다른 클러스터 시스템의 상응하는 병렬 포트를 연결하여, 병렬 Heartbeat 채널을 구성합니다.	병렬 Heartbeat 채널만을 위함

표 2-8. 1:1 병렬 Heartbeat 채널 하드웨어 표

하드웨어	양	설명	필요
터미널 서버	1개	터미널 서버는 사용자를 통해 많은 시스템을 한 원거리에서 관리할 수 있도록 해줍니다.	아니요
KVM	1개	KVM은 사용자를 통해 여러 시스템이 1개의 키보드, 모니터와 마우스를 공유할 수 있도록 하여 줍니다. KVM에 따라 스위치와 시스템을 연결하는 케이블이 다를 수도 있습니다.	아니요

표 2-9. 콘솔 스위치 하드웨어 표

하드웨어	양	설명	필요
UPS 시스템	1개 혹은 2개	*Uninterruptible power supply* (UPS)는 시스템을 전원에 문제가 생겼을때로 부터 보호합니다. 클러스터의 동작에 UPS는 꼭 필요합니다. 알맞게는 공유 저장 장치 인클로져의 전원 케이블과 양쪽의 전원 스위치의 케이블을 UPS 시스템에 연결하는 것이 좋습니다. 덧붙여서, UPS 시스템은 적당한 시간동안 전력을 공급할 수 있어야 하며, 자기만의 회로에 연결되어야 합니다. 권장하는 UPS 시스템은 APC Smart-UPS 1400랙 마운트로, http://www.apc.com에서 구입 가능합니다.	활용성을 위해 강하게 추천합니다.

표 2-10. UPS 시스템 하드웨어 표

2.1.5. 최소 클러스터 설정의 예제

표 2-11에 명시된 하드웨어 부품으로 최소 클러스터 설정을 설치할 수 있습니다. 이 설정은 모든 문제시 데이타의 무결성을 보장하지 못합니다, 이유는 전원 스위치를 포함하지 않았기 때문입니다. 기억할 것은, 이것은 간단한 예제 설정입니다; 다른 하드웨어를 이용해서 최소 설정을 설정할 수 있습니다.

하드웨어	양
서버 2개	각 클러스터 시스템은 다음과 같은 하드웨어를 포함합니다: 이더넷 Heartbeat채널과 클라이언트 사용을 위한 네트워크 인터페이스 공유 저장 장치의 연결을 위한 1개의 Adaptec 29160 SCSI adapter (termination disabled)
RJ45 연결이 있는 2개의 네트워크 케이블	네트워크 케이블은 각 클러스터 시스템을 클라이언트 사용과 이더넷 Heartbeat를 위해 네트워크에 연결 시킵니다.
RAID 저장 장치 인클로저	최소 2개의 호스트 포트를 가진 RAID 저장 장치 인클로저.
2개의 HD68 SCSI 케이블	각 케이블은 한개의 HBA를 RAID 컨트롤러에 연결하여, 2개의 single-initiator SCSI 버스를 제공합니다.

표 2-11. 최소 클러스터 하드웨어 설정 구성

2.1.6. '무적' 설정의 예제

표 2-12에 명시된 구성을 통해 2개의 single-initiator SCSI 버스와 데이타 무결성을 보장할 전원 스위치를 가진 '무적'설정을 설정할 수 있습니다. 기억할 것은 이것은 예제 설정이라는 것입니다; 다른 하드웨어를 사용해서 '무적' 설정을 설정 할수 있습니다.

하드웨어	수
2개의 서버	각 클러스터 시스템은 다음과 같은 하드웨어를 포함합니다: 다음을 위한 2개의 네트워크 인터페이스: 1:1 이더넷 Heartbeat 채널 클라이언트의 네트워크 사용과 이더넷 Heartbeat 연결 다음을 위한 3개의 병렬 포트: 1:1 병렬 Heartbeat 채널 원격 전원 스위치 연결 터미널 서버로의 연결 공유 디스크 저장 장치와의 연결을 위한 1개의 Tekram Ultra2 DC-390U2W adapter (termination enabled)
1개의 네트워크 스위치	네크워크 스위치로 여러개의 시스템이 네트워크에 연결할 수 있습니다.
1개의 Cyclades 터미널 서버	터미널 서버를 통해, 한곳에서 떨어져 있는 여러 시스템을 관리할 수 있습니다. (터미널 서버는 클러스터의 동작에 필요하지 않습니다.)
3개의 네트워크 케이블	네트워크 케이블로, 터미널 서버와 각 클러스터 시스템에 연결되어 있는 네트워크 인터페이스를 네트워크 스위치에 연결할 수 있습니다.
2개의 RJ45에서 DB9으로의 크로스오버 케이블	RJ45에서 DB9으로의 크로스오버 케이블로 각 클러스터의 병렬 포트를 Cyclades 터미널 서버로 연결할 수 있습니다.

하드웨어	수
1개의 네트워크 크로스오버 케이블	네트워크 크로스오버 케이블은 한 클러스터 시스템의 네트워크 인터페이스와 다른 클러스터 시스템의 인터페이스를 연결하여 1:1 이더넷 Heartbeat 채널을 구축합니다.
2개의 RPS-10 전원 스위치	전원 스위치들은 각 클러스터 시스템이 다른 시스템들의 서비스를 재시작 하기 전에, 전원-사이클 시킬수 있습니다. 각 클러스터의 전원 케이블은 각자의 전원 스위치에 연결됩니다.
3개의 null modem 케이블	Null modem 케이블로 각 클러스터 시스템의 병렬 포트를 전원 스위치에 연결시켜서 다른 클러스터 시스템에 전원을 공급합니다. 이 연결로 각 클러스터 시스템에 다른 시스템에 전원-사이클 시킬 수도 있습니다. Null 모뎀 케이블로 한 클러스터 시스템의 병렬포트와 다른 시스템에 상응되는 병렬포트를 연결하여, 1:1 병렬 Heartbeat 채널을 만들수 있습니다.
2중 컨트롤러를 가진 FlashDisk RAID 디스크 Array	2중 RAID 컨트롤로는 디스크와 컨트롤러 문제에서 시스템들을 보호합니다. 이 RAID 컨트롤러는 호스트 포트에 있는 모든 logical 유닛에게 동시 사용을 가능케 합니다.
2개의 HD68 SCSI 케이블	HD68 케이블은 각 호스트 버스 아답터를 RAID 인클로저의 "in" 포트와 연결 시켜, 2개의 single-initiator SCSI 버스를 구성합니다.
2개의 terminators	Terminator는 각 RAID 인클로저의 "out" 포트에 연결되어 모든 single-initiator SCSI 버스를 멈춥니다.
Redundant UPS 시스템	UPS 시스템은 고차원의 전원을 공급합니다. 전원 스위치의 전원 케이블과 RAID 인클로져는 2개의 UPS 시스템에 연결됩니다.

표 2-12. 무적 설정 구성 표

그림 2-1 는 위의 표에서 명시된 '무적' 하드웨어 설정을 보여줍니다, 2개의 single-initiator SCSI 버스들, 그리고 데이타 무결성을 보장할 전원 스위치. 동그라미 안에 "T"라고 써있는 것은 SCSI Terminator를 나타냅니다.

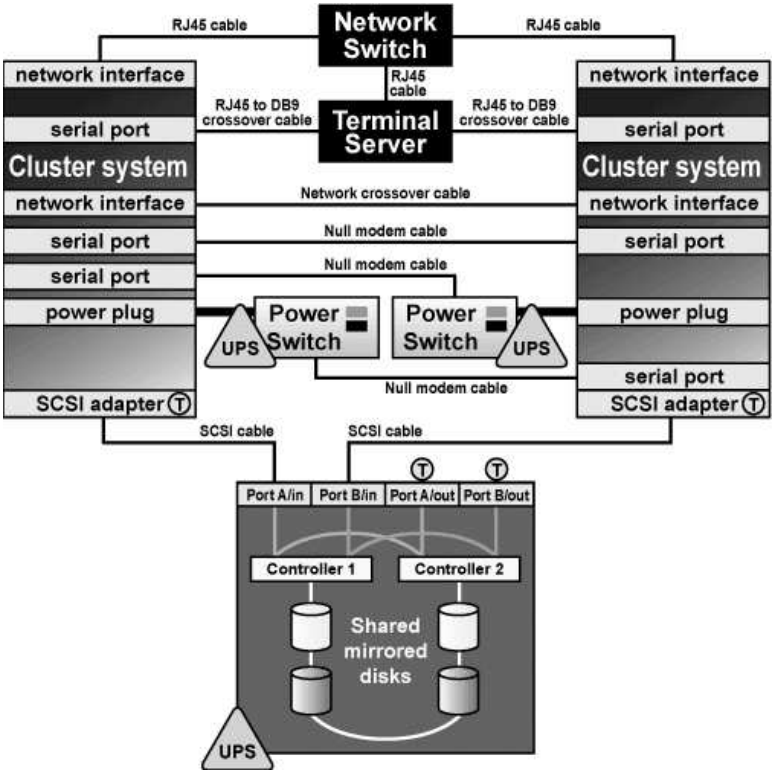

그림 2-1. No-Single-Point-Of-Failure 설정 예시

2.2. 클러스터 시스템을 설정하기 위한 절차

2.1 절에 표시된 것 처럼 하드웨어 구성을 선택한 후, 간단한 클러스터 시스템 하드웨어를 설정하고, 부가적으로 콘솔 스위치, 네트워크 스위치 또는 힙을 시스템에 연결합니다. 이것을 위해 다음을 따릅니다:

1. 양쪽 클러스터 시스템에, 필요한 네트워크 아답터, 벙럴 카드, 호스트 버스 아답터를 설치합니다. 이 일을 하기 위해 2.2.1 절를 참조하십시요.
2. 부수적인 콘솔 스위치를 설치하고 양 클러스터 시스템을 연결 하십시요. 이 일을 하기 위해 2.2.2 절를 참조 하십시요.

   만일 콘솔 스위치가 사용되지 않았다면, 콘솔 터미널에 양 시스템을 연결하십시요.
3. 부수적인 네트워크 스위치 또는 힙을 설치하고 보통 네트워크 케이블로 시스템과 터미널 서버 (만일 존재한다면)로 연결하십시요. 이 일을 하기 위해 2.2.3 절를 참조 하십시요.

만일 네트워크 스위치 혹은 헙이 사용되지 않았다면, 보통 네트워크 케이블이 시스템들과 터미널 서버 (만일 존재 한다면)로 연결하는데 사용해서는 안됩니다.

위의 일들을 끝낸 후에, 2.3 절에 설명된것 같이 리눅스 배포판을 설치 하십시요.

2.2.1. 기본 시스템 하드웨어 설치하기

클러스터 시스템은 응용프로그램이 필요로하는 CPU 프로세싱 과 메모리를 뒷받침해야 합니다. 권장하는 바로는 최소한 450 MHz의 CPU 속도와 256MB의 메모리가 필요합니다.

덧붙여, 클러스터 시스템은 SCSI 혹은 FC 아답터를 사용할 수 있어야 하며, 하드웨어 설정에 필요한 병렬 포트를 가져야 합니다. 시스템에는 한정된 기본 내장 병렬, 네트워크 포트와 PCI 확장 슬롯이 있습니다. 다음 표는 클러스터 시스템에 어느 정도의 용량이 필요한 지를 아는데 도움이 될 것입니다:

클러스터 하드웨어 구성	병렬 포트	네트워크 슬롯	PCI 슬롯
원격 전원 스위치 연결 (부수적, 강하게 권장)	1개		
공유 디스크 저장 장치를 위한 SCSI 혹은 광 채널 아답터			버스 아답터 마다 1개
클라이언트 사용과 이더넷 Heartbeat를 위한 네트워크 연결		네트워크 연결당 1개	
1:1 이더넷 Heartbeat 채널 (부수적)		각 채널당 1개	
1:1 병렬 Heartbeat 채널 (부수적)	각 채널당 1개		
터미널 서버 연결 (부수적)	1개		

표 2-13. 기본 시스템 하드웨어 설치하기

대부분의 시스템은 최소한 하나의 병렬 포트를 가지고 있습니다. 이상적으로, 최소한 2개의 병렬 포트를 가진 시스템을 선택하십시요. 만일 시스템이 크래픽 출력 능력을 가지고있다면, 병렬 콘솔 포트를 병렬 Heartbeat 채널 혹은 전원 스위치 연결에 사용할 수 있습니다. 병렬 포트를 확장하려면, 다-포트 병렬 PCI 카드를 사용하십시요.

덧붙여서, 로컬 시스템 디스크가 공유 디스크와 같은 SCSI 버스를 사용하지 않도록 하십시요. 예를 들어 two-channel SCSI 아답터, 예를 들어 Adaptec 39160 시리즈 카드 에서, 한 채널에 내부 장치를 또 다른 채널에 공유 디스크를 연결하지 마십시요. 가능 하다면 여러개의 SCSI카드를 사용하십시요.

더 자세한 설치 관련 자료는 제조 회사에서 제공한 시스템 문서를 참고 하십시요. 호스트 버스 아답터를 클러스터 상에서 사용하는 것에 관련 하드웨어 관련 자료는 부록 A를 참조 하십시요.

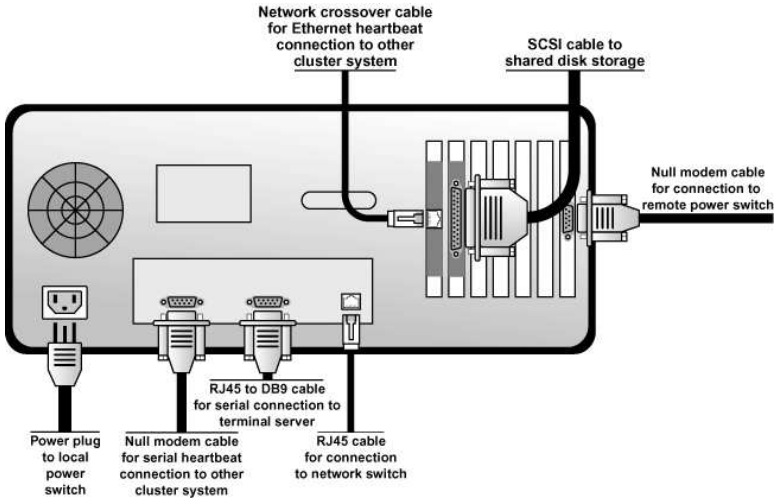

그림 2-2. 보편적인 클러스터 시스템의 외부 케이블

그림 2-2 는 예제 클러스터 시스템의 측면과 보편적인 클러스터 설정에서의 외부 케이블을 보여줍니다.

2.2.2. 콘솔 스위치 설치하기

콘솔 스위치가 클러스터의 작동에는 필요하지 않지만, 클러스터 시스템의 손쉬운 관리와 각 클러스터 시스템을 위한 여러개의 모니터, 마우스, 키보드의 필요를 줄일 수 있습니다. 여러 유형의 콘솔 스위치가 있습니다.

예를 들어, 터미널 서버는 병렬 콘솔과 원거리에서 많은 시스템을 관리하는 것을 연결하여 줍니다. 적은 비용을 위해서는, 여러 시스템이 키보드, 마우스 그리고 모니터를 공유할 수 있는 KVM (Keyboard, monitor 그리고 mouse) 스위치를 사용할 수 있습니다. KVM 스위치는 GUI를 사용하여 시스템 관리를 할때 사용 될 수 있습니다.

제조사에서 제공된 문서에 따라 콘솔 스위치를 설치합니다.

콘솔 스위치가 설치되 후, 각 클러스터 시스템과 연결합니다. 콘솔 스위치에 따라 이때 사용되는 케이블이 다를 수 있습니다. 만일 Cyclades 터미널 서버가 사용되었다면, RJ45를 DB9으로 바꿔주는 크로스오버 케이블이 각 클러스터 시스템의 병렬 포트와 터미널 서버에 연결됩니다.

2.2.3. 네트워크 스위치나 허브(hub) 설정

네트워크 스위치나 허브(hug)는 클러스터 작업에 필요하지는 않지만, 클러스터와 클라이언트 시스템 간의 네트워크 작업을 용이하게 해줍니다.

제조업체에서 제공한 문서에 나온 지시 사항을 따라서 네트워크 스위치나 허브를 설정하시기 바랍니다.

네트워크 스위치나 허브를 설정하신 후, 일반 네트워크 케이블을 사용하여 각 클러스터 시스템에 연결하십시오. 터미널 서버를 사용하시는 경우, 네트워크 케이블은 터미널 서버를 네트워크 스위치나 허브에 연결합니다.

2.3. Red Hat Linux 배포판 설치와 설정 단계

기본 시스템 하드웨어를 설정하신 후, 양 클러스터 시스템 상에 Red Hat Linux를 설치를 진행하신 후 연결된 장치들이 인식되는지 확인해 보십시오. 다음과 같은 절차를 따르십시오:

1. 양 클러스터 시스템에서 Red Hat Linux 배포판을 설치하시기 바랍니다. 커널을 사용자 정의하신다면, 2.3.1 절 커널 요건과 지시 사항을 따르셔야 합니다.
2. 클러스터 시스템을 재부팅하십시오.
3. 터미널 서버를 사용하시는 경우, 콘솔 포트로 콘솔 메시지를 보내도록 리눅스를 설정하시기 바랍니다.
4. 각 클러스터 시스템에서 `/etc/hosts` 파일을 편집하신 후 클러스터에 사용된 IP 주소를 추가하십시오. 이 작업을 수행하는 방법에 대한 보다 많은 정보는 2.3.2 절을 참조하시기 바랍니다.
5. 클러스터 시스템의 부팅 시간을 줄이기 위해 대체 커널 부트 타임아웃 제한을 줄여야 합니다. 이 작업 수행 방법에 대한 보다 많은 정보를 원하신다면, 2.3.3 절을 참조하시기 바랍니다.
6. 시리얼 heartbeat 채널이나 원격 전원 스위치 연결에 사용된 시리얼 포트와 관련된 로그인 (또는 `getty`) 프로그램들이 사용되고 있지 않음을 확인해 주십시오. 확인을 위해, `/etc/inittab` 파일에서 시리얼 채널과 원격 전원 스위치에 사용되는 시리얼 포트에 상응하는 엔트리를 우물정자 표시 (#)를 사용하여 주석화 하시면 됩니다. 그 후 `init q` 명령을 입력하시기 바랍니다.
7. 양 시스템은 설치된 모든 하드웨어를 인식하는지 확인해 주십시오:
   - 콘솔 시작 메시지를 보기위해 `dmesg` 명령을 사용하시기 바랍니다. 이 작업 수행 방법에 대한 보다 많은 정보는 2.3.4 절을 참조하시기 바랍니다.
   - 커널에서 설정된 장치를 보기 위해 `cat /proc/devices` 명령을 사용하십시오. 이 작업 수행 방법에 대한 보다 많은 정보는 2.3.5 절을 참조하시기 바랍니다.
8. `ping` 명령을 사용하여 한 시스템에서 다른 시스템으로 테스트 패킷을 보내어, 클러스터 시스템이 모든 네트워크 인터페이스를 통해 통신할 수 있는지 확인하십시오.
9. Samba 서비스를 설정하시려면, Samba 관련 **RPM** 패키지가 여러분의 시스템에 설치되어 있는지 확인해 보십시오.

2.3.1. 커널 요건

커널을 직접 설정하실 경우, 다음과 같은 커널 요건을 따르셔야 합니다:

- `CONFIG_IP_ALIAS` 커널 옵션을 `y`로 설정하여 커널 내에서 IP 별칭(aliasing) 지원을 활성화 하시기 바랍니다. 커널 옵션을 지정하실 때, `Networking Options`에서 `IP aliasing support` 옵션을 선택하시면 됩니다.
- `CONFIG_PROC_FS` 커널 옵션을 `y`로 설정하여 `/proc` 파일 시스템에 대한 지원을 활성화 하시기 바랍니다. 커널 옵션을 지정하실 때, `Filesystems`에서 `/proc filesystem support` 옵션을 선택하시면 됩니다.
- 클러스터 소프트웨어가 시작하기 전에 SCSI 드라이버가 시작된 것을 확인하시기 바랍니다. 예를 들어, 클러스터 스크립트가 시작하기 전에 드라이버가 시작되도록 시작 스크립트를 편집하시면 됩니다. 또한 `/etc/modules.conf` 파일을 편집하여 SCSI 드라이버를 로드 가능한 모듈로 포함하는 대신, 커널에 정적으로 SCSI 드라이버를 내장시키는 것도 가능합니다.

또한 Linux 배포판을 설치하실 때, 다음과 작업을 수행하시기를 적극 권장합니다:

- 리눅스를 설치하시기 전에 클러스터 시스템과 point-to-point 이더넷 heartbeat 인터페이스의 IP 주소 정보를 알아내시기 바랍니다. point-to-point 이더넷 인터페이스의 IP 주소는 사실 IP 주소 (예, 10.*x.x.x*)일 수도 있습니다.

- 선택 사항으로서, IP 주소를 "클러스터 별칭"으로 사용될 수 있도록 보존하시기 바랍니다. 이 주소는 일반적으로 원격 모니터링에 사용됩니다.
- 다음과 같은 리눅스 커널 옵션을 활성화하여 시스템 설정과 이벤트에 대한 자세한 정보를 보여주어, 여러분이 보다 쉽게 문제를 진단할 수 있게 합니다:
  - `CONFIG_SCSI_LOGGING` 커널 옵션을 `y`로 설정하여 SCSI 로깅 지원을 활성화 합니다. 커널 옵션을 지정하실 때, `SCSI Support`에서 `SCSI logging facility` 옵션을 선택하시면 됩니다.
  - `CONFIG_SYSCTL` 커널 옵션을 `y`로 설정하여 `sysctl` 지원을 활성화 합니다. 커널 옵션을 지정하실 때, `General Setup`에서 `Sysctl support` 옵션을 선택하십시오.

- `/`, `/etc`, `/tmp`, `/var`와 같은 로컬 파일 시스템을 공유 디스크나 동일한 SCSI 버스에서 공유 디스크로 놓지 마십시오. 만일 그렇지 않으면, 다른 클러스터 구성원이 이 파일 시스템을 실수로 마운팅하고, 클러스터 디스크에 사용되는 한 개의 버스에 제한된 숫자의 SCSI ID 번호를 보존하는 문제를 초래하게 됩니다.
- `/tmp`와 `/var`를 다른 파일 시스템에 놓으십시오. 시스템 성능이 호전됩니다.
- 클러스터 시스템이 부팅시, 시스템이 디스크 장치를 리눅스 설치 과정에서 검색되었던 순서대로 검색하는지 확인하시기 바랍니다. 만일 장치가 동일한 순서로 검색되지 않는다면, 시스템 부팅에 실패할 가능성이 있습니다.
- 0 보다 큰 논리 유닛 번호 (LUN)를 사용하여 설정된 RAID 저장 장치를 사용하실 경우, `/etc/modules.conf` 파일에 다음과 같은 부분을 첨가하여 LUN 지원을 활성화 하셔야 합니다:
  `options scsi_mod max_scsi_luns=255`

  `modules.conf` 파일을 수정하신 후, `mkinitrd`를 사용하여 초기 ram 디스크를 재구축해야 합니다. `mkinitrd`를 사용하여 ramdisk를 생성하는 방법에 대한 보다 많은 정보는 공식 Red Hat Linux 사용자 정의 가이드를 참조하시기 바랍니다.

2.3.2. `/etc/hosts` 파일 편집하기

`/etc/hosts` 파일은 IP 주소에서 호스트명으로 번역 테이블을 포함하고 있습니다. 각 클러스터 시스템에 위치한 `/etc/hosts` 파일에는 다음과 같은 항목들이 포함됩니다:

- 양 클러스터 시스템의 IP 주소와 관련 호스트명
- point-to-point 이더넷 heartbeat 연결에 사용된 IP 주소와 관련 호스트명 (사설 IP 주소일 수도 있습니다)

`/etc/hosts` 파일의 대체로서, DNS 또는 NIS와 같은 이름 부여 서비스 (naming service)를 사용하여 클러스터에 의해 사용되는 호스트명을 지정할 수 있습니다. 그러나 의존성을 줄이고 가용성의 최적화하기 위하여, `/etc/hosts` 파일을 사용하여 클러스터 네트워크 인터페이스의 IP 주소를 정의하시길 적극 추천합니다.

다음은 클러스터 시스템 상에서 `/etc/hosts` 파일의 예시입니다:

```
127.0.0.1    localhost.localdomain  localhost
193.186.1.81   cluster2.yourdomain.com   cluster2
10.0.0.1     ecluster2.yourdomain.com  ecluster2
193.186.1.82   cluster3.yourdomain.com   cluster3
10.0.0.2     ecluster3.yourdomain.com  ecluster3
193.186.1.83   clusteralias.yourdomain.com clusteralias
```

앞의 예시에서는 두 개의 클러스터 시스템 (*cluster2*와 *cluster3*)에 사용되는 IP 주소와 호스트명, 그리고 원격 클러스터 모니터링에 사용된 IP 별칭 *clusteralias*를 비롯하여 각 클러스터 시스템 (*ecluster2*와 *ecluster3*) 상의 point-to-point heartbeat 연결에 사용된 이더넷 인터페이스의 사설 IP 주소와 호스트명을 보여줍니다.

`/etc/hosts` 파일에서 로컬 호스트 항목에서 로컬이 아닌 시스템이 포함되지 않도록 로컬 호스트 항목이 올바른 포맷으로 구성되어 있는지 확인해 주십시오. 로컬이 아닌 시스템 (*server1*)이 포함된 잘못된 로컬 호스트 항목의 예시는 다음과 같습니다:

```
127.0.0.1  localhost.localdomain  localhost server1
```

만일 포맷이 올바르지 않다면 heartbeat 채널이 적절하게 작동하지 못하게 됩니다. 예를 들어, 채널이 오프라인 *(offline)*으로 잘못 표시될 수도 있습니다. `/etc/hosts` 파일을 확인하신 후 필요하다면, 로컬 호스트 항목에서 로컬이 아닌 시스템을 삭제하여 파일 포맷을 정정하시기 바랍니다.

각 네트워크 어댑터는 적절한 IP 주소와 넷마스크를 사용하여 설정되어야만 한다는 점에 유의해 주십시오.

다음은 클러스터 시스템에서 `/sbin/ifconfig` 명령의 출력 결과의 일부 예시입니다:

```
# ifconfig

eth0   Link encap:Ethernet HWaddr 00:00:BC:11:76:93
     inet addr:192.186.1.81 Bcast:192.186.1.245 Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
     RX packets:65508254 errors:225 dropped:0 overruns:2 frame:0
     TX packets:40364135 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:100
     Interrupt:19 Base address:0xfce0


eth1   Link encap:Ethernet HWaddr 00:00:BC:11:76:92
     inet addr:10.0.0.1 Bcast:10.0.0.245 Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
     RX packets:0 errors:0 dropped:0 overruns:0 frame:0
     TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:100
     Interrupt:18 Base address:0xfcc0
```

위의 예시는 한 개의 클러스터 상에서 두 개의 네트워크 인터페이스를 보여줍니다; 클러스터 시스템의 *eth0* 네트워크 인터페이스와 *eth1* (point-to-point heartbeat 연결에 사용된 네트워크 인터페이스).

2.3.3. 커널 부트 타임아웃 제한 줄이기

커널 부트 타임아웃 제한을 줄임으로서 클러스터 시스템의 부팅 시간을 줄일 수 있습니다. 리눅스 부팅 순서에서 부트로더에 대체 커널을 부팅하도록 지정하는 것이 가능합니다. 커널에 지정된 디폴트 타임아웃 제한은 10초 입니다.

클러스트 시스템의 커널 부트 타임아웃 제한을 수정하시려면, `/etc/lilo.conf` 파일에서 *`timeout`* 매개변수의 값을 원하시는 값으로 지정하시면 됩니다. 다음 예시에서는 타임아웃 제한을 3초로 설정합니다:

```
timeout = 30
```

`/etc/lilo.conf` 파일에 대한 변경 사항을 적용하시려면, `/sbin/lilo` 명령을 실행하시기 바랍니다.

앞에서 설명된 것과 유사하게, `grub` 부트로더 사용시, `/boot/grub/grub.conf` 파일에서 타임아웃 매개변수를 적절한 값으로 지정해 주십시오. 타임아웃 하기전 시간 간격을 3초로 설정하시려면, 다음과 같이 변수를 편집하시면 됩니다:

```
timeout = 3
```

2.3.4. 콘솔 시작 메시지 표시

`dmesg` 명령을 사용하여 콘솔 시작 메시지를 표시합니다. 보다 많은 정보를 원하시면, dmesg(8) 메뉴얼 페이지를 참조하시기 바랍니다.

다음에 출력된 `dmesg` 명령 결과는 시작시 인식된 시리얼 확장 카드를 보여줍니다:

```
May 22 14:02:10 storage3 kernel: Cyclades driver 2.3.2.5 2000/01/19 14:35:33
May 22 14:02:10 storage3 kernel: built May 8 2000 12:40:12
May 22 14:02:10 storage3 kernel: Cyclom-Y/PCI #1: 0xd0002000-0xd0005fff, IRQ9,
```

```
 4 channels starting from port 0.
```

다음에 출력된 dmesg 명령 결과 예시는 시스템 상에서 검색된 두 개의 외부 SCSI 버스와 9 개의 디스크를 보여줍니다. (슬래시로 시작된 줄은 대부분의 화면에서 한 줄로 인쇄된다는 사실에 유의해 주십시오):

```
May 22 14:02:10 storage3 kernel: scsi0 : Adaptec AHA274x/284x/294x \
   (EISA/VLB/PCI-Fast SCSI) 5.1.28/3.2.4
May 22 14:02:10 storage3 kernel:
May 22 14:02:10 storage3 kernel: scsi1 : Adaptec AHA274x/284x/294x \
      (EISA/VLB/PCI-Fast SCSI) 5.1.28/3.2.4
May 22 14:02:10 storage3 kernel:
May 22 14:02:10 storage3 kernel: scsi : 2 hosts.
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST39236LW    Rev: 0004
May 22 14:02:11 storage3 kernel: Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdb at scsi1, channel 0, id 0, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdc at scsi1, channel 0, id 1, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdd at scsi1, channel 0, id 2, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sde at scsi1, channel 0, id 3, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdf at scsi1, channel 0, id 8, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdg at scsi1, channel 0, id 9, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST31820LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdh at scsi1, channel 0, id 10, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: SEAGATE  Model: ST318203LC    Rev: 0001
May 22 14:02:11 storage3 kernel: Detected scsi disk sdi at scsi1, channel 0, id 11, lun 0
May 22 14:02:11 storage3 kernel:  Vendor: Dell   Model: 8 BAY U2W CU   Rev: 0205
May 22 14:02:11 storage3 kernel:  Type:  Processor \
           ANSI SCSI revision: 03
May 22 14:02:11 storage3 kernel: scsi1 : channel 0 target 15 lun 1 request sense \
   failed, performing reset.
May 22 14:02:11 storage3 kernel: SCSI bus is being reset for host 1 channel 0.
May 22 14:02:11 storage3 kernel: scsi : detected 9 SCSI disks total.
```

다음에 출력된 dmesg 명령 결과는 시스템 상에서 검색된 quad 이더넷 카드를 보여줍니다:

```
May 22 14:02:11 storage3 kernel: 3c59x.c:v0.99H 11/17/98 Donald Becker
May 22 14:02:11 storage3 kernel: tulip.c:v0.91g-ppc 7/16/99 becker@cesdis.gsfc.nasa.gov
May 22 14:02:11 storage3 kernel: eth0: Digital DS21140 Tulip rev 34 at 0x9800, \
   00:00:BC:11:76:93, IRQ 5.
May 22 14:02:12 storage3 kernel: eth1: Digital DS21140 Tulip rev 34 at 0x9400, \
   00:00:BC:11:76:92, IRQ 9.
May 22 14:02:12 storage3 kernel: eth2: Digital DS21140 Tulip rev 34 at 0x9000, \
   00:00:BC:11:76:91, IRQ 11.
May 22 14:02:12 storage3 kernel: eth3: Digital DS21140 Tulip rev 34 at 0x8800, \
   00:00:BC:11:76:90, IRQ 10.
```

2.3.5. 커널에서 설정된 장치 표시

시리얼과 네트워크 인터페이스와 같은 설치된 장치가 커널에서 설정되었는지 확인하시려면, 각 클러스터 시스템에서 `cat /proc/devices` 명령을 사용하시기 바랍니다. 이 명령은 또한 시스템 상에 원장치 지원이 설치되었는지 알아내는데도 사용됩니다. 예를 들면:

```
# cat /proc/devices
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 ttyS
  5 cua
  7 vcs
 10 misc
 19 ttyC
 20 cub
128 ptm
136 pts
162 raw

Block devices:
  2 fd
  3 ide0
  8 sd
 65 sd
#
```

위의 예시에서는 다음을 보여줍니다:

- 내장형 시리얼 포트 (ttyS)
- 시리얼 확장 카드 (ttyC)
- 원장치 (raw)
- SCSI 장치 (sd)

2.4. 클러스터 하드웨어 설정과 연결 단계

Red Hat Linux를 설치하신 후, 클러스터 하드웨어 구성 요소를 설정하시고 설치를 검증하여 클러스터 시스템이 연결된 모든 장치를 인식하는지 확인해 주셔야 합니다. 하드웨어를 설정하는 정확한 단계는 설정 유형에 따라 달라진다는 사실을 유의해 주십시오. 클러스터 설정에 대한 보다 많은 정보는 2.1 절을 참조하시기 바랍니다.

클러스터 하드웨어를 설정하시려면, 다음 단계를 따르십시오:

1. 클러스터 시스템을 종료하고 전원 소스에서 연결 해제하시기 바랍니다.
2. point-to-point 이더넷 연결과 시리얼 heartbeat 채널을 설정하십시오. 이 작업을 수행하는 방법에 대한 보다 많은 정보를 원하신다면, 2.4.1 절을 참조하시기 바랍니다.
3. 전원 스위치를 사용하실 때, 장치를 설정하시고 각 클러스터 시스템을 전원 스위치에 연결하십시오. 이 작업을 수행하는 방법에 대한 보다 많은 정보가 필요하시면, 2.4.2 절을 참조하시기 바랍니다.

   추가로 각 전원 스위치를 (전원 스위치를 사용하지 않는 경우 각 클러스터 시스템의 전원 코드를) 다른 UPS 시스템에 연결하시길 권장합니다. 옵션인 UPS 시스템을 사용하는 방법에 대한 정보를 원하신다면, 2.4.3 절을 참조하시기 바랍니다.

4. 제조업체의 지시 사항에 따라 공유 디스크 저장 공간을 설정하시고 외부 저장 공간으로 클러스터 시스템을 연결하시기 바랍니다. 이 작업을 수행하는 방법에 대한 보다 많은 정보를 원하신다면, 2.4.4 절을 참조하시기 바랍니다.

   추가로 중복 UPS 시스템에 저장 공간을 연결하시길 권장합니다. 옵션인 UPS 시스템을 사용하는 방법에 대한 보다 많은 정보를 원하신다면, 2.4.3 절을 참조하시기 바랍니다.

5. 하드웨어로 전원을 켜신 후 각 클러스터 시스템을 부팅하십시오. 부팅 과정에서 BIOS 유틸리티로 들어가셔서 시스템 설정을 다음과 같이 변경해 주십시오:

   - HBA에 의해 사용되는 SCSI ID 번호가 연결된 SCSI 버스에 유일한 번호인지 확인해 주십시오. 이 작업을 수행하는 방법에 대한 보다 많은 정보는 A.5 절을 참조하시기 바랍니다.
   - 저장 장치 설정에 의해 요구되는 대로 각 호스트 버스 어댑터에 내장된 종료 기능을 활성화 또는 비활성화 하시기 바랍니다. 이 작업을 수행하는 방법에 대한 보다 많은 정보는 2.4.4 절과 A.3 절을 참조하시기 바랍니다.
   - 전원이 꺼졌을 때 자동으로 부팅하도록 클러스터 시스템을 활성화 하십시오.

6. BIOS 유틸리티를 종료하시고 각 시스템을 계속해서 부팅하시기 바랍니다. 리눅스 커널이 설정되었는지 공유 디스크 모든 세트를 인식 가능한지를 확인하기 위해 시작 메시지를 살펴 보십시오. `dmesg` 명령을 사용하여 콘솔 시작 메시지를 보실 수 있습니다. 이 명령어에 대한 보다 많은 정보는 2.3.4 절을 참조하십시오.
7. `ping` 명령을 사용하여 각 네트워크 인터페이스로 패킷을 전달하여 클러스터 시스템이 각 point-to-point 이더넷 heartbeat 연결 상에서 통신할 수 있는지 확인해 보십시오.
8. 공유 디스크 저장 장치 상에 quorum 디스크 파티션을 설정하십시오. 이 작업을 수행하는 방법에 대한 보다 많은 정보는 2.4.4.3 절을 참조하시기 바랍니다.

2.4.1. Heartbeat 채널 설정하기

클러스터는 클러스터 시스템의 페일 오버 과정에서 heartbeat 채널을 입력 수단으로 사용합니다. 예를 들어 만일 클러스터 시스템이 quorum 파티션에서 시간 도장 업데이트를 멈춘다면, 다른 클러스터 시스템은 heartbeat 채널의 상태를 확인하여 페일오버를 시작하기 전에 추가 시간이 할당되어야 할 지 여부를 결정합니다.

클러스터는 최소한 한 개의 heartbeat 채널을 가지고 있어야 합니다. 클라이언트 액세스와 heartbeat 채널 모두에 이더넷 연결을 사용하는 것도 가능하비낟. 그러나 고가용성을 위해 추가 heartbeat 채널을 설정하시길 권장합니다. 한 개 이상의 시리얼 heartbeat 채널에 중복 이더넷 heartbeat 채널을 추가하여 사용하시기 바랍니다.

예를 들어 이더넷 heartbeat 채널과 시리얼 heartbeat 채널을 모두 사용하는 경우 이더넷 채널에 상요되는 케이블이 연결 해제되어도 클러스터 시스템은 시리얼 heartbeat 채널을 통하여 다른 시스템의 상태를 확인 가능합니다.

중복 이더넷 heartbeat 채널을 설정하시려면, 네트워크 크로스오버 케이블을 사용하여 한 클러스터 시스템 상의 네트워크 인터페이스를 다른 클러스터 시스템 상의 네트워크 인터페이스로 연결하시기 바랍니다.

시리얼 heartbeat 채널을 설정하시려면, 널(null) 모뎀 케이블을 사용하여 한 클러스터 시스템 상의 시리얼 포트를 다른 클러스터 상의 시리얼 포트로 연결하시면 됩니다. 클러스터 시스템에서 서로 상응하는 시리얼 포트에 연결하셔야 합니다; 원격 전원 스위치 연결에 사용되는 시리얼 포트에 연결하시면 안됩니다. 향후 두 개 이상의 클러스터 구성원에 대한 지원이 추가될 것입니다. 그렇게 되면 시리얼 기반 heartbeat 채널은 더 이상 사용되지 않을 것입니다.

2.4.2. 전원 스위치 설정하기

전원 스위치는 한 개의 클러스터 시스템이 페일 오버 과정의 일부로 자신의 서비스를 재시작하기 전에 다른 클러스터 시스템에 전원사이클할 수 있게 해줍니다. 원격적으로 시스템을 비활성화할 수 있는 기능 덕분에 시스템 고장이 발생하는 경우에도 데이터 무결성이 유지됩니다. 생산 환경에서는 클러스터 설정에 전원 스위치나 감시 타이머 (watchdog timer)를 사용하시길 바랍니다. 개발용 (테스트) 환경에서만 전원 스위치가 없는

설정 ("None" 입력)을 사용하셔야 합니다. 다양한 유형의 전원 스위치에 대한 설명을 원하신다면, 2.1.3 절을 참조하시기 바랍니다. 이 장에서는 일반 용어로서 "전원 스위치"에는 감시 타이머도 포함됩니다.

물리적 전원 스위치를 사용하는 클러스터 설정에서, 각 클러스터 시스템의 전원 케이블은 (스위치 유형에 따라서) 시리얼 연결이나 네트워크 연결을 통하여 전원 스위치에 연결됩니다. 페일 오버가 발생하는 경우, 클러스터 시스템은 이 연결을 사용하여 자신의 서비스를 재시작하기 전에 다른 클러스터 시스템에 전력을 공급할 수 있습니다.

전원 스위치는 응답이 없는 (또는 멈춘) 시스템이 서비스가 페일 오버된 후 다시 반응하며, 다른 클러스터 시스템으로 부터 I/O를 받는 디스크에 I/O를 발행할 경우 데이터가 손상되는 것을 방지합니다. 추가로 만일 quorum 데몬이 클러스터 시스템에서 실패할 경우, 그 시스템은 더 이상 quorum 파티션을 모니터할 수 없게 됩니다. 만일 전원 스위치나 감시 타이머가 클러스터에서 사용되지 않는다면, 이러한 오류 상태로 인해 서비스가 한 개 이상의 클러스터 시스템에서 실행되는 결과를 초래할 수도 있습니다. 이 경우 데이터가 손상되고 시스템이 파손될 가능성이 있습니다.

클러스터에서 전원 스위치를 사용하시길 적극 권장합니다. 그러나 위험 가능성을 인지하고 계신 관리자 분들은 전원 스위치 없이 클러스터를 설정하실 수도 있습니다.

클러스터 시스템이 교환되거나 시스템 작업 부하가 높을 경우 몇 초간 멈출 수도 있습니다. 이러한 이유로 충분한 시간을 두고 다른 시스템이 실패했는지 결정합니다. (보통 12초)

하드웨어가 실패하거나 커널 오류가 발생한 경우 클러스터 시스템이 무기한으로 "멈출" 경우가 있습니다. 이러한 경우에는, 다른 클러스터 시스템은 멈춘 시스템이 quorum 파티션에 시간 도장을 업데이트하지 않으며 heartbeat 채널을 통해 보낸 ping 신호에 응답하지 않는다는 사실을 인식하게 됩니다.

만일 클러스터 시스템이 멈춘 시스템이 다운되었다고 결정한다면, 클러스터에서 전원 스위치가 사용되어 자신의 서비스를 시작하기 이전에 멈춘 시스템에 전원사이클 합니다. 감시 타이머를 사용하도록 설정된 클러스터는 시스템이 멈춘 경우 대부분 스스로 재부팅합니다. 따라서 멈춘 시스템은 손상되지 않은 정상적인 상태에서 재부팅할 수 있으며 I/O를 발행하거나 서비스 테이터가 손상되는 것을 방지합니다.

클러스터에서 전원 스위치가 사용되지 않은 경우, 클러스터 시스템은 멈춘 시스템이 다운되었다고 판단한다면, 실패한 시스템의 상태를 quorum 파티션에서 `DOWN`으로 설정한 후 멈춘 시스템의 서비스를 재시작합니다. 만일 멈춘 시스템이 반응한다면, 시스템의 상태가 `DOWN`으로 설정된 것을 알아차리고 시스템 재부팅을 시작합니다. 이렇게 함으로서 양 클러스터 시스템이 동일한 디스크에 I/O를 발행할 수 있는 시간을 최소화 해줍니다. 그러나 전원 스위치처럼 데이터 무결성을 보호할 수 있다고 장담할 수는 없습니다. 만일 멈춘 시스템이 끝까지 반응하지 않는 경우 전원 스위치가 사용되지 않았다면 수동으로 재부팅하셔야 합니다.

전원 스위치가 사용하시려면, 판매업체의 지시 사항에 따라 전원 스위치를 설정하셔야 합니다. 그러나 클러스터에서 전원 스위치를 사용하시려면 일부 클러스터 특유의 작업이 필요합니다. 전원 스위치 (와 감시 타이머)에 대한 자세한 정보를 보시려면, A.1 절을 참조하시기 바랍니다. 경고나 특정 전원 스위치 타입의 기능 숙성을 기록해 두십시오. 클러스터 특유의 정보는 판매업체에서 제공한 정보 보다 이 문서에서 제공된 정보를 사용하시기 바랍니다.

전원 스위치를 케이블로 연결시, 각별히 주의하여 각 케이블을 적절한 접속 포트로 플러그하시기 바랍니다. 적절한 케이블 연결 여부를 확인할 소프트웨어가 존재하지 않기 때문에 이 과정을 특히 중요합니다. 케이블을 적절히 연결하지 않을 경우, 틀린 시스템이 전원사이클 되거나, 시스템이 다른 클러스터 구성원을 성공적으로 전원사이클 했다고 잘못 판단하게 되는 결과를 초래할 수 있습니다.

전원 스위치를 설정하신 후, 다음과 같은 작업을 실행하여 클러스터 시스템에 전원 스위치를 연결하십시오:

1. 각 클러스터 시스템에 사용되는 전원 케이블을 전원 스위치에 연결하십시오.
2. 각 클러스터 시스템에서 시리얼 포트를 다른 클러스터 시스템으로 전원을 제공하는 전원 스위치 상의 시리얼 포트로 연결하시기 바랍니다. 시리얼 연결에 사용된 케이블은 전원 스위치의 유형에 따라 결정됩니다. 예를 들면, RPS-10 전원 스위치는 널(null) 모뎀 케이블을 사용하는 반면에 네트워크에 연결된 전원 스위치는 네트워크 케이블을 사용합니다.
3. 각 전원 스위치에 사용되는 전원 케이블을 전원 소스로 연결하십시오. 각 전원 스위치를 다른 UPS 시스템에 연결하시길 권장합니다. 보다 많은 정보를 원하신다면, 2.4.3 절을 참조하시기 바랍니다.

클러스터 소프트웨어를 설치하신 후, 클러스터를 시작하시기 전에 각 클러스터 시스템이 다른 시스템을 전원사이클 할 수 있는지 전원 스위치를 테스트해 보시기 바랍니다. 보다 많은 정보는 3.2.2 절을 참조해 보십시오.

2.4.3. UPS 시스템 설정하기

UPS (Uninterruptible power supply) 시스템은 가용성이 높은 전원 자원을 제공합니다. 이상적으로 다중 UPS를 (서버 당 한개씩) 통합하기 위해서 중복 솔루션을 사용하시는 것이 좋습니다. 장애 안전 기능을 최대화하기 위해서는, APC의 자동 전송 스위치를 사용하여 서버의 전원과 종료 작업을 관리할 뿐만 아니라 서버 당 두 개의 UPS를 통합하는 것이 가능합니다. 두 가지 솔루션은 원하시는 가용성의 수준에 따라 달라집니다.

대형 UPS 구조를 클러스터에 사용되는 유일한 전원 자원으로 사용하시는 것을 권장하지 않습니다. 클러스터 전용 UPS 솔루션은 관리와 가용성 면에서 보다 많은 융통성을 허용합니다.

완전한 UPS 시스템은 오랜 기간 동안 충분한 전압과 전류를 제공할 수 있어야 합니다. 비록 모든 전원 요건을 충족시킬 수 있는 단독 UPS는 존재하지 않지만, 특정 설정에 맞는 솔루션을 만들어낼 수는 있습니다. 여러분의 서버에 맞는 UPS 설정을 알아 보시려면, APC의 UPS 설정 프로그램인 http://www.apcc.com/template/size/apc 사이트를 방문해 보시기 바랍니다. APC Smart-UPS 제품은 Red Hat Linux 용 소프트웨어 관리 도구를 함께 배포합니다. **RPM** 패키지 이름은 `pbeagent` 입니다.

만일 클러스터 디스크 저장 하부 시스템이 별개의 전원 코드를 통해 두 곳에서 전원사이클 받는다면, 두 개의 UPS 시스템을 설정하시고 한 개의 전원 스위치를 연결하시며 (또는 전원 스위치를 사용하지 않는 경우 한 개의 클러스터 시스템의 전원 코드를 연결하시며) 각 UPS 시스템에 저장 하부 시스템의 전원 코드 중 하나를 연결하시기 바랍니다. 중복 UPS 시스템 설정은 그림 2-3에서 보여진 것과 같습니다.

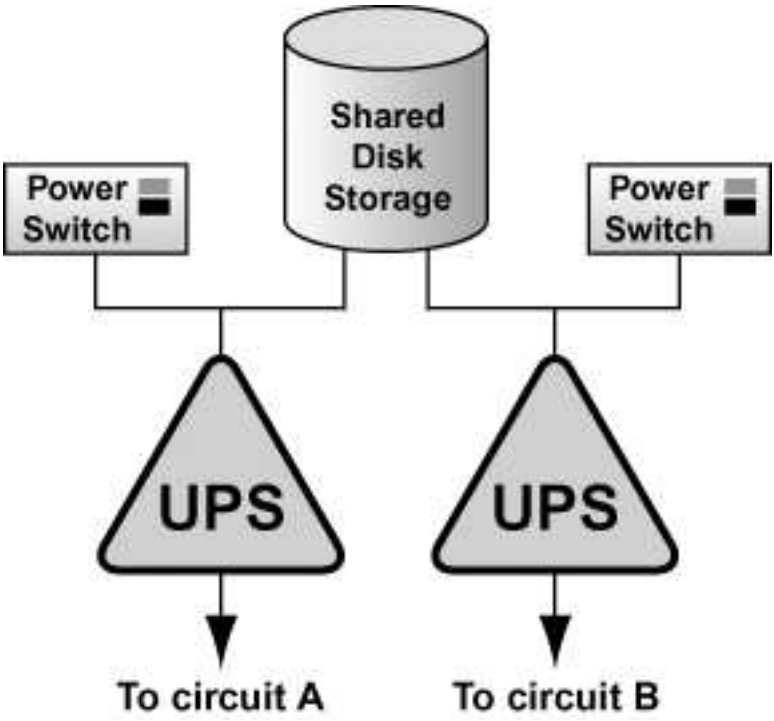

그림 2-3. 중복 UPS 시스템 설정

대안적인 중복 전원 설정은 양 전원 스위치 (또는 양 클러스터 시스템의 전원 코드)와 디스크 저장 하부 시스템을 동일한 UPS 시스템에 연결하는 것입니다. 이것은 가장 비용이 절감되는 설정으로서 정전이 발생할 경우 보호 기능을 제공합니다. 그러나 만일 정전이 발생한다면, 단독 UPS 시스템이 실패할 경우 모든 시스템에 영향을 미치게 됩니다. 또한 한 개의 UPS 시스템으로는 연결된 모든 장치에 필요한 시간 동안 충분한 전원을 공급하지 못할 수도 있습니다. 단독 UPS 시스템 설정은 그림 2-4에서 보여진 것과 같습니다.

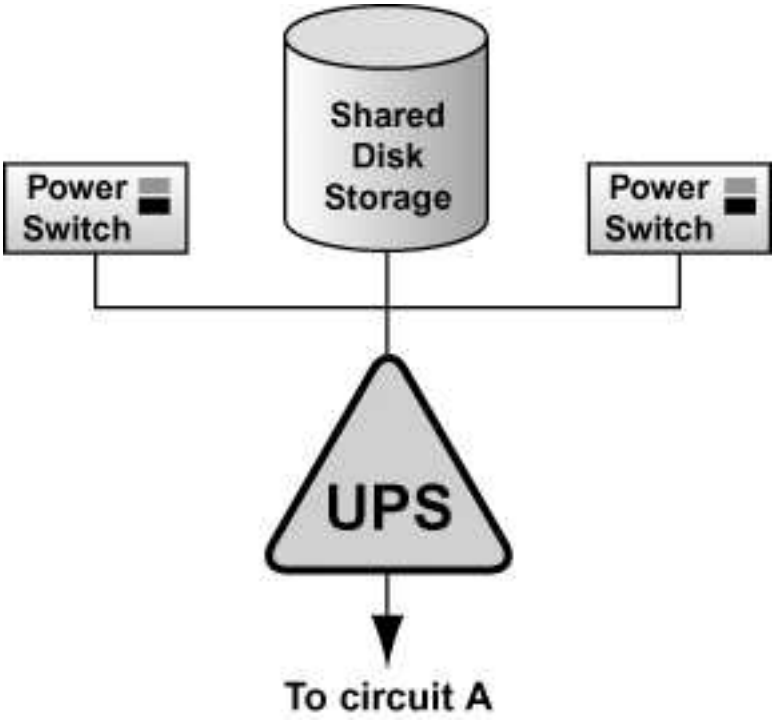

그림 2-4. 단독 UPS 시스템 설정

다수의 상용 UPS 시스템에는 시리얼 포트 연결을 통해 UPS 시스템의 작업 상태를 모니터하는 리눅스 응용 프로그램이 포함되어 있습니다. 만일 배터리 전원이 낮은 경우, 감시 소프트웨어는 안전하게 시스템을 종료합니다. 이러한 상황이 발생하는 경우, 클러스터 소프트웨어는 System V 런레벨 스크립트 (예, `/etc/rc.d/init.d/cluster`)에 의해 조정되기 때문에 적절히 종료됩니다.

보다 자세한 설치 정보를 원하신다면 판매업체에서 제공하는 UPS 문서자료를 참조하시기 바랍니다.

2.4.4. 공유 디스크 저장 공간 설정하기

클러스터에서 공유 디스크 저장 공간은 서비스 데이터와 두 개의 quorum 파티션을 저장하는데 사용됩니다. 이 저장 공간은 양 클러스터 시스템에서 사용 가능해야 하기 때문에, 한 개의 시스템의 가용성에 의존하는 디

스크 상에 위치해서는 안됩니다. 보다 자세한 제품 정보와 설치 정보는 판매업체의 문서 자료를 참조하시기 바랍니다.

클러스터에서 공유 디스크 저장을 설정하실 때 고려해야할 몇 가지 요소는 다음과 같습니다:

- 외부 RAID

  서비스 데이터와 quorum 파티션의 가용성을 높이기 위해 RAID 1 (미러링)을 사용하시길 적극 권장합니다. 고 가용성을 위해 패리티(parity) RAID를 사용하실 수도 있습니다. quorum 파티션에 RAID 0 (스트라이핑)만 사용하지는 마십시오.

- 다중 시작 프로그램 SCSI 설정

  적절한 버스 종료가 힘들기 때문에 다중 시작 프로그램 SCSI 설정은 지원되지 않습니다.

- 각 공유 저장 장치에 대한 리눅스 장치명은 각 클러스터 시스템에서 동일해야 합니다. 예를 들어, 한 클러스터 시스템에서 `/dev/sdc` 장치라고 불리운다면 다른 클러스터 시스템 상에서도 `/dev/sdc`라고 불리워야 합니다. 양 클러스터 시스템에서 동일한 하드웨어를 사용함으로서 이러한 장치명도 동일하게 이름붙도록 하실 수 있습니다.

- 디스크 파티션은 오직 한 개의 클러스터 서비스에 의해 사용됩니다.

- 클러스터 시스템의 로컬 `/etc/fstab` 파일에 클러스터 서비스에서 사용된 어떠한 파일 시스템도 포함시키지 마십시오. 왜냐하면 클러스터 소프트웨어가 서비스 파일 시스템의 마운트 작업과 마운트 해제 작업을 조정해야 하기 때문입니다.

- 최적의 성능을 위해서는, 공유 파일 시스템을 생성하실 때 4 KB 블록 크기를 사용하시기 바랍니다. 파일 시스템을 생성시 많은 **mkfs** 파일 시스템 개발 유틸리티가 1 KB 블록 크기로 디폴트 지정되어 있기 때문에 `fsck` 시간이 길어 집니다. 따라서 대부분의 경우 블록 크기를 4 KB로 지정해 주셔야 합니다.

다음 목록은 병렬 *SCSI* 요건을 보여주고 있습니다. 클러스터 환경에서 병렬 SCSI를 사용한다면 다음과 같은 요건이 반드시 지켜져야 합니다:

- SCSI 버스는 각 마지막 지점에서 종료되어야 하며, 길이 제한과 핫 플러깅 제한을 지켜야 합니다.
- SCSI 버스 상 장치들 (디스크들, 핫 버스 어댑터, RAID 제어기들)은 유일한 SCSI 식별 번호가 있어야 합니다.

보다 많은 정보를 원하신다면, A.2 절을 참조하시기 바랍니다.

추가로 전력의 고가용성 소스로서 공유 저장 공간을 중복 UPS 시스템으로 연결하시길 적극 권장합니다. 보다 많은 정보는 2.4.3 절을 참조하시기 바랍니다.

공유 저장 공간을 설정하는 방법에 보다 많은 정보를 원하신다면, 2.4.4.1 절과 2.4.4.2 절을 참조하시기 바랍니다.

공유 디스크 저장 하드웨어를 설정하신 후, 디스크를 파티션하시고 그 파티션에 파일 시스템을 생성하시거나 원장치를 생성하시기 바랍니다. 일차 quorum 파티션과 백업 quorum 파티션에 대한 두 개의 원장치를 생성하셔야 합니다. 보다 많은 정보를 원하신다면, 2.4.4.3 절, 2.4.4.4 절, 2.4.4.5 절, 그리고 2.4.4.6 절을 참조하시기 바랍니다.

2.4.4.1. 단독 시작 프로그램 SCSI 버스 설정하기

단독 시작 프로그램 SCSI 버스에는 오직 한 클러스터 시스템만이 연결됩니다. 단독 시작 프로그램은 다중 시작 프로그램 버스 보다 나은 호스트 분리와 향상된 성능을 제공할 수 있습니다. 단독 시작 프로그램 상호 연결은 각 클러스터 시스템이 작업 부하, 초기화 또는 다른 클러스터 시스템을 복구하면서 시스템이 손상되는 것을 막아줍니다.

만일 다중 호스트 포트를 가진 RAID 어레이를 사용하시는 경우 RAID 어레이가 저장 장치 상의 호스트 포트에서 모든 공유 논리 장치로 동시에 접근하게 해준다면, 각 클러스터 시스템을 RAID 어레이로 연결하도록 두 개의 단독 시작 프로그램 SCSI 버스를 설정하시기 바랍니다. 만일 논리 장치가 한 제어기에서 다른 제어기로 페일 오버된다면, 페일 오버되는 과정은 운영 체제에게 투명해야 합니다. 일부 RAID 제어기는 특정 제어기 또는 포트에 디스크 세트를 제한합니다. 이러한 경우, 단독 시작 프로그램 버스 설정은 불가능합니다.

단독 시작 프로그램 버스는 A.2 절에서 설명된 요건을 충족 시켜야 합니다. 추가로 호스트 버스 어댑터를 종료하고 단독 시작 프로그램 버스를 설정하는 방법에 대한 보다 자세한 정보는 A.6 절을 참조하시기 바랍니다.

단독 시작 프로그램 SCSI 버스를 설정하시려면, 다음과 같은 요건이 충족되어야 합니다:

- 각 호스트 버스 어댑터에 대한 보드 장착된 멈춤 기능 활성화.
- 각 RAID 제어기에 해단 멈춤 기능 활성화.
- 적절한 SCSI 케이블을 사용하여 각 호스트 버스 어댑터를 저장 장치로 연결.

호스트 버스 어댑터 멈춤 기능을 설정하는 것은 보통 시스템 부팅시 어댑터 BIOS 유틸리티에서 이루어 집니다. RAID 제어기 멈춤을 설정하시려면, 판매업체에서 나온 문서 자료를 참조해 보십시오. 그림 2-5에서는 두 개의 단독 시작 프로그램 SCSI 버스를 사용하는 설정을 보여줍니다.

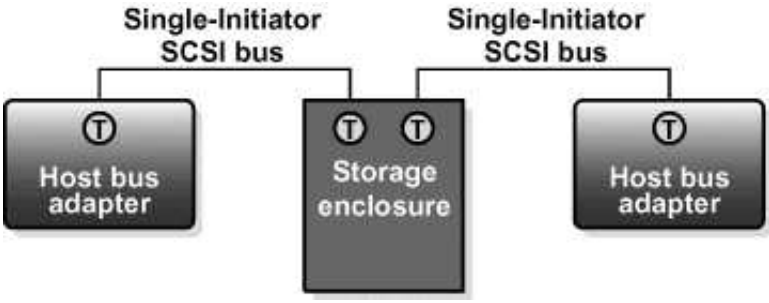

그림 2-5. 단독 시작 프로그램 SCSI 버스 설정

그림 2-6에서는 두 개의 단독 시작 프로그램 SCSI 버스에 연결된 단독 제어기 RAID 어레이에서 멈춤을 보여줍니다.

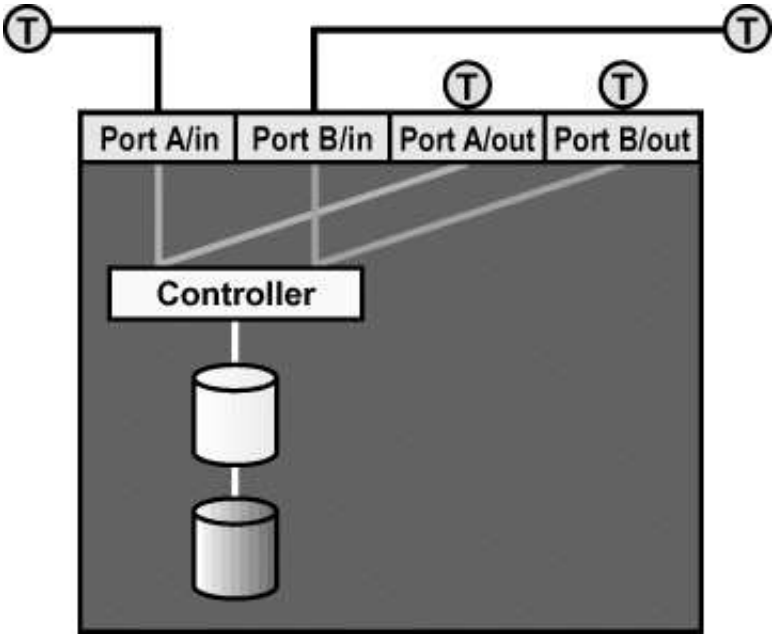

그림 2-6. 단독 시작 프로그램 SCSI 버스에 연결된 단독 제어기 RAID 어레이

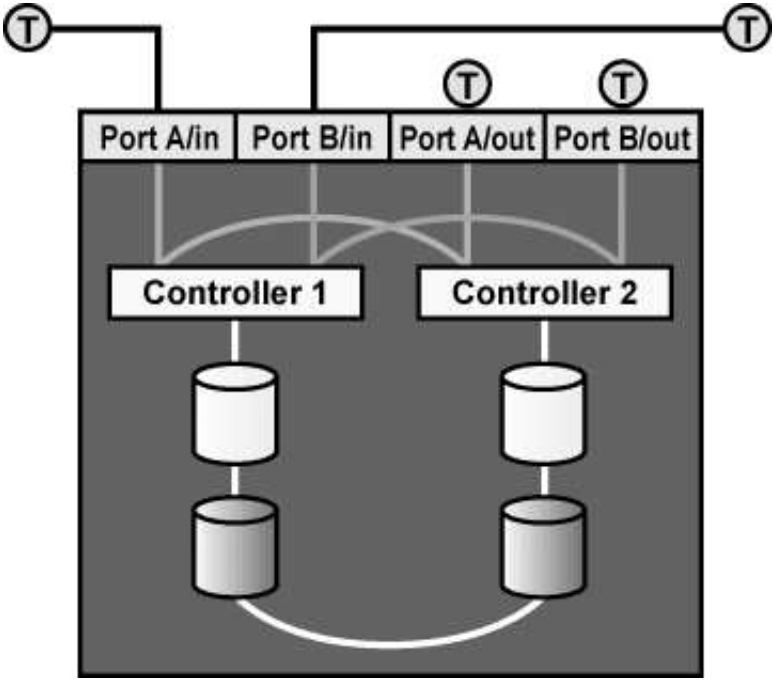

그림 2-7. 단독 시작 프로그램 SCSI 버스에 연결된 이중 제어기 RAID 어레이

2.4.4.2. 광 채널 상호 연결 설정하기

광 채널은 단독 시작 또는 다중 시작 프로그램 설정에 사용될 수 있습니다.

단독 시작 프로그램 광 채널 상호 연결에는 오직 한 클러스터 시스템만이 연결됩니다. 단독 시작 프로그램은 다중 시작 프로그램 버스 보다 나은 호스트 분리와 향상된 성능을 제공할 수 있습니다. 단독 시작 프로그램 상호 연결은 각 클러스터 시스템이 작업 부하, 초기화 또는 다른 클러스터 시스템을 복구하면서 시스템이 손상되는 것을 막아줍니다.

만일 다중 호스트 포트를 가진 RAID 어레이를 사용하시는 경우 RAID 어레이가 저장 장치 상의 호스트 포트에서 모든 공유 논리 장치로 동시에 접근하게 해준다면, 각 클러스터 시스템을 RAID 어레이로 연결하도록 두 개의 단독 시작 프로그램 광 채널 상호 연결을 설정하시기 바랍니다. 만일 논리 장치가 한 제어기에서 다른 제어기로 페일 오버된다면, 페일 오버되는 과정은 운영 체제에게 투명해야 합니다.

그림 2-8에서는 두 갱의 호스트 포트를 가진 단독 제어기 RAID 어레이와 광 채널 헙(hub)이나 스위치를 사용하지 않고 RAID 제어기에 직접 연결된 호스트 버스 어댑터를 보여줍니다.

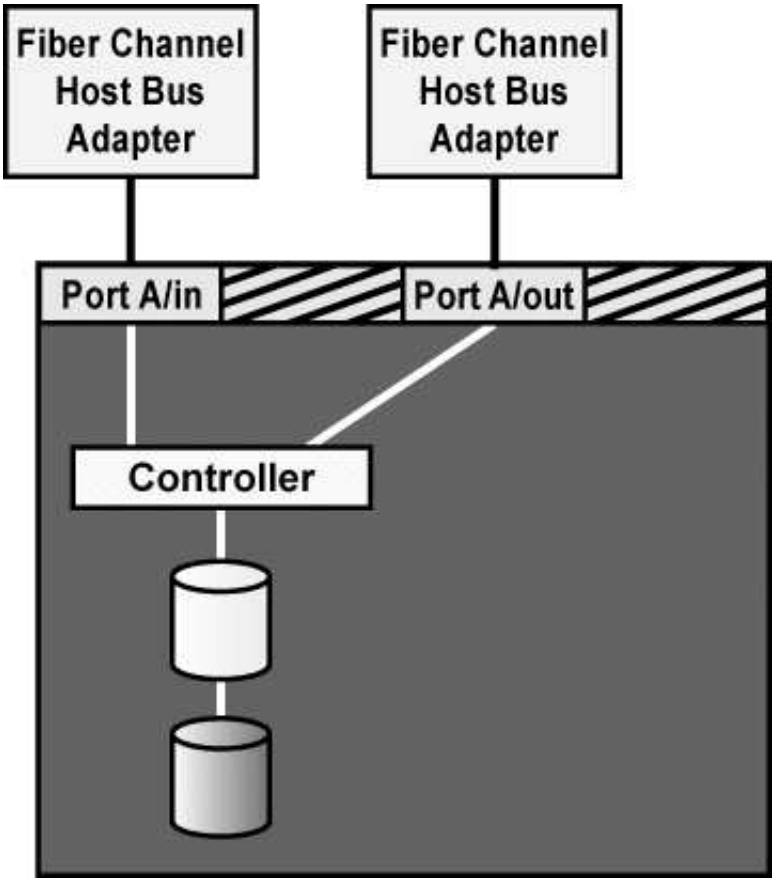

그림 2-8. 단독 시작 프로그램 광 채널 상호 연결에 연결된 단독 제어기

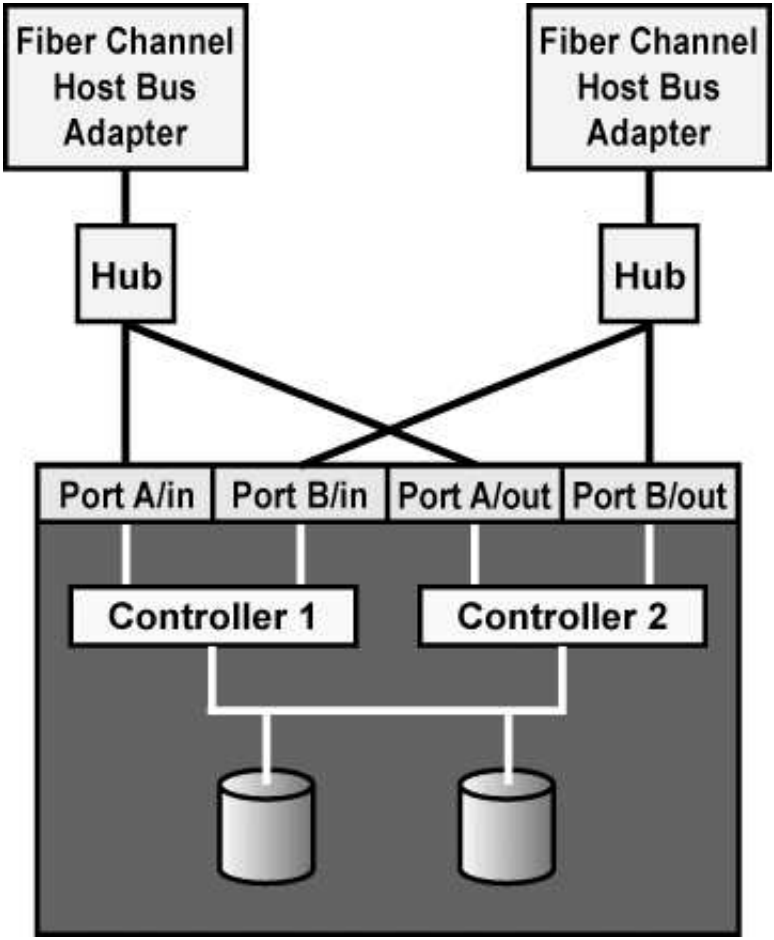

그림 2-9. 단독 시작 프로그램 광 채널 상호 연결에 연결된 이중 제어기 RAID 어레이

만일 각 제어기에서 두 개의 호스트 포트를 가진 이중 제어기 RAID 어레이가 사용된다면, 그림 2-9에서 보여진 것처럼 양 제어기에 위치한 한 개의 포트에 각 호스트 버스 어댑터를 연결하기 위해 광 채널 헙(hub)또는 스위치가 필요합니다.

다중 시작 프로그램 광 채널이 사용된 경우, 광 채널 헙(hub)이나 스위치가 필요합니다. 이러한 경우 각 HBA는 헙이나 스위치에 연결되며, 헙이나 스위치는 각 RAID 제어기 상의 호스트 포트로 연결됩니다.

2.4.4.3. Quorum 파티션 설정하기

공유 디스크 저장 장치 상에 일차 quorum 파티션과 백업 quorum 파티션를 위한 두 개의 원 장치를 생성하셔야 합니다. 각 quorum 파티션은 최소한 10 MB이어야 합니다. quorum 파티션에 저장된 데이터 용량은

일정합니다; 즉, 시간이 지나도 용량이 늘거나 줄거나 하지 않습니다.

quorum 파티션은 클러스터 상태 정보를 저장하는데 사용됩니다. 각 클러스터 시스템은 주기적으로 시스템 상태 정보 (*UP* 또는 *DOWN*), 시간 도장과 서비스의 상태를 기록합니다. 또한 quorum 파티션에는 클러스터 데이터베이스의 버전이 포함됩니다. 따라서 각 클러스터 시스템은 일관된 클러스터 설정을 갖추게 됩니다.

클러스터가 제대로 실행되고 있는지를 모니터하기 위해서, 크러스터 시스템은 주기적으로 일차 quorum 파티션으로부터 상태 정보를 읽어온 후 그 파티션이 업데이트되었는지 확인합니다. 만일 일차 파티션이 손상되었다면, 클러스터 시스템은 백업 quorum 파티션에서 정보를 가져와서 동시에 일차 파티션에 기록하여 복구합니다. 체크섬을 통하여 데이터의 일관성이 지켜지며, 파티션 사이에 데이터가 일관적이지 않다면 자동으로 고쳐집니다.

만일 시스템이 시작시 양 quorum 파티션에 기록하지 못한다면, 클러스터에 들어갈 수 없게 됩니다. 또한 활성 클러스터 시스템이 더 이상 양 quorum 파티션에 기록할 수 없다면, 그 시스템은 재부팅하여 클러스터에서 스스로 빠져나갑니다 (그리고 활동 중인 클러스터 구성원에 의해 원격적으로 전원이 다시 넣어질 수 있습니다).

다음은 *quorum* 파티션 요건 입니다:

- 양 quorum 파티션은 최소한 10 MB의 용량을 가지고 있어야 합니다.
- Quorum 파티션은 원장치이어야 합니다. 이 파티션은 파일 시스템을 포함할 수 없습니다.
- Quorum 파티션은 오직 클러스터 상태와 설정 정보에만 사용됩니다.

다음은 quorum 파티션을 설정하는데 권장되는 지침 사항 입니다:

- 공유 저장 장치에 대한 RAID 하부 시스템을 설정하시고, quorum 파티션을 포함한 논리 장치의 가용성을 높이기 위해 RAID 1 (미러링)을 사용하시길 적극 권장합니다. 고 가용성을 위해 패리티(parity) RAID를 사용하실 수도 있습니다. quorum 파티션에 RAID 0 (스트라이핑)만 사용하지는 마십시오.
- 양 quorum 파티션을 동일한 RAID 세트에 위치시키거나 만일 RAID가 사용되지 않는다면 동일한 디스크 상에 위치시키셔야 합니다. 그 이유는 양 quorum 파티션이 사용 가능해야 클러스터가 실행되기 때문입니다.
- 자주 사용되는 서비스 데이터가 포함된 디스크 상에 quorum 파티션을 놓지 마십시오. 가능하다면 서비스 데이터가 거의 사용되지 않는 디스크에 quorum 파티션을 위치하시기 바랍니다.

quorum 파티션을 설정하는 방법에 대한 보다 많은 정보를 원하신다면, 2.4.4.4 절과 2.4.4.5 절을 참조하시기 바랍니다.

클러스터 시스템이 부팅할 때마다 원 문자 장치를 블록 장치에 바인드하도록 `rawdevices` 파일을 편집하는 방법에 대한 정보는 3.1.1 절을 참조해 보십시오.

2.4.4.4. 디스크 파티션하기

공유 디스크 저장 하드웨어를 설정하신 후, 클러스터에서 사용할 수 있도록 디스크를 파티션 분할하시기 바랍니다. 그 후 그 파티션에 파일 시스템이나 원장치를 생성하십시오. 예를 들어, 2.4.4.3 절에서 설명된 지시 사항을 따라서 quorum 파티션에 사용될 두 개의 원 장치를 생성하셔야 합니다.

디스크 파티션 테이블을 수정하고 디스크를 파티션 분할하기 위해 상호 대화식 `fdisk` 명령을 입력하십시오. `fdisk`가 시작되면, `p` 명령을 사용하여 현재 파티션 테이블을 보신 후 새로운 파티션을 생성하기 위해 `n` 명령을 입력하시기 바랍니다.

다음 예시는 fdisk 명령을 사용하여 디스크를 파티션하는 방법을 보여줍니다:

1. 상호 대화식 `fdisk` 명령과 함께 사용 가능한 공유 디스크 장치를 지정하여 입력해 주십시오. 프롬프트에서, `p` 명령을 지정하시면 현재 파티션 테이블을 볼 수 있습니다.
```
# fdisk /dev/sde
Command (m for help): p

Disk /dev/sde: 255 heads, 63 sectors, 2213 cylinders
Units = cylinders of 16065 * 512 bytes
```

```
Device  Boot  Start   End  Blocks  Id System
/dev/sde1      1   262 2104483+ 83 Linux
/dev/sde2     263   288  208845  83 Linux
```

2. 다음으로 사용 가능한 파티션의 번호를 알아낸 후 파티션을 추가하기 위해 n 명령을 지정하시기 바랍니다. 만일 디스크 상에 이미 3개의 파티션이 존재한다면, e 명령을 입력하여 확장 파티션을 생성하시거나 p 명령을 입력하여 일차 파티션을 생성하실 수 있습니다.
   ```
   Command (m for help): n
   Command action
    e extended
    p primary partition (1-4)
   ```
3. 필요한 파티션 번호를 지정해 주십시오:
   ```
   Partition number (1-4): 3
   ```
4. [Enter] 키를 누르시거나 다음으로 사용 가능한 실린더를 지정하시기 바랍니다:
   ```
   First cylinder (289-2213, default 289): 289
   ```
5. 필요한 파티션 용량을 지정해 주십시오:
   ```
   Last cylinder or +size or +sizeM or +sizeK (289-2213, default 2213): +2000M
   ```
   fsck을 사용하여 대용량 파티션 상의 파일 시스템을 검사한다면, 클러스터 서비스 페일오버 시간이 증가된다는 사실을 염두해 주십시오. Quorum 파티션은 최소한 10 MB가 되어야 합니다.
6. w 명령을 지정하여 디스크에 새로운 파티션 테이블을 작성할 수 있습니다:
   ```
   Command (m for help): w
   The partition table has been altered!

   Calling ioctl() to re-read partition table.

   WARNING: If you have created or modified any DOS 6.x
   partitions, please see the fdisk manual page for additional
   information.

   Syncing disks.
   ```
7. 만일 양 클러스터 시스템이 켜져 있고 공유 저장 장치에 연결된 경우 파티션이 추가되었다면, 다른 클러스터 시스템이 새로운 파티션을 인식할 수 있도록 그 클러스터 시스템을 재부팅하시기 바랍니다.

디스크를 파티션하신 후, 클러스터에서 사용할 수 있도록 그 파티션을 포맷하시기 바랍니다. 예를 들어 quorum 파티션에 사용될 파일 시스템이나 원장치를 생성하십시오.

보다 많은 정보를 원하신다면, 2.4.4.5 절 와 2.4.4.6 절을 참조하시기 바랍니다.

설치 시 하드 디스크를 파티션 분할하는 방법에 대한 기본적인 정보를 원하시면, 공식 *Red Hat Linux x86* 설치 가이드를 참조하시기 바랍니다. 공식 *Red Hat Linux x86* 설치 가이드의 부록 *E.* 디스크 파티션 소개 장에서도 파티션 분할에 대한 기본 개념을 설명하고 있습니다.

fdisk를 사용하여 디스크를 파티션하는 방법에 대한 기본 정보를 원하시면, 다음 URL http://kb.redhat.com/view.php?eid=175을 참조하시기 바랍니다.

2.4.4.5. 원장치 (Raw device) 생성하기

공유 저장 디스크를 파티션 분할 후, 그 파티션에 원장치를 생성하십시오. 파일 시스템은 블록 장치 (예, /dev/sda1)로서 성능을 향상시키기 위해 메모리에서 최근에 사용된 데이터를 캐시 저장합니다. 원장치는 캐싱을 위해 시스템 메모리를 사용하지 않습니다. 보다 많은 정보는 2.4.4.6 절을 참조하시기 바랍니다.

리눅스는 특정 블록 장치에 대해 불변 코드 (hard-coded)되지 않은 원 문자 장치를 지원합니다. 대신 리눅스는 문자 major 번호 (현재 162)를 사용하여 /dev/raw 디렉토리에서 바운드되지 않은 여러 개의 원 장치를 구현합니다. 모든 블록은 문자 원 장치 프론트 엔드를 가질 수 있으며, 런타임 이후에 읽어들인 블록 장치도 마찬가지 입니다.

원장치를 생성하시려면, 원 문자 장치를 적절한 블록 장치에 바인드하도록 `/etc/sysconfig/rawdevices` 파일을 편집하시기 바랍니다. 원장치가 블록 장치에 바운드되었다면, 그 원장치를 열고, 읽고 쓰기할 수 있습니다.

Quorum 파티션과 일부 데이터베이스 응용 프로그램은 성능 개선을 목적으로 독자적인 버퍼 캐싱을 수행하기 때문에 원장치를 필요로 합니다. Quorum 파티션은 파일 시스템을 포함할 수 없습니다. 그 이유는 만일 상태 데이터가 시스템 메모리에 캐시된다면, 클러스터 시스템의 상태 데이터가 지속적이지 않기 때문입니다.

원장치는 시스템이 부팅할 때마다 블록 장치에 바인드되어야 합니다. 이러한 작업을 확실히 하시려면, `/etc/sysconfig/rawdevices` 파일을 편집하여 quorum 파티션 바인딩을 지정하시기 바랍니다. 만일 클러스터 서비스에서 원장치를 사용하신다면, 이 파일을 사용하여 부팅시 장치를 바인드합니다. 보다 많은 정보는 3.1.1 절을 참조하시기 바랍니다.

`/etc/sysconfig/rawdevices` 파일을 편집하신 후, 시스템을 재부팅하시거나 다음 명령을 실행하시면 변경 사항이 효력을 발생할 것입니다:

```
# service rawdevices restart
```

`raw -aq` 명령을 사용하여 모든 원장치를 질의하시기 바랍니다:

```
# raw -aq
/dev/raw/raw1  bound to major 8, minor 17
/dev/raw/raw2  bound to major 8, minor 18
```

원 장치의 경우, 원 장치와 블록 장치 간에는 캐시가 일관성을 지니고 있지 않다는 점을 기억해 주십시오. 또한 서비스 요청은 메모리와 디스크 상에서 모두 512 바이트로 정렬되어야 합니다. 예를 들어, 표준 `dd` 명령은 원 장치와 함께 사용될 수 없습니다. 그 이유는 이 명령이 시스템 호출을 작성하기 위해 전달하는 메모리 버퍼가 512 바이트 경계에 정렬되지 않기 때문입니다.

`raw` 명령을 사용하는 방법에 대한 보다 많은 정보는 raw(8) 메뉴얼 페이지를 참조하시기 바랍니다.

2.4.4.6. 파일 시스템 생성하기

파티션 상에 ext2 파일 시스템을 생성하시려면 `mkfs` 명령을 사용하십시오. 다음과 같이 드라이브 문자와 파티션 번호를 지정하시기 바랍니다:

```
# mkfs -t ext2 -b 4096 /dev/sde3
```

공유 파일 시스템이 최적의 성능을 발휘하기 위해서, 위의 예시에는 블록 크기를 4 KB로 지정하였습니다. 파일 시스템을 생성시 많은 **mkfs** 파일 시스템 개발 유틸리티가 1 KB 블록 크기로 디폴트 지정되어 있기 때문에 `fsck` 시간이 길어 집니다. 따라서 대부분의 경우 블록 크기를 4 KB로 지정해 주셔야 합니다.

마찬가지로 ext3 파일 시스템을 생성하실 때 다음 명령을 사용하시면 됩니다:

```
# mkfs -t ext2 -j -b 4096 /dev/sde3
```

파일 시스템을 생성하는 방법에 대한 보다 많은 정보를 원하신다면, mkfs(8) 메뉴얼 페이지를 참조하시기 바랍니다.

3장 .

클러스터 소프트웨어 설치와 설정

클러스터 하드웨어를 설치하고 설정한 후, 클러스터 시스템 소프트웨어가 설치될 수 있습니다. 다음에서는 클러스터 소프트웨어를 설치, 초기화, 설정 검사, `syslog` 이벤트로그 설정, 그리고 `cluadmin` 사용하는 방법에 대해 설명하고 있습니다.

3.1. 클러스터 소프트웨어를 설치, 초기화하는 단계

Red Hat 클러스터 관리자을 설치하기 전에, 2.3.1 절에 나온 것처럼 필요한 소프트웨어를 모두 설치하시기 바랍니다.

클러스터 소프트웨어를 업데이트하고, 클러스터 데이터베이스의 설정을 유지하려면, 클러스터 데이터베이스를 백업하시고 클러스터 소프트웨어를 정지한 후 다시 설치하시기 바랍니다. 8.7 절을 참조해 보십시오.

Red Hat 클러스터 관리자를 설치 하시려면, `rpm --install clumanager-`*`x`*`.rpm`를 실행하십시요. 여기서 *x*는 **Red Hat 클러스터 관리자** 현재의 버전 번호를 나타냅니다. 이 패키지는 Red Hat Linux 어드밴스 서버에 기본적으로 설치되어 있습니다. 그렇기에 수동으로 이 패키지를 설치하실 필요는 없습니다.

클러스터 소프트웨어를 초기화시키고 시작하려면, 다음 단계를 따르시기 바랍니다:

1. 모든 클러스터 시스템의 `/etc/sysconfig/rawdevices`를 수정한 후, 원 장치 특별 파일, 기본 quorum 파티션과 백업 파티션을 위한 문자 장치를 지정하십시요. 더 자세한 내용은 2.4.4.3 절과 3.1.1 절을 참조하십시오.
2. 한 클러스터 시스템에 **/sbin/cluconfig**를 실행하십시오. 만일 클러스터 소프트웨어를 업데이트하신다면, 현존하는 클러스터 데이터베이스를 사용하기 전에 프로그램에서 질문을 할 것입니다. 만일 사용하지 않는다면, 프로그램이 현존하는 클러스터 데이터베이스를 삭제할 것입니다.

   프로그램이 다음과 같이 클러스터에 관련된 정보를 물을 것이며, 이것은 클러스터 데이터베이스 중에 **member** 란에 입력이 될것입니다. 이것의 복사본은 `/etc/cluster.conf` 파일 안에 있습니다:

   - `/etc/sysconfig/rawdevices`에 명시된 바와 같은 기본과 백업 quorum 파티션을 위한 원 장치 특별 파일 (예, `/dev/raw/raw1` 과 `/dev/raw/raw2`)
   - `hostname` 명령을 통해 얻어진 클러스터 시스템 호스트명
   - Heartbeat 연결 (채널) 수, 이더넷과 벙럴 모두 포함
   - 각 heartbeat 벙럴 연결을 위한 장치 특별 파일 (예를 들어, `/dev/ttyS1`)
   - 각 heartbeat 이더넷 인터페이스와 연관된 IP 호스트명
   - 원격 클러스터 감시를 위한 IP 주소, 또는, "클러스터 별칭"이라고 불림. 자세한 것을 위해서는 3.1.2 절을 참조하시기 바랍니다.
   - 만일 존재한다면, 전원 스위치가 연결된 벙럴 포트를 위한 장치 특별 파일 (예를 들어, /dev/ttyS0), 혹은 네트워크 전원 스위치가 연결되어 있는 IP 주소.
   - 전원 스위치 유형 (예를 들어, **RPS10** 혹은 **None**, 만일 전원 스위치를 사용하지 않는다면)
   - 시스템이 원격 감시를 할 것인지 아닌지를 물을 것입니다. 자세한 것은 3.1.2 절를 참조하십시요.

   프로그램을 실행하는 방법은 3.1.4 절를 참조하십시요.
3. 한 클러스터 시스템에 클러스터 초기화를 마친 후, 다음을 다른 클러스터 시스템에서 실행하시기 바랍니다:
   - `/sbin/cluconfig --init=`*`raw_file`*을 실행하십시오. 여기서 *`raw_file`*은 기본 quorum 파티션을 지정합니다. 이 스크립트는 첫 클러스터 시스템에서 지정된 내용을 기본으로 사용할 것입니다. 예를 들어:

```
cluconfig --init=/dev/raw/raw1
```

4. 클러스터 설정 검사:
   - `cludiskutil`를 `-t` 옵션과 함께 모든 클러스터 시스템에 사용해 quorum 파티션이 같은 장치를 지정하도록 합니다. 자세한 것은 3.2.1 절를 참조하십시요.
   - 만일 전원 스위치를 사용하신다면, `clustonith`를 모든 클러스터 시스템에 실행한 후, 전원 스위치로의 원격 연결을 테스트합니다. 3.2.2 절을 참조하십시요.
5. 옵션으로, 이벤트 로깅을 설정하여 클러스터 메세지가 다른 파일에 저장 되도록 합니다. 3.3 절을 참조하십시요.
6. System V `init` 디렉토리에 있는 `cluster start`를 실행하여, 클러스터를 시작합니다. 예를 들어: `service cluster start`

클러스터를 초기화시킨 후, 클러스터 서비스들을 더합니다. 3.4 절와 4.1 절 를 참조하십시요.

3.1.1. `rawdevices` 파일을 수정하기

`/etc/sysconfig/rawdevices`는 클러스터 시스템이 시작될 때, quorum 파티션이 있는 원장치를 지시하도록 되어 있습니다. 클러스터 소프트웨어 설치 중 한 단계로, `rawdevices`를 수정하고, 본 quorum 파티션과 백업 quorum 파티션이 있는 원 문자 장치와 블럭 장치를 지정하십시요. 이것은 **cluconfig** 프로그램이 실행되기 전에 실행되어야 합니다.

만일 원 장치가 클러스터 서비스에 사용된다면, `rawdevices` 파일은 시스템 시작에 장치를 연결하는 데 사용될 것입니다. 파일을 수정하고, 시스템이 시작될 때 연결되어야 할 원 문자 장치와 블럭 장치를 지정합니다. 원장치 파일을 시스템 재시작 없이 수정하려면 다음을 따라합니다:

```
service rawdevices restart
```

다음은 예제 원장치 파일로, 두개의 quorum 파티션이 지정되어 있습니다:

```
# raw device bindings
# format: <rawdev> <major> <minor>
#     <rawdev> <blockdev>
# example: /dev/raw/raw1 /dev/sda1
#     /dev/raw/raw2 8 5
/dev/raw/raw1 /dev/sdb1
/dev/raw/raw2 /dev/sdb2
```

Quorum 파티션 설정에 대해 더 자세한 내용은 2.4.4.3 절를 참조 하십시요. 또한 원 문자 장치를 블럭 장치에 연결하는 원 명령어에 대해서는 2.4.4.5 절를 참조하십시요.

주의

원장치 설정은 모든 클러스터 구성원에 되어야 합니다.

3.1.2. 클러스터 별칭 설정하기

클러스터 별칭은 한 IP 주소를 사용중인 클러스터 구성원에 연결하는 것을 나타냅니다. 어떤 시간에든 이 IP 주소는 한 클러스터 구성원에 연결될 것입니다. 이 IP 주소는 시스템 관리와 감시 목적으로 사용됩니다. 예를 들어, 서버 관리자가 운영 중인 클러스터 구성원에 `telnet`으로 접속하려는 경우, 어떤 클러스터 구성원인지는 중요하지 않습니다. 이 경우에는, 간단하게 클러스터 별칭 IP 주소 (혹은 관련된 호스트명)에 `telnet`으로

연결합니다. 가장 근본된 클러스터 별칭의 목적은 클러스터 GUI 감시 인터페이스가 현재 운영중인 클러스터 구성원에 연결하는 것입니다. 이와 같이, 만일 어떤 구성원도 운영중이 아니더라도 클러스터의 상태를 구성원을 지정하지 않은 상태에서 얻어낼 수 있습니다.

cluconfig를 실행하는 중, 클러스터 별칭을 사용할지 여부를 질문을 받을 것입니다. 질문은 다음과 같습니다:

```
Enter IP address for cluster alias [NONE]: 172.16.33.105
```

앞에서와 같이, 기본값은 **NONE**으로 지정되어 있습니다, 이말은 클러스터 별칭이 없다는 것입니다, 하지만, 사용자가 이 기본값 대신에 IP 주소인 172.16.33.105를 사용하게 할 수 있습니다. 이곳에 사용된 IP 주소는 클러스터 구성원의 호스트명에 사용된 것과는 다릅니다. 또한 다른 클러스터 서비스에서 사용된 IP 주소와도 다릅니다.

3.1.3. 원격 감시 활성화하기

cluconfig를 실행하는 도중, 클러스터 설정 변수를 지정해야 합니다. 프로그램이 다음과 같은 질문을 할 것입니다:

```
Do you wish to enable monitoring, both locally and remotely, via \
  the Cluster GUI? yes/no [yes]:
```

yes라 대답함으로 클러스터 GUI를 통해 지역과 원격 모니터링을 할 수 있습니다. 현재는 보안 규정 제어 클러스터 모니터링만 사용할 수 있습니다. 클러스터 GUI는 모니터링 요구만 수행 가능하며 어떠한 활성 설정을 변경할 수 없습니다.

no라 대답하면, 클러스터 GUI의 사용을 전면 금지합니다.

3.1.4. cluconfig 예제

이 곳에서는 클러스터 설정 프로그램인 **cluconfig**의 자세한 예제를 담고 있습니다. 이 프로그램은 클러스터 구성원에 대한 자세한 내용을 물으며, 클러스터 데이터베이스에 입력합니다. 이 내용의 복사본은 `cluster.conf` 파일 안에 남습니다. 이 예제에서는 다음과 같은 설정을 **cluconfig**의 질문에 입력하게 됩니다:

- **storage0** 클러스터 시스템:
  Ethernet heartbeat channels: **storage0**
  Power switch serial port: **/dev/ttyC0**
  Power switch: **RPS10**
  Quorum partitions: **/dev/raw/raw1** and **/dev/raw/raw2**
- **storage1** 클러스터 시스템:
  Ethernet heartbeat channels: **storage1** and **cstorage1**
  Serial heartbeat channel: **/dev/ttyS1**
  Power switch serial port: **/dev/ttyS0**
  Power switch: **RPS10**
  Quorum partitions: **/dev/raw/raw1** and **/dev/raw/raw2**
- 클러스터 별칭에 사용될 IP 주소: **10.0.0.154**

```
/sbin/cluconfig

Red Hat Cluster Manager Configuration Utility (running on storage0)

- Configuration file exists already.
  Would you like to use those prior settings as defaults? (yes/no) [yes]: yes
Enter cluster name [Development Cluster]:
  Enter IP address for cluster alias [10.0.0.154]: 10.0.0.154

--------------------------------
```

```
Information for Cluster Member 0
--------------------------------
 Enter name of cluster member [storage0]: storage0
Looking for host storage0 (may take a few seconds)...

Enter number of heartbeat channels (minimum = 1) [1]: 1
Information about Channel 0
Channel type: net or serial [net]:
Enter hostname of the cluster member on heartbeat channel 0 \
 [storage0]: storage0
Looking for host storage0 (may take a few seconds)...

Information about Quorum Partitions
Enter Primary Quorum Partition [/dev/raw/raw1]: /dev/raw/raw1
Enter Shadow Quorum Partition [/dev/raw/raw2]: /dev/raw/raw2

Information About the Power Switch That Power Cycles Member 'storage0'
Choose one of the following power switches:
 o NONE
 o RPS10
 o BAYTECH
 o APCSERIAL
 o APCMASTER
 o WTI_NPS
Power switch [RPS10]: RPS10
Enter the serial port connected to the power switch \
 [/dev/ttyS0]: /dev/ttyS0

--------------------------------
Information for Cluster Member 1
--------------------------------
Enter name of cluster member [storage1]: storage1
Looking for host storage1 (may take a few seconds)...

Information about Channel 0
Enter hostname of the cluster member on heartbeat channel 0 \
 [storage1]: storage1
Looking for host storage1 (may take a few seconds)...

Information about Quorum Partitions
Enter Primary Quorum Partition [/dev/raw/raw1]: /dev/raw/raw1
Enter Shadow Quorum Partition [/dev/raw/raw2]: /dev/raw/raw2

Information About the Power Switch That Power Cycles Member 'storage1'
Choose one of the following power switches:
 o NONE
 o RPS10
 o BAYTECH
 o APCSERIAL
 o APCMASTER
 o WTI_NPS
Power switch [RPS10]: RPS10
Enter the serial port connected to the power switch \
 [/dev/ttyS0]: /dev/ttyS0

Cluster name: Development Cluster
Cluster alias IP address: 10.0.0.154
Cluster alias netmask: 255.255.254.0

Serial port connected to the power switch \
 [/dev/ttyS0]: /dev/ttyS0
```

```
Cluster name: Development Cluster
Cluster alias IP address: 10.0.0.154
Cluster alias netmask: 255.255.254.0

--------------------
Member 0 Information
--------------------
Name: storage0
Primary quorum partition: /dev/raw/raw1
Shadow quorum partition: /dev/raw/raw2
Heartbeat channels: 1
Channel type: net, Name: storage0
Power switch IP address or hostname: storage0
Identifier on power controller for member storage0: storage0
--------------------
Member 1 Information
--------------------
Name: storage1
Primary quorum partition: /dev/raw/raw1
Shadow quorum partition: /dev/raw/raw2
Heartbeat channels: 1
Channel type: net, Name: storage1
Power switch IP address or hostname: storage1
Identifier on power controller for member storage1: storage1

--------------------------
Power Switch 0 Information
--------------------------
Power switch IP address or hostname: storage0
Type: RPS10
Login or port: /dev/ttyS0
Password: 10
--------------------------
Power Switch 1 Information
--------------------------
Power switch IP address or hostname: storage1
Type: RPS10
Login or port: /dev/ttyS0
Password: 10

Save the cluster member information? yes/no [yes]:
Writing to configuration file...done
Configuration information has been saved to /etc/cluster.conf.
----------------------------
Setting up Quorum Partitions
----------------------------
Running cludiskutil -I to initialize the quorum partitions: done
Saving configuration information to quorum partitions: done
Do you wish to enable monitoring, both locally and remotely, via the \
  Cluster GUI? yes/no [yes]: yes

------------------------------------------------------------------

Configuration on this member is complete.

To configure the next member, invoke the following command on that system:

# /sbin/cluconfig --init=/dev/raw/raw1

See the manual to complete the cluster installation
```

3.2. 클러스터 설정 검사하기

클러스터 소프트웨어가 제대로 설정되었는지 알기 위해 /sbin안에 있는 다음과 같은 도구들을 사용합니다:

- Quorum 파티션들을 검사하고 사용 가능한지 검사합니다.

  cludiskutil를 -t 옵션과 함께 quorum 파티션에 사용합니다. 3.2.1 절에 좀더 자세한 내용이 있습니다.

- 전원 스위치들의 상태를 검사합니다.

  만일 전원 스위치가 클러스터 하드웨어 설정에 사용되었다면, clustonith를 모든 클러스터 시스템에 실행하여 원격으로 전원-사이클을 다른 클러스터 시스템에 사용할 수 있는지 확인합니다. 클러스터 소프트웨어가 실행 중일때, 이 명령어를 사용하지 마십시요. 자세한 내용은 3.2.2 절를 참조 하십시요.

- 모든 클러스터 시스템에 같은 소프트웨어 버전이 실행중인 것을 확인합니다.

  rpm -q clumanager를 모든 클러스터 시스템에 실행하여 클러스터 RPM 관련 정보를 얻습니다.

다음은 클러스터 프로그램에 대해 보다 자세히 설명하고 있습니다.

3.2.1. Quorum 파티션 검사하기

Quorum 파티션은 모든 클러스터 시스템의 같은 장치를 지시하고 있어야 합니다. cludiskutil 프로그램을 -t 옵션과 함께 사용하여, quorum 파티션을 검사하고 이용 가능한지 확인합니다.

만일 명령이 성공적이면, cludiskutil -p를 모든 클러스터 시스템에 실행하면 quorum 파티션에 있는 헤더 데이타 체제의 요약이 보입니다. 만일 시스템들의 출력이 다를 경우, quorum 파티션이 같은 장치를 지시하고 있지 않습니다. 원장치가 있는지, 그리고 /etc/sysconfig/rawdevices 파일에 정확하게 명시되어 있는지 확인합니다. 2.4.4.3 절에 자세한 내용이 있습니다.

다음 예제는 quorum 파티션이 두 클러스터 시스템 (devel0와 devel1)의 같은 장치를 지시하고 있는것을 나타냅니다:

```
/sbin/cludiskutil -p
----- Shared State Header ------
Magic# = 0x39119fcd
Version = 1
Updated on Thu Sep 14 05:43:18 2000
Updated by node 0
--------------------------------


/sbin/cludiskutil -p
----- Shared State Header ------
Magic# = 0x39119fcd
Version = 1
Updated on Thu Sep 14 05:43:18 2000
Updated by node 0
--------------------------------
```

모든 클러스터 설정에서 **Magic#** 와 **Version** 부분은 같습니다. 마지막 두 줄은 quorum 파티션이 cludiskutil -I를 통해 기초화 된 날짜를 가르치며, 기초화 명령을 실행한 클러스터 시스템의 번호를 나타냅니다.

만일 cludiskutil에 -p 옵션을 더한 출력이 양쪽 클러스터 시스템이 같지 않다면 다음과 같은 행동을 취하십시요:

- 모든 클러스터 시스템의 /etc/sysconfig/rawdevices파일을 확인한 후, 본 quorum과 백업 quorum 파티션이 있는 원 문자 장치와 블럭 장치가 제대로 지정되었는지 확인합니다. 만일 서로 같지 않다면, 파일을 수정하여 실수를 정정합니다. 그리고 다시 **cluconfig**를 실행합니다. 더 자세한 내용은 3.1.1 절를 참조하십시요.

- 각 클러스터 시스템에 꼭 quorum 파티션을 위한 원 장치를 생성하십시요. 자세한 것은 2.4.4.3 절를 참조하십시요.
- 각 클러스터 시스템에, SCSI 섭시스템이 버스 설정을 찾는 곳의 시스템 시작 메세지를 확인하십시요. 모든 클러스터 시스템이 같은 공유 저장 장치를 인식하며, 같은 이름으로 지정하는 지를 확인합니다.
- 클러스터 시스템이 quorum 파티션에 파일 시스템을 장착하려 하지 않는지 확인 하십시요. 예를 들어, 원래 장치 (예를 들어, /dev/sdb1)이 /etc/fstab 파일에 포함이 안되어 있도록 합니다.

위의 작업을 수행한 후, 다시 cludiskutil 명령어에 -p 옵션을 더해 실행합니다.

3.2.2. 전원 스위치 검사하기

만일 클러스터 하드웨어 설정에, 네트워크 기반 혹은 벙럴 기반의 전원 스위치가 사용되었다면, 클러스터 소프트웨어를 설치한 후, clustonith 를 실행하여 전원 스위치를 검사합니다. 모든 클러스터 시스템에 실행하여, 다른 클러스터 시스템에 전원 사이클을 원격으로 사용할 수 있는지 확인합니다. 만일 검사가 성공적이면, 클러스터가 시작될 수 있습니다. 만일 감시 타이머 혹은 스위치의 유형이 "None"인 경우, 이 검사를 제외해도 됩니다.

clustonith는 클러스터 소프트웨어가 실행 중이지 않을 시에만 정확하게 전원 스위치의 검사를 할 수 있습니다. 이것은 벙럴 기반의 스위치 때문으로, 한 프로그램만 한번에 벙럴 포트를 사용할 수 있습니다. clustonith 프로그램이 실행되면, 클러스터 소프트웨어의 상태를 점검합니다. 만일 클러스터 소프트웨어가 실행 중이면, 클러스터 소프트웨어의 정지를 권장하는 메세지를 보내며 종료합니다.

clustonith 명령어의 포맷은 다음과 같습니다:

```
clustonith [-sSlLvr] [-t devicetype] [-F options-file] \
  [-p stonith-parameters]
Options:
-s      Silent mode, supresses error and log messages
-S      Display switch status
-l      List the hosts a switch can access
-L      List the set of supported switch types
-r hostname   Power cycle the specified host
-v      Increases verbose debugging level
```

전원 스위치를 검사할 때는, 처음 단계가 각 클러스터 시스템이 연결된 전원 스위치와 제대로 통신할 수 있는지를 확인합니다. 다음은 clustonith의 통신 출력의 예로 각 클러스터 구성원이 이들의 전원 스위치와 통신할 수 있음을 나타냅니다:

```
 clustonith -S
WTI Network Power Switch device OK.
An example output of the clustonith command when it is unable
to communicate with its power switch appears below:
 clustonith -S
Unable to determine power switch type.
Unable to determine default power switch type.
```

위에 나타난 에러는 다음과 같은 문제일 수 있습니다:

- 벙럴 기반의 전원 스위치:
  - 원격 전원 스위치 연결 벙럴 포트 (예를 들어, /dev/ttyS0)를 위한 장치 특별 파일이 cluconfig 명령을 통해 주어진 것처럼 클러스터 데이터베이스에 제대로 지정되어 있는지 확인합니다. 만일 필요하다면, 클러스터 시스템이 벙럴 포트를 제대로 사용할 수 있는지 **minicom**과 같은 터미널 에뮬레이션 패키지를 사용할 수 있습니다.
  - 원격 전원 스위치 연결을 위해 클러스터 기반 프로그램이 아닌 (예를 들어, getty 같은) 프로그램이 벙럴 포트를 사용하지 않는지 확인합니다. 이것을 위해 lsof를 사용할 수 있습니다.

- 전원 스워치로의 케이블 연결을 확인합니다. 바른 유형의 케이블이 사용되었는지 (예를 들어, RPS-10 전원 스위치는 Null 모뎀 케이블을 사용합니다), 또한 확실히 연결되어 있는지 확인합니다.
- 전원 스위치에 딥 스위치나 로타리 스위치가 제대로 설정되어 있는지 확인합니다. 만일 RPS-10 전원 스위치를 사용한다면, A.1.1 절을 참조하십시오.

- 네트워크 기반의 전원 스위치:
  - 네트워크 기반의 네트워크 연결이 제대로 되어 있는지 확인합니다. 거의 대부분의 스위치에는 link 불이 있어서, 연결을 보여줍니다.
  - 네트워크 스위치를 `ping` 가능할 것입니다; 만일 그렇지 않다면, 전원 스위치에서 네트워크 번수에 대하여 제대로 설정되지 않은 것입니다.
  - 클러스터 설정 데이터베이스에 정확한 비밀번호와 로그인 이름이 (스위치 유형에 따라 다름) 지정되어 있는지 확인합니다 (`cluconfig` 실행으로 정해집니다). 도움이 될 진단 방법은 클러스터 설정에 사용된 똑같은 번수를 사용해서 네트워크 스위치에 **telnet**을 시도해 봅니다.

스위치와의 성공적인 통신을 확인한 후, 다른 클러스터 구성원에 전원 사이클을 시도해 봅니다. 이것을 하기 전에, 다른 클러스터 구성원이 중요한 일 (예를 들어, 클라이언트들에게 클러스터 서비스를 제공하고 있는지)을 하는지 확인합니다. 다음은 성공적인 전원 사이클 실행을 보여줍니다.

```
clustonith -r clu3
Successfully power cycled host clu3.
```

3.2.3. 클러스터 소프트웨어 버전 보기

`rpm -qa clumanager`을 실행하여, 현재 설치되어 있는 클러스터 RPM의 정보를 얻습니다. 모든 클러스터 시스템이 같은 버전을 사용하고 있는지 확인합니다.

3.3. syslog 이벤트 기록 설정하기

클러스터가 `/var/log/messages`와 다른 파일에 이벤트 기록하도록 `/etc/syslog.conf` 파일을 수정할 수 있습니다. 다른 파일에 클러스터 메세지들을 저장함으로 문제 발생시 문제 진단을 쉽게 할 수 있습니다.

클러스터 시스템들은 **syslogd** 데몬을 사용해서 클러스터 관련 이벤트들을 `/etc/syslog.conf`에 지정한 한 파일에 저장하도록 합니다. 그 로그 파일은 클러스터 안에 생기는 문제의 진단을 담고 있습니다. **syslogd** 데몬을 사용하여, 실행중인 시스템의 클러스터 메세지들을 로깅하도록 하는 것을 권장합니다. 그렇게 하여, 클러스터에 관련해서 정확한 것을 알려면, 모든 클러스터 시스템의 로그 파일을 살펴보아야 합니다.

syslogd 데몬은 다음과 같은 클러스터 데몬의 메시지를 로그합니다:

- `cluquorumd` — Quorum 데몬
- `clusvcmgrd` — 서비스 관리 데몬
- `clupowerd` — 전원 데몬
- `cluhbd` — Heartbeat 데몬
- `clumibd` — 관리차원 시스템 검사 데몬

이벤트의 중요함은 로그 항목의 보안 수준을 결정합니다. 중요한 이벤트는 클러스터의 사용에 문제가 생기기 전에 확인해 보아야 합니다. 클러스터는 다음과 같은 보안 수준에 따라 메세지를 기록할 수 있습니다:

- **emerg** — 클러스터 시스템이 사용 불가능합니다.

- **alert** — 문제를 확인하러 행동을 취해야 합니다.
- **crit** — 심각한 문제가 생겼습니다.
- **err** — 에러가 생겼습니다.
- **warning** — 주의를 요하는 중요한 이벤트가 생겼습니다.
- **notice** — 시스템의 활동에 문제를 주지 않는 이벤트가 생겼습니다.
- **info** — 보통 클러스터 활동이 생겼습니다.
- **debug** — 보통 클러스터 활동의 진단 관련 출력입니다.

클러스터 데몬의 기본 기록 보안 수준은 **warning** 이상입니다.

로그 파일의 예제는 다음과 같습니다:

```
May 31 20:42:06 clu2 clusvcmgrd[992]: <info> Service Manager starting
May 31 20:42:06 clu2 clusvcmgrd[992]: <info> mount.ksh info: /dev/sda3 \
 is not mounted
May 31 20:49:38 clu2 clulog[1294]: <notice> stop_service.ksh notice: \
 Stopping service dbase_home
May 31 20:49:39 clu2 clusvcmgrd[1287]: <notice> Service Manager received \
 a NODE_UP event for stor5
Jun 01 12:56:51 clu2 cluquorumd[1640]: <err> updateMyTimestamp: unable to \
 update status block.
Jun 01 12:34:24 clu2 cluquorumd[1268]: <warning> Initiating cluster stop
Jun 01 12:34:24 clu2 cluquorumd[1268]: <warning> Completed cluster stop
Jul 27 15:28:40 clu2 cluquorumd[390]: <err> shoot_partner: successfully shot partner.
  [1]    [2]   [3]      [4]        [5]
```

로그 파일의 각 항목에는 다음과 같은 내용이 담겨 있습니다:

- [1]시간 도장
- [2] 이벤트가 로그된 클러스터 시스템
- [3] 이벤트가 나타난 하부시스템
- [4] 이벤트의 보안 수준
- [5] 이벤트 관련 설명

클러스터 소프트웨어를 설정한후, `/etc/syslog.conf`를 설정하여 클러스터가 기본 로그 파일인, `/var/log/messages`에서 다른 파일에 이벤트를 로그하도록 할 수 있습니다.클러스터 프로그램과 데몬은 각자 *`local4`* 라는 syslog 테그를 사용해 메세지들을 저장합니다. 클러스터 관련 로그 파일은 클러스터 감시와 문제 진단에 사용할 수 있습니다. 클러스터 이벤트를 `/var/log/cluster`와 `/var/log/messages` 파일에 로그하려면, `/etc/syslog.conf` 파일에 다음과 비슷한 내용을 더합니다:

```
#
# Cluster messages coming in on local4 go to /var/log/cluster
#
local4.*           /var/log/cluster
```

`/var/log/cluster` 파일에 중복되는 메세지들과 클러스터 이벤트 메세지의 로그를 방지하려면, `/etc/syslog.conf` 파일에 다음과 비슷한 내용을 더합니다:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;local4.none  /var/log/messages
```

앞의 내용들을 적용하려면, `killall -HUP syslogd`를 실행하거나 `/etc/rc.d/init.d/syslog restart`를 통해 syslog를 재시작 해야합니다.

이에 덧붙여, 각 클러스터 데몬들이 필요한 심각성 레벨에 맞추어서 이벤트들을 로그 하도록 수정할 수 있습니다. 자세한 내용은 8.6 절을 참조하십시오.

3.4. cluadmin 프로그램 사용하기

cluadmin 프로그램은 명령어 기반 사용자 인터페이스를 제공하여 관리자로 클러스터 시스템과 서비스를 모니터링 관리할 수 있도록 도와줍니다. cluadmin 프로그램을 다음과 같은 일을 하는데 사용할 수 있습니다:

- 서비스 더하기, 수정하기, 삭제하기
- 서비스 활성화 혹은 비활성화하기
- 클러스터와 서비스 상태 보이기
- 클러스터 데몬 이벤트 기록 수정하기
- 클러스터 데이터베이스 백업과 재구축하기

클러스터는 클러스터 데이터베이스가 동시에 클러스터 시스템에 있는 여러명의 사용자에 의해 수정되는 것을 방지하기 위해 권고형 잠금을 사용합니다. 사용자는 그 권고형 잠금을 소유하지 않으면 데이터베이스를 수정할 수 없습니다.

cluadmin이 사용될 때, 클러스터 소프트웨어는 사용자에게 그 잠금이 주어졌는지 확인합니다. 만일 잠금이 다른 사용자에게 주어졌다면, 클러스터 소프트웨어는 잠금을 요구하게 됩니다. 사용자가 cluadmin에서 나가면, 잠금이 양도됩니다.

만일 다른 사용자가 잠금을 가지고 있으면, 데이터베이스에 잠금이 있다고 주의가 나타납니다. 클러스터 소프트웨어는 잠금 갖는 것을 허용합니다. 만일 잠금이 현재 요구하는 사용자에게 주어질 경우, 전의 잠금 사용자는 클러스터 데이터베이스를 수정할 수 없습니다.

필요할 때만, 잠금을 가져가십시요, 왜냐하면, 정리되지 않은 동시 설정은 클러스터의 예측하지 못할 결과를 유도할 수 있습니다. 덧붙여, 클러스터 데이터베이스에 한번에 한가지 (예를 들어, 서비스 더하기, 수정하기와 삭제하기) 수정만 하십시요. cluadmin 프로그램의 명령 옵션은 다음과 같습니다:

`-d` or `--debug`

‘ 자세한 진단 관련 내용을 보여줍니다.

`-h`, `-?`, or `--help`

‘ 프로그램 관련된 도움말을 보여줍니다.

`-n` or `--nointeractive`

‘ cluadmin의 가장 기본의 loop 프로세스를 지나갑니다. 이 옵션은 cluadmin의 디버깅을 위해 사용됩니다.

`-t` or `--tcl`

‘ **Tcl** 명령을 cluadmin의 기본 해석기에 더합니다. **Tcl** 명령어를 프로그램의 내부 **Tcl** 해석기에 전하려면, cluadmin> 프롬프트에서, **Tcl** 명령어를 tcl을 앞에 명시하고 사용합니다. 이것은 cluadmin의 디버깅에 사용됩니다.

`-V` or `--version`

‘ cluadmin의 버전을 나타냅니다.

cluadmin 프로그램에 -n 옵션이 없이 사용이 되면, cluadmin>이 나타납니다. 여기에 명령어와 하부 명령어를 지정할 수 있습니다. 표 3-1에 cluadmin의 명령어와 하부 명령어가 설명되어 있습니다:

cluadmin 명령어	cluadmin 하부 명령어	설명	예
help	None	지정된 cluadmin의 명령어 혹은 하부 명령어.	help service add
cluster	status	현재 클러스터의 상태를 보여줍니다. 보다 자세한 내용은 8.1 절을 참조하시기 바랍니다.	cluster status
	loglevel	지정된 클러스터 데몬 기록의 보안 수준을 특정 수준으로 설정합니다. 8.6 절을 참조하십시오.	cluster loglevel cluquorumd 7
	reload	클러스터 데몬에게 강제로 클러스터 설정 데이터베이스를 재구축하도록 합니다. 자세한 내용은 8.8 절을 참조하십시오.	cluster reload
	name	클러스터의 이름을 지정된 이름으로 설정합니다. 클러스터 이름은 클러스터 모니터링 명령어인 clustat의 출력에 포함되어 있습니다. 자세한 내용은 8.9 절을 참조하십시오.	cluster name dbasecluster
	backup	클러스터 설정 데이터베이스를 /etc/cluster.conf.bak 파일에 저장합니다. 자세한 내용은 8.5 절을 참조하십시오.	cluster backup
	restore	클러스터 설정 데이터베이스를 백업인 /etc/cluster.conf.bak에서 재구축 합니다. 자세한 내용은 8.5 절을 참조하십시오.	cluster restore
	saveas	클러스터 설정 데이터베이스를 지정된 파일에 저장합니다. 자세한 사항은 8.5 절을 참조하십시오.	cluster saveas cluster_backup.conf
	restorefrom	클러스터 설정 데이터베이스를 지정된 파일로부터 재구축합니다. 자세한 것은 8.5 절을 참조 바랍니다.	cluster restorefrom cluster_backup.conf
service	add	클러스터 서비스를 클러스터 데이터베이스에 추가합니다. 이 명령어는 서비스에 관련된 설정과 자원에 대한 정보를 요구합니다. 자세한 내용은 4.1 절을 참조하십시요.	service add
	modify	지정된 서비스에 관련된 자원과 자료를 수정합니다. 서비스를 생성시 입력했던 모든 자원과 관련 자료를 수정할 수 있습니다. 자세한 내용은 4.5 절을 참조하십시요.	service modify dbservice
	show state	모든 서비스나 지정된 서비스의 현재 상태를 보여줍니다. 자세한 내용은 8.1 절을 참조하십시오.	service show state dbservice

cluadmin 명령어	cluadmin 하부 명령어	설명	예
	relocate	서비스를 현재 실행 중인 클러스터 구성원에서 정지하고 다른 구성원에서 재시작하도록 합니다. 자세한 내용은 4.6 절을 참조하십시오.	service relocate nfs1
	show config	지정된 서비스의 현재 설정을 보여줍니다. 자세한 내용은 4.2 절을 참조하십시요.	service show config dbservice
	disable	지정된 서비스를 정지시킵니다. 서비스를 사용하려면, 서비스를 활성화 시켜야 합니다. 자세한 내용은 4.3 절을 참조하십시요.	service disable dbservice
	enable	지정된 비활성화된 서비스를 시작합니다. 자세한 내용은 4.4 절을 참조하십시오.	service enable dbservice
	delete	지정된 서비스를 클러스터 설정 데이터베이스로부터 삭제합니다. 자세한 내용은 4.7 절을 참조하십시오.	service delete dbservice
apropos	None	지정된 글자로 시작하는 cluadmin 명령들을 보이거나, 만일 지정되지 않았다면 모든 cluadmin 명령어를 출력합니다.	apropos service
clear	None	화면 출력을 지웁니다.	clear
exit	None	cluadmin에서 나갑니다.	exit
quit	None	cluadmin에서 나갑니다.	quit

표 3-1. cluadmin 명령어

cluadmin 프로그램을 사용 중 [Tab] 키를 사용해 cluadmin의 명령어들을 확인할 수 있습니다. 예를 들어, cluadmin>프롬프트에서 [Tab] 키를 누르시면 사용 가능한 모든 명령어를 보여줍니다. 프롬프트에서 글자 하나를 입력하고 [Tab] 키를 사용하면, 그 글자로 시작하는 명령어가 나타납니다. 명령어를 지정하고 [Tab] 키를 사용하면, 지정된 명령어와 사용할 수 있는 하부 명령어들이 나타납니다.

사용자는 프롬프트에서 위, 아래 화살표를 사용함으로 이전에 사용한 명령어들을 찾아볼 수 있습니다. 이 history들은 사용자의 홈디렉토리의 .cluadmin_history 파일에 저장되어 있습니다.

4장 .

서비스 설정과 관리

다음에서는 서비스를 어떻게 설정, 보이며, 활성/비활성, 수정, 재배치, 그리고 삭제하는지와 더불어 서비스가 시작하지 않을시 어떻게 회복하는지에 대해 설명하고 있습니다.

4.1. 서비스 설정하기

서비스를 설정 이전에 클러스터 시스템이 먼저 준비되어야 합니다. 예를 들어, 저장 디스크를 준비한다든지, 서비스에서 사용될 응용 프로그램들을 준비하여야 합니다. 그 후, 서비스 관련 내용들과 필요한 자원들을 `cluadmin`를 사용해 클러스터 데이터 베이스에 입력하여야 합니다. 이 내용들은 서비스를 시작하고 정지하는데 변수로 사용될 것입니다.

서비스를 설정하려면, 다음과 같은 단계를 따르십시요:

1. 만일 필요하다면, 서비스에서 사용되는 응용 프로그램을 시작하고 정지할 수 있는 스크립트를 만드십시요. 4.1.2 절에서 보다 많은 정보를 찾으실 수 있습니다.
2. 서비스 자원과 서비스 관련 자료를 수집합니다. 4.1.1 절을 참조하십시요.
3. 파일 시스템을 설정하거나 서비스가 사용할 원장치(Raw device)를 준비합니다. 4.1.3 절을 참조하십시요.
4. 양 클러스터 시스템에서 필요한 응용 프로그램들이 실행될 수 있는지 확인하시고, 만일 필요하다면, 서비스 스크립트가 서비스 응용 프로그램들을 시작, 정지시킬수 있는지 확인하시기 바랍니다. 4.1.4 절을 참조 하십시요.
5. `/etc/cluster.conf` 파일을 백업합니다. 8.5 절을 참조하시기 바랍니다.
6. `cluadmin`를 시작 후, `service add`를 지정합니다. `cluadmin`가 2번째 단계에서 언급한 서비스 자원과 서비스 관련 내용을 필요로 할 것 입니다. 설정 체크들을 통과하면, 사용자가 서비스를 비활성화 시키지 않는 한, 서비스를 사용자 지정한 클러스터 시스템에서 시작할 것입니다. 예를 들어:
   ```
   cluadmin> service add
   ```

클러스터 서비스 추가에 대한 보다 많은 정보를 원하신다면, 다음을 참조하시기 바랍니다:

- 5.1 절
- 5.3 절
- 5.4 절
- 6.1 절
- 6.2 절
- 7.1 절

4.1.1. 서비스 관련 자료 모으기

서비스를 만들기 전, 서비스 자원과 서비스 관련 자료를 모으는 것이 필요합니다. 서비스를 클러스터 데이터 베이스에 추가시, `cluadmin`가 이 정보를 요구할 것입니다.

어떤 경우에는, 한 서비스를 위해 여러 자원을 지정할 수도 있습니다. (예, 다중 IP 주소와 디스크 장치들).

서비스 설정과 자원 정보의 예들이 다음 표에 나와 있습니다.

서비스 설정과 자원	설명
서비스 이름	각 서비스는 특유의 이름을 가져야 합니다. 서비스 이름은 63자까지 가능하며 (대문자와 소문자로)글자, 숫자, 밑줄, 점, '-'로 이루어질 수 있습니다. 그러나, 서비스의 이름은 글자나 밑줄로 시작해야 합니다.
우선권이 있는 구성원	만일 있다면, 수동으로 옮겨지거나, 고장 회복으로 인한 경우를 제외하고 서비스가 어느 시스템에서 실행될지, 클러스터 시스템을 지정합니다
우선권이 있는 구성원 재배치 원칙	이 원칙이 활성화되면, 자동적으로 그 시스템이 클러스터에 연결되었을때 서비스를 우선권이 있는 구성원에게 옮겨줍니다. 만일 이 방침이 비활성화되었다면, 서비스가 계속적으로 비우선 구성원에서 실행될 것입니다. 예를 들어, 만일 관리자가 이 방침을 활성화시키면, 문제가 발생한 우선권이 있는 구성원이 재부팅한 후 클러스터에 다시 연결하면, 서비스가 우선권이 있는 구성원에서 자동으로 재시작될 것 입니다.
스크립트 위치	만일 스크립트가 존재한다면, 서비스가 시작하고 정지하는데 사용될 스크립트의 완전 경로를 지정합니다. 4.1.2 절을 참조하십시요.
IP 주소	
디스크 파티션	서비스에 사용되는 각 공유 디스크 파티션을 지정합니다.
마운트 지점, 파일 시스템 유형, 장착 옵션, NFS export 옵션, 삼바 (samba) 공유	
서비스 체크 간격	얼마나 자주 (초 단위로), 서비스와 관련있는 응용 프로그램들의 상태를 검사하는지 지정합니다. 예를 들어, 필요한 NFS나 삼바 데몬이 실행되고 있는지 검사합니다. 또한, 서비스 유형에 따라, 서비스 스크립트 상의 "status"의 값들을 확인하고 검사합니다. 이 검사 간격을 0으로 둠으로써, 비활성화 시킬 수 있습니다.
비활성화 서비스 방침	만일 사용자가 서비스를 클러스터에 더한 후 서비스가 자동적으로 시작되는 것을 원치 않는다면, 사용자가 활성화 시킬때까지, 비활성화인 상태를 유지하게 만들 수도 있습니다.

표 4-1. 서비스 설정과 자원 정보

4.1.2. 서비스 스크립트 만들기

클러스터는 각 서비스 별로 알맞는 스크립트를 실행함으로써, 그 스크립트에 알맞는 응용 프로그램들을 실행합니다. NFS나 삼바 서비스를 위해서, 관련된 스크립트는 클러스터 안에 구성되어 있습니다. 그렇기에, `cluadmin`를 실행하여 NSF나 삼바를 설정하게 되면, 서비스 스크립트의 이름을 적지 않아도 됩니다. 다른 응용프로그램은 알맞는 스크립트를 지정하여야 합니다. 예를 들어, 데이터베이스 응용 프로그램을 `cluadmin`안에서 지정하려고 하면, 데이터베이스 시작 스크립트의 전체 경로를 적어 주어야 합니다.

서비스 스크립트의 포맷은 System V의 init 스크립트의 방식을 따르고 있습니다. 이 방식은 꼭 `start`, `stop`, 그리고 `status`를 가지도록 하고 있습니다. 위의 명령어들이 성공적으로 실행되지 않을때는 exit 값을 주도록 되어 있습니다. 클러스터는 서비스가 시작하는데 실패할 경우 서비스를 멈출 것입니다. 서비스를 시작하지 못하게 되면, 서비스는 비활성화 되게 됩니다.

시작하거나, 멈추는것에 덛붙여서, 서비스 스크립트에는 응용 서비스들을 감시하는 기능도 가지고 있습니다. 이것은 `status`를 통하여 이루어 집니다. 이것을 활성화 시키려면, 0이 아닌 어떤 값을 `cluadmin`안에서 `Status check interval:` 에 지정을 해 주시면 됩니다. 0이 아닌 값이 이 시스템 감시에 전해지면, 클

러스터는 먼저 응용프로그램을 먼저 실행되고 있던 구성원에서 재시작하려고 할 것입니다. 이 감시 부분은 스크립트 안에 꼭 있어야 하는 것은 아닙니다. 만일 어떤 감시에 관련된 것이 스크립트 안에 없을 경우, 보통의 `status`는 성공했다라는 말만 보이게 될것입니다.

응용 프로그램의 상태 감시는 응용 프로그램의 필요에 따라 편집되거나 다른 변수가 지정될 수 있습니다. 예를 들어, 데이터베이스를 위한 간단한 모니터링은 데이터베이스가 현재 실행 되고 있는지 여부만 보고합니다. 보다 광범위한 모니터링은 원하신다면, 보다 많은 데이터베이스 테이블 질의를 수행하시면 됩니다.

`/usr/share/cluster/doc/services/examples` 디렉토리에는 서비스 스크립트의 예제를 비롯한 서비스 스크립트를 생성하는데 사용되는 템플릿이 포함되어 있습니다. 샘플 스크립트를 보시려면 5.1 절, 5.3 절, 7.1 절, and 5.4 절을 살펴보시기 바랍니다.

4.1.3. 서비스 저장 디스크 설정하기

서비스를 만들기 전에, 서비스에서 사용될 공유 파일 시스템과 원장치를 설정해야 합니다. 2.4.4 절를 참조하십시요.

만일 클러스터 서비스안에 있는 원장치를 사용할 경우, 시스템 시작시 `/etc/sysconfig/rawdevices`을 이용해 장치를 준비해 둘 수 있습니다. 파일을 편집해서 정확히 어떤 원 문자 장치나 원 블럭 장치가 사용될지 지정해 주십시요. 3.1.1 절을 참조하시기 바랍니다.

소프트웨어 RAID에서 주의할 것은, 호스트를 기본한 RAID는 공유 저장 디스크에서 지원하지 않습니다. 그렇기 때문에, 인증된 SCSI 아답터를 기본한 RAID 카드만이 공유 저장 디스크에 사용될수 있습니다.

서버 관리자는 다음의 서비스 저장 디스크 추천을 따르시기 바랍니다:

- 최상의 성능을 위해선, 파일 시스템을 만들때, 4 KB의 블럭 크기를 사용하십시요. 주의 할것은 **mkfs**에 있는 파일 시스템 생성 프로그램에서는 1 KB를 기본으로 사용하며, 이로 인해 **fsck**이 시간이 오래 걸릴 수 있습니다.
- 빠른 오류 복구 시간을 원하신다면, ext3 파일 시스템을 사용하실 것을 권장합니다. 좀 더 자세한 내용은 2.4.4.6 절을 참조하십시오.
- 큰 파일 시스템은, `mount`에 `nocheck`을 사용하여, 모든 그룹 블럭을 검사하는 것을 피하십시요. `nock-eck`을 큰 파일 시스템 마운트에 사용하게 되면, 마운트에 걸리는 시간을 줄일 수 있습니다.

4.1.4. 응용 프로그램과 서비스 스크립트 확인

서비스를 설정하기 전에, 각 시스템에 서비스가 사용할 모든 응용 프로그램을 설치합니다. 응용 프로그램을 설치한 후, 응용 프로그램들이 실행 되는지, 공유 저장 디스크를 사용할수 있는지를 확인해야 합니다. 데이터 변조를 막으려면, 응용 시스템을 양 시스템에서 같이 실행시키지 마십시요.

만일 서비스 프로그램을 시작하고 정지하는데 스크립트를 사용한다면, 양쪽 클러스터 시스템에 스크립트를 설치, 시험해 본후, 스크립트를 사용하여, 프로그램을 시작하고 정지할 수 있는지 확인합니다. 보다 자세한 내용은 4.1.2 절을 참조해 주십시요.

4.2. 서비스 설정 보기

시스템 관리자는 서비스의 설정에 대한 자료를 자세하게 볼 수 있습니다. 다음과 같은 정보가 포함됩니다:

- 서비스 이름
- 서비스가 시작된 후 비활성화 되었는지의 여부
- 우선권을 가진 구성원 시스템
- 우선권을 가진 구성원이 클러스터에 연결되었을때 시스템의 재배치 여부

- 서비스 감시 시간단위
- 서비스 시작 스크립트의 위치 IP 주소
- 디스크 파티션들
- 파일 시스템 유형
- 장착점과 장착 변수
- NFS exports
- 삼바 공유

클러스터 서비스의 상태를 보시려면, 8.1 절을 참조하십시오.

서비스 설정 자료를 보이려면, `cluadmin`를 실행한 후 `service show config`를 실행하십시요:

```
cluadmin> service show config
 0) dummy
 1) nfs_pref_clu4
 2) nfs_pref_clu3
 3) nfs_nopref
 4) ext3
 5) nfs_eng
 6) nfs_engineering
 c) cancel

Choose service: 6
name: nfs_engineering
disabled: no
preferred node: clu3
relocate: yes
IP address 0: 172.16.33.164
device 0: /dev/sdb11
 mount point, device 0: /mnt/users/engineering
 mount fstype, device 0: ext2
 mount options, device 0: rw,nosuid,sync
 force unmount, device 0: yes
NFS export 0: /mnt/users/engineering/ferris
 Client 0: ferris, rw
NFS export 0: /mnt/users/engineering/denham
 Client 0: denham, rw
NFS export 0: /mnt/users/engineering/brown
 Client 0: brown, rw
cluadmin>
```

만일 서비스의 이름을 알 경우, `service show config` *`service_name`*를 사용해 지정할 수 있습니다.

4.3. 서비스 비활성화 시키기

현재 실행중인 서비스도 서비스를 멈추기 위해 비활성화 될수 있습니다. 비활성화되면, 다시 활성화될 수 있습니다. 자세한 내용은 4.4 절을 참조하십시오.

다음과 같은 경우에 서비스를 비활성화해야 합니다:

- 서비스를 편집할 경우

  서비스를 편집하려면 현재 실행중인 서비스가 비활성화 되어야 합니다. 4.5 절을 참조하십시오.

- 잠시 서비스를 멈출 경우

  서비스는 완전히 자체를 지우지 않고도 클라이언트에게 사용 불가능하게 만들도록 비활성화 될수 있다.

서비스를 비활성화 하려면, cluadmin를 실행한 후, service disable *service_name*를 실행하시면 된다. 예를 들어:

```
cluadmin> service disable user_home
Are you sure? (yes/no/?) y
notice: Stopping service user_home ...
notice: Service user_home is disabled
service user_home disabled
```

4.4. 서비스 활성화하기

비활성화된 서비스는 활성화하여 사용 가능합니다.

비활성화된 서비스를 활성화 하려면, 먼저 cluadmin를 실행하시고 service enable *service_name*를 다음과 같이 실행하시면 됩니다:

```
cluadmin> service enable user_home
Are you sure? (yes/no/?) y
notice: Starting service user_home ...
notice: Service user_home is running
service user_home enabled
```

4.5. 서비스 모니터링

서비스가 만들어 질때 입력된 서비스 관련 자료는 편집될 수 있습니다. 예를 들어, 지정 IP 주소가 바뀔 수 있습니다. 더 많은 자원이 더해질 수도 있습니다 (예, 더 많은 파일 시스템). 4.1.1 절을 참조하시기 바랍니다.

서비스가 편집되기 전에는 꼭 비활성화 되어야 합니다. 현재 실행 중인 서비스를 편집하려 하면, 클러스터 관리 프로그램이 비활성화 시키도록 알릴 것입니다. 4.3 절을 참조하시기 바랍니다.

서비스가 편집될 때는 서비스를 사용할 수 없기에, 서비스의 멈춘 시간을 줄이기 위해 필요한 모든 자료를 얻은후 비활성화 시키십시요. 아울러, 서비스를 편집하시기 전에 클러스터 데이터 베이스를 백업하십시요. 8.5 절을 참조하십시오.

비활성화인 서비스를 편집하려시면, 먼저 cluadmin를 실행하신 후, service modify *service_name*를 실행하시면 됩니다.

```
cluadmin> service modify web1
```

서비스 관련 자료나, 자원은 필요에 따라 편집될 수 있습니다. 클러스터가 서비스의 편집된 자료를 다시 볼것이며, 만일 문제가 있다면, 고치실 수도 있습니다. 클러스터가 바뀐 내용을 검사하고, 문제가 없다면 서비스를 다시 시작할 것입니다. 만일 편집된 내용이 없다면, 원본의 설정을 사용해서 재시작할 것입니다.

4.6. 서비스 재배치하기

서비스의 자동적인 오류 복구 뿐만 아니라, 클러스터는 한 클러스터 시스템에서 서비스를 정지시키고 다른 클러스터 시스템에서 문제없이 재 시작할수 있게 해줍니다. 이 서비스 재배치를 통해, 시스템 관리자가 프로그램들이나 데이타가 클라이언트들의 사용에 전혀 문제 없이 클러스터 시스템의 관리를 할수 있도록 합니다.

cluadmin를 사용해서 서비스를 재배치하려면, service relocate 명령어를 사용하십시요:

```
cluadmin> service relocate nfs1
```

만일 서비스를 지정하지 않으시면, 현재 진행되고 있는 서비스들의 이름이 고를 수 있도록 나타날 것입니다.

만일 서비스 재배치 중 문제가 생기면, 그 서비스를 클러스터 구성원에서 멈추고, 다른 곳에서 시작하려는 방법이 사용될 것입니다.

4.7. 서비스 제거하기

클러스터 서비스는 제거 가능합니다. 주의 하실 것은 클러스터 서비스를 제거하시기 전에 클러스터 데이터베이스를 백업하시기 바랍니다. 자세한 것은 8.5 절을 참조 하시기 바랍니다.

cluadmin를 통해 서비스를 제거하시려면, 다음과 같은 단계를 따르십시요:

1. cluadmin를 서비스가 실행되고 있는 클러스터에 실행하신후, service disable *service_name* 명령을 실행하십시오. 4.3 절을 참조하시기 바랍니다.
2. service delete *service_name*에 서비스 이름을 지정함으로 서비스를 제거하실 수 있습니다.

예를 들어:

```
cluadmin> service disable user_home
Are you sure? (yes/no/?) y
notice: Stopping service user_home ...
notice: Service user_home is disabled
service user_home disabled

cluadmin> service delete user_home
Deleting user_home, are you sure? (yes/no/?): y
user_home deleted.
cluadmin>
```

4.8. 시작하지 않는 서비스 고치기

클러스터가 서비스를 성공적으로 시작할 수 없으면, 서비스를 **disabled** 상태라 부릅니다. **disabled** 상태는 여러가지 상황을 통해 나타날 수 있지만, 이런 서비스는 성공적으로 시작하지 않았거나, 계속적으로 서비스가 정지되거나 실패 했기 때문일 수 있습니다.

시작하는데 실패한 서비스를 다룰 경우 주의하셔야 합니다. 만일 서비스 자원이 현 시스템에 설정이 되어있다면, 서비스를 다른 시스템에서 시작하려 하면 더 큰 문제를 일으킬 수 있습니다. 예를 들어, 만일 현 시스템에 파일 시스템이 장착되어 있는 경우, 만일 다른 클러스터 시스템에서 서비스를 시작하려 한다면, 파일 시스템이 양쪽 시스템에 장착이 될 것이며, 이것을 통해 파일 시스템의 문제를 초래할 수 있습니다. 만일 활성화에 실패하면, 서비스는 **disabled** 상태에 있을 것입니다.

disabled 상태에 있는 서비스는 편집할 수 있습니다. **disabled** 상태를 야기시킨 문제를 해결하기 위해서는 서비스 편집이 불가피합니다. 서비스는 편집된 후, 현 시스템에서 활성화되거나 **disabled** 상태로 남아 있을 것입니다. 다음은 서비스 실패의 경우에 따르셔야할 단계들 입니다:

1. 디버깅 메시지를 기록하도록 클러스터 이벤트 로그를 수정합니다. 8.6 절을 참조하십시오.
2. cluadmin 명령을 사용하여 서비스를 소유한 클러스터 시스템 상에서 서비스를 활성화 또는 비활성화 시킵니다. 4.3 절와 4.4 절을 참조하시기 바랍니다.
3. 만일 서비스가 주 클러스터에서 시작/정지하지 않을 경우, /var/log/messages의 로그 파일을 확인하시고 문제를 바로 고치시기 바랍니다. 클러스터 데이터베이스에 있는 서비스 관련 자료를 편집해야 할 경우도 있습니다 (예, 정확치 못한 시작 스크립트), 혹은 서비스를 소유한 시스템 상에서 직접 작업을 수행하셔야할 경우도 있습니다 (예, 파일 시스템 마운트해제).
4. 주 클러스터 시스템에서 서비스를 활성화/비활성화 하십시요. 만일 재시도에서 실패할 경우, 주 클러스터 시스템을 재부팅해 보십시요.
5. 만일 아직도 서비스를 시작하실 수 없다면, 클러스터 밖에서는 서비스가 시작될 수 있는지 확인 하십시요. 예를 들어, 수동으로 파일 시스템을 마운트 한다던지, 수동으로 시작 스크립트를 시작해 보시기 바랍니다.

5장 .

데이터베이스 서비스

이 장에서는 Red Hat Linux 어드밴스 서버에서 고성능 데이터베이스를 설정하는 방법을 서술하고 있습니다.


주의

다음은 특정 데이터베이스 설정을 보여주고 있습니다. 각 데이터베이스의 새 버전에서는 설정 방법이 조금씩 다를 수도 있다는 것을 염두에 두시기 바랍니다. 그렇기에, 이 자료들이 정확치 않을 수도 있습니다.

5.1. 오라클 서비스 설정하기

데이터베이스 서비스를 통해, 데이터베이스 응용 프로그램에 고성능 데이터를 제공할 수 있습니다.. 그 후에, 웹 서버와 같은 네트워크를 이용해 데이터베이스 프로그램은 자료를 제공할 수 있습니다. 만일 서비스가 문제 복구에 들어가면, 프로그램은 새로운 클러스터 시스템을 통해 공유 데이터베이스의 자료를 이용합니다. 네트워크를 이용하는 데이터베이스에는 일반적으로 IP 주소가 주어지며, 이 IP 주소는 문제 복구에 들어갈 시에 서비스와 더불어 새로운 클러스터로 이동하므로 클라이언트들은 문제없이 접근할 수 있습니다.

이 곳에서는 오라클 데이터베이스를 클러스터 서비스로 설정하는 방법이 나옵니다. 이 곳에 나오는 변수들은 특정 오라클 설정에 따라 달라질 수도 있으나, 예를 통해 한 환경에서 서비스를 설정하는데 도움이 될 것입니다. 5.2 절에서 서비스 성능을 개선하는 방법에 대한 자료를 얻으실 수 있습니다.

다음의 예제에서는:

- 서비스가 오라클 클라이언트가 사용하도록 한 IP주소를 가지고 있습니다.
- 서비스는 파일 시스템을 두개 장착하고 있으며, 하나는 오라클 소프트웨어를 위해 (`/u01`) 그리고 또 하나는 오라클 데이터베이스를 위해 (`/u02`) 사용되며, 모두 서비스가 더해지기 이전에 설정되었습니다. 합니다.
- 서비스가 더해지기 전에 양쪽 클러스터 시스템에, **oracle** 이라는 오라클 관리자 계정이 만들어 졌습니다.
- 이 예제에서는 Perl DBI Proxy를 통해 네트워크가 사용될 것입니다.
- 관리 디렉토리는 공유 디스크에 있으며, 오라클 서비스 (예를 들어, `/u01/app/oracle/admin/db1`)와 더불어 사용될 것입니다.

오라클 서비스 예제는 `/home/oracle`에 넣어져야 하며, 오라클 관리자 계정이 가지고 있는 5개의 스크립트를 사용하고 있습니다. `oracle` 스크립트는 오라클 서비스를 시작하고 정지하는데 사용됩니다. 서비스를 더할때, 이 스크립트를 지정해야 합니다. 이 스크립트는 또한 다른 오라클 예제 스크립트를 실행시킵니다. `startdb`와 `stopdb`는 오라클을 시작하고 정지시키는데 사용되는 스크립트 입니다. `startdbi`와 `stopdbi`는 오라클 데이터베이스에 Perl 스크립트와 데이터베이스와 함께 실행되는 모듈들을 사용하는 웹 프로그램을 시작하고 정지하는데 사용됩니다. 응용 프로그램은 여러 가지 다양한 방식으로 오라클 데이터베이스를 사용할 수 있다는 점을 기억해 두십시오.

다음은 오라클 서비스를 시작하고 정지시킬 `oracle` 스크립트의 예입니다. 이 스크립트가 루트 계정이 아닌 오라클 계정으로 실행되고 있다는 점에 주의해 주십시오.

```
#!/bin/sh
#
# Cluster service script to start/stop oracle
#

cd /home/oracle

case $1 in
```

```
'start')
  su - oracle -c ./startdbi
  su - oracle -c ./startdb
  ;;
'stop')
  su - oracle -c ./stopdb
  su - oracle -c ./stopdbi
  ;;
esac
```

다음은 오라클 데이터베이스 서버를 시작하고 정지시킬 때 사용되는 startdb 스크립트의 예제입니다:

```
#!/bin/sh
#

#
# Script to start the Oracle Database Server instance.
#
########################################################################
#
# ORACLE_RELEASE
#
# Specifies the Oracle product release.
#
########################################################################

ORACLE_RELEASE=8.1.6

########################################################################
#
# ORACLE_SID
#
# Specifies the Oracle system identifier or "sid", which is the name of
# the Oracle Server instance.
#
########################################################################

export ORACLE_SID=TESTDB

########################################################################
#
# ORACLE_BASE
#
# Specifies the directory at the top of the Oracle software product and
# administrative file structure.
#
########################################################################

export ORACLE_BASE=/u01/app/oracle

########################################################################
#
# ORACLE_HOME
#
# Specifies the directory containing the software for a given release.
# The Oracle recommended value is $ORACLE_BASE/product/<release>
#
########################################################################

export ORACLE_HOME=/u01/app/oracle/product/${ORACLE_RELEASE}

########################################################################
```

```
#
# LD_LIBRARY_PATH
#
# Required when using Oracle products that use shared libraries.
#
###############################################################################

export LD_LIBRARY_PATH=/u01/app/oracle/product/${ORACLE_RELEASE}/lib

###############################################################################
#
# PATH
#
# Verify that the users search path includes $ORCLE_HOME/bin
#
###############################################################################

export PATH=$PATH:/u01/app/oracle/product/${ORACLE_RELEASE}/bin

###############################################################################
#
# This does the actual work.
#
# The oracle server manager is used to start the Oracle Server instance
# based on the initSID.ora initialization parameters file specified.
#
###############################################################################

/u01/app/oracle/product/${ORACLE_RELEASE}/bin/svrmgrl << EOF
spool /home/oracle/startdb.log
connect internal;
startup pfile = /u01/app/oracle/admin/db1/pfile/initTESTDB.ora open;
spool off
EOF

exit 0
```

다음은 데이터베이스 서버를 정지시킬 때 사용되는 stopdb 스크립트의 예제입니다:

```
#!/bin/sh
#
#
# Script to STOP the Oracle Database Server instance.
#
###############################################################################
#
# ORACLE_RELEASE
#
# Specifies the Oracle product release.
#
###############################################################################

ORACLE_RELEASE=8.1.6

###############################################################################
#
# ORACLE_SID
#
# Specifies the Oracle system identifier or "sid", which is the name
# of the Oracle Server instance.
#
###############################################################################
```

```
export ORACLE_SID=TESTDB

#############################################################################
#
# ORACLE_BASE
#
# Specifies the directory at the top of the Oracle software product
# and administrative file structure.
#
#############################################################################

export ORACLE_BASE=/u01/app/oracle

#############################################################################
#
# ORACLE_HOME
#
# Specifies the directory containing the software for a given release.
# The Oracle recommended value is $ORACLE_BASE/product/<release>
#
#############################################################################

export ORACLE_HOME=/u01/app/oracle/product/${ORACLE_RELEASE}

#############################################################################
#
# LD_LIBRARY_PATH
#
# Required when using Oracle products that use shared libraries.
#
#############################################################################

export LD_LIBRARY_PATH=/u01/app/oracle/product/${ORACLE_RELEASE}/lib

#############################################################################
#
# PATH
#
# Verify that the users search path includes $ORCLE_HOME/bin
#
#############################################################################

export PATH=$PATH:/u01/app/oracle/product/${ORACLE_RELEASE}/bin

#############################################################################
#
# This does the actual work.
#
# The oracle server manager is used to STOP the Oracle Server instance
# in a tidy fashion.
#
#############################################################################

/u01/app/oracle/product/${ORACLE_RELEASE}/bin/svrmgrl << EOF
spool /home/oracle/stopdb.log
connect internal;
shutdown abort;
spool off
EOF

exit 0
```

다음은 네트워크 DBI 프록시 데몬을 시작할때 사용되는 startdbi 스크립트의 예제입니다:

```
#!/bin/sh
#
#
#############################################################################
#
# This script allows are Web Server application (perl scripts) to
# work in a distributed environment. The technology we use is
# base upon the DBD::Oracle/DBI CPAN perl modules.
#
# This script STARTS the networking DBI Proxy daemon.
#
#############################################################################

export ORACLE_RELEASE=8.1.6
export ORACLE_SID=TESTDB
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/${ORACLE_RELEASE}
export LD_LIBRARY_PATH=/u01/app/oracle/product/${ORACLE_RELEASE}/lib
export PATH=$PATH:/u01/app/oracle/product/${ORACLE_RELEASE}/bin

#
# This line does the real work.
#

/usr/bin/dbiproxy --logfile /home/oracle/dbiproxy.log --localport 1100 &

exit 0
```

다음은 네트워크 DBI 프록시 데몬을 정지시킬 때 사용되는 stopdbi 스크립트의 예제입니다:

```
#!/bin/sh
#
#
#####################################################################
#
# Our Web Server application (perl scripts) work in a distributed
# environment. The technology we use is base upon the
# DBD::Oracle/DBI CPAN perl modules.
#
# This script STOPS the required networking DBI Proxy daemon.
#
#####################################################################


PIDS=$(ps ax | grep /usr/bin/dbiproxy | awk '{print $1}')

for pid in $PIDS
do
    kill -9 $pid
done

exit 0
```

다음에서는 cluadmin를 사용해서 오라클 서비스를 더하는 방법을 설명하고 있습니다.

```
cluadmin> service add oracle

 The user interface will prompt you for information about the service.
 Not all information is required for all services.
```

```
 Enter a question mark (?) at a prompt to obtain help.

 Enter a colon (:) and a single-character command at a prompt to do
 one of the following:

 c - Cancel and return to the top-level cluadmin command
 r - Restart to the initial prompt while keeping previous responses
 p - Proceed with the next prompt

Preferred member [None]: ministor0
Relocate when the preferred member joins the cluster (yes/no/?) \
 [no]: yes
User script (e.g., /usr/foo/script or None) \
 [None]: /home/oracle/oracle

Do you want to add an IP address to the service (yes/no/?): yes

    IP Address Information

IP address: 10.1.16.132
Netmask (e.g. 255.255.255.0 or None) [None]: 255.255.255.0
Broadcast (e.g. X.Y.Z.255 or None) [None]: 10.1.16.255

Do you want to (a)dd, (m)odify, (d)elete or (s)how an IP address,
 or are you (f)inished adding IP addresses: f

Do you want to add a disk device to the service (yes/no/?): yes

    Disk Device Information

Device special file (e.g., /dev/sda1): /dev/sda1
Filesystem type (e.g., ext2, reiserfs, ext3 or None): ext2
Mount point (e.g., /usr/mnt/service1 or None) [None]: /u01
Mount options (e.g., rw, nosuid): [Return]
Forced unmount support (yes/no/?) [no]: yes

Do you want to (a)dd, (m)odify, (d)elete or (s)how devices,
 or are you (f)inished adding device information: a

Device special file (e.g., /dev/sda1): /dev/sda2
Filesystem type (e.g., ext2, reiserfs, ext3 or None): ext2
Mount point (e.g., /usr/mnt/service1 or None) [None]: /u02
Mount options (e.g., rw, nosuid): [Return]
Forced unmount support (yes/no/?) [no]: yes


Do you want to (a)dd, (m)odify, (d)elete or (s)how devices,
 or are you (f)inished adding devices: f

Disable service (yes/no/?) [no]: no

name: oracle
disabled: no
preferred node: ministor0
relocate: yes
user script: /home/oracle/oracle
IP address 0: 10.1.16.132
 netmask 0: 255.255.255.0
 broadcast 0: 10.1.16.255
device 0: /dev/sda1
 mount point, device 0: /u01
```

```
 mount fstype, device 0: ext2
 force unmount, device 0: yes
device 1: /dev/sda2
 mount point, device 1: /u02
 mount fstype, device 1: ext2
 force unmount, device 1: yes

Add oracle service as shown? (yes/no/?) y
notice: Starting service oracle ...
info: Starting IP address 10.1.16.132
info: Sending Gratuitous arp for 10.1.16.132 (00:90:27:EB:56:B8)
notice: Running user script '/home/oracle/oracle start'
notice, Server starting
Added oracle.
cluadmin>
```

5.2. 오라클 서비스 사용자화 하기

오라클 데이터베이스가 오류에서 복구하는데 소유되는 시간은 멈춰진 트렌스액션의 수와 데이터베이스의 크기에 정비례합니다. 다음의 매개 변수는 데이터베이스 오류 복구 시간을 조정합니다:

- **LOG_CHECKPOINT_TIMEOUT**
- **LOG_CHECKPOINT_INTERVAL**
- **FAST_START_IO_TARGET**
- **REDO_LOG_FILE_SIZES**

복구 시간을 최소화 시키려면, 전과 같은 변수를 비교적 적은 숫자로 설정하시면 됩니다. 주의하실 것은 너무 적은 숫자를 사용하게 되면, 성능에 직접적으로 영향을 끼칠 수 있습니다. 알맞는 값을 찾기위해 몇가지의 값을 사용해 보는 것이 좋습니다.

오라클은 또 데이터베이스 트렌스액션 재실행 또는 재실행 시간을 설정하는 다른 사용자화 변수들을 제공 합니다. 클러스터 환경이 충분한 오류 복구 시간을 가질 수 있도록 적당한 숫자를 사용해 주십시요. 이러한 과정을 통해 데이터베이스 클라이언트 프로그램 사용에 절대 지장이 없을 것이며, 프로그램들이 재접속해야 할 필요가 없습니다.

5.3. MySQL 서비스 설정하기

데이터베이스 서비스는 MySQL 데이터베이스 응용 프로그램에 고성능 데이터를 제공할 수 있습니다. 고성능 데이터를 제공받은 응용 프로그램은 웹 서버와 같은 데이터베이스 클라이언트 시스템이 네트워크에 접속할 수 있도록 해줍니다. 만일 서비스가 오류 복구에 들어가면, 프로그램은 새로운 클러스트 시스템을 통하여 공유 데이터베이스 자료를 사용할 것입니다. 네트워크를 사용하는 데이터베이스 서비스에는 일반적으로 IP 주소가 주어지며, 이것은 서비스와 함께 오류 복구시 새로운 시스템으로 옮겨져, 클라이언트의 사용에 전혀 지장이 없도록 합니다.

다음은 MySQL 데이터베이스 서비스의 예제입니다:

- MySQL 서버와 데이터베이스 모두 공유 저장 디스크에 있는 디스크 파티션에 저장 됩니다. 이것을 통해 오류 복구시 필요한 데이터베이스 데이터와 이의 실행 환경이 양쪽 클러스터 시스템에 의해 사용될 수 있습니다. 예를 들어, 파일 시스템은 공유 저장 파티션인 `/dev/sda1`를 사용하여 `/var/mysql`로 마운트 됩니다.
- MySQL 데이터베이스에, 네트워크를 사용한 클라이언트들이 데이터베이스를 사용할 수 있도록 IP 주소가 주어집니다. 이 IP 주소는 오류 복구시 자동적으로 클러스터 구성원 사이에서 사용됩니다. 아래의 예제에서 IP 주소는 10.1.16.12입니다.

- MySQL 데이터베이스를 시작하고 멈추는데 사용되는 스크립트는 표준 System V `init` 스크립트이며, 설치된 데이터베이스에 따라 알맞는 설정 변수가 넣어질 수 있습니다.
- 기본적으로, MySQL 서버에 접속한 클라이언트는 8시간 동안 사용이 없을 시, 자동으로 끊기게 되어있습니다. 이 한계는 `mysqld`를 시작할때, `wait_timeout` 변수를 정해줌으로써 바꿀 수 있습니다. 예를 들어, 이 timeout을 4시간으로 바꾸려면, 다음과 같이 MySQL 데몬을 시작하면 됩니다:
  `mysqld -O wait_timeout=14400`

  MySQL 서버가 timeout이 되었는지를 확인하려면, `mysqladmin version`를 실행한 후, uptime을 확인해 보시면 됩니다. 다시 서버에 재접속하기 위해 다시 질의합니다.

  리눅스 배포판에 따라 다르지만, 다음 중 하나는 MySQL 서버의 timeout을 나타냅니다:
  `CR_SERVER_GONE_ERROR`
  `CR_SERVER_LOST`

`/usr/share/cluster/doc/services/examples/mysql.server`에 MySQL 데이터베이스를 시작하고 정지시킬 스크립트가 있으며, 다음은 예제입니다:

```
#!/bin/sh# Copyright Abandoned 1996 TCX DataKonsult AB & Monty Program KB & Detron HB# This file is public domain and comes with NO WARRANTY of any kind# Mysql daemor
```

다음은 `cluadmin`를 사용해 MySQL 서비스를 더하는 방법입니다.

```
cluadmin> service add

 The user interface will prompt you for information about the service.
 Not all information is required for all services.

 Enter a question mark (?) at a prompt to obtain help.

 Enter a colon (:) and a single-character command at a prompt to do
 one of the following:

 c - Cancel and return to the top-level cluadmin command
 r - Restart to the initial prompt while keeping previous responses
 p - Proceed with the next prompt

Currently defined services:

 databse1
 apache2
 dbase_home
 mp3_failover

Service name: mysql_1
Preferred member [None]: devel0
Relocate when the preferred member joins the cluster (yes/no/?) [no]: yes
User script (e.g., /usr/foo/script or None) [None]: \
 /etc/rc.d/init.d/mysql.server

Do you want to add an IP address to the service (yes/no/?): yes

    IP Address Information

IP address: 10.1.16.12
Netmask (e.g. 255.255.255.0 or None) [None]: [Return]
Broadcast (e.g. X.Y.Z.255 or None) [None]: [Return]

Do you want to (a)dd, (m)odify, (d)elete or (s)how an IP address,
 or are you (f)inished adding IP addresses: f

Do you want to add a disk device to the service (yes/no/?): yes
```

```
    Disk Device Information

Device special file (e.g., /dev/sda1): /dev/sda1
Filesystem type (e.g., ext2, reiserfs, ext3 or None): ext2
Mount point (e.g., /usr/mnt/service1 or None) [None]: /var/mysql
Mount options (e.g., rw, nosuid): rw
Forced unmount support (yes/no/?) [no]: yes

Do you want to (a)dd, (m)odify, (d)elete or (s)how devices,
 or are you (f)inished adding device information: f

Disable service (yes/no/?) [no]: yes

name: mysql_1
disabled: yes
preferred node: devel0
relocate: yes
user script: /etc/rc.d/init.d/mysql.server
IP address 0: 10.1.16.12
 netmask 0: None
 broadcast 0: None
device 0: /dev/sda1
 mount point, device 0: /var/mysql
 mount fstype, device 0: ext2
 mount options, device 0: rw
 force unmount, device 0: yes

Add mysql_1 service as shown? (yes/no/?) y
Added mysql_1.
cluadmin>
```

5.4. DB2 서비스 설정하기

이 곳에서는 IBM DB2 엔터프라이즈/워크그룹 프로그램을 클러스터에 서비스로 설정하여 오류 복구하는 예제를 보여주고 있습니다. 이 예제에서는 NIS가 실행되지 않고 있다고 가정합니다. 프로그램을 설치하려면, 다음과 같은 단계를 따라 주십시요:

1. 양쪽 클러스터 시스템에, 루트로 로그인하신 후, IP 주소와 DB2가 사용할 호스트 이름을 /etc/hosts에 다음과 같이 입력해 줍니다:
   ```
   10.1.16.182   ibmdb2.class.cluster.com    ibmdb2
   ```
2. 공유 저장 디스크에 사용되지 않은 파티션을 DB2 관리 및 인스턴스 데이타를 저장하기 위해 선택하고 그곳에 파일 시스템을 생성합니다. 예를 들어:
   ```
   # mke2fs /dev/sda3
   ```
3. 두번째 단계로서, 양쪽 클러스터 시스템 상에 생성하신 파일 시스템에 대한 마운트 지점을 생성합니다. 예를 들어:
   ```
   # mkdir /db2home
   ```
4. 첫 클러스터 시스템인, devel0에 두번째 단계에서 생성하신 파일 시스템을 세번째 단계에서 만든 마운트 지점으로 마운트합니다. 예를 들면:
   ```
   devel0# mount -t ext2 /dev/sda3 /db2home
   ```
5. 첫 클러스터 시스템, devel0에, DB2 cdrom을 마운트하신 후 배포판에 포함된 설정 응답 파일들을 /root로 복사해 옵니다. 예를 들어:
   ```
   devel0% mount -t iso9660 /dev/cdrom /mnt/cdrom
   devel0% cp /mnt/cdrom/IBM/DB2/db2server.rsp /root
   ```
6. 설정 응답 파일인 db2server.rsp를 현 시스템에 알맞게 편집합니다. 기억하실 것은 UID와 GID가 양 클러스터 시스템에 공존하도록 해야 합니다. 예를 들어:

```
-----------Instance Creation Settings------------
-------------------------------------------------
DB2.UID = 2001
DB2.GID = 2001
DB2.HOME_DIRECTORY = /db2home/db2inst1

-----------Fenced User Creation Settings----------
------------------------------------------------
UDF.UID = 2000
UDF.GID = 2000
UDF.HOME_DIRECTORY = /db2home/db2fenc1

-----------Instance Profile Registry Settings------
-----------------------------------------------
DB2.DB2COMM = TCPIP

----------Administration Server Creation Settings---
-----------------------------------------------
ADMIN.UID = 2002
ADMIN.GID = 2002
ADMIN.HOME_DIRECTORY = /db2home/db2as

---------Administration Server Profile Registry Settings-
-------------------------------------------------------
ADMIN.DB2COMM = TCPIP

---------Global Profile Registry Settings-------------
-----------------------------------------------
DB2SYSTEM = ibmdb2
```

7. 다음과 같이 설치를 시작합니다:
```
devel0# cd /mnt/cdrom/IBM/DB2
devel0# ./db2setup -d -r /root/db2server.rsp 1>/dev/null \
  2>/dev/null &
```
8. 설치 로그 파일인, `/tmp/db2setup.log`을 사용해서 설치 상에 아무 문제가 없었는지 확인합니다. 로그 파일의 마지막에 모든 단계에 **SUCCESS** 라고 적혀 있어야 합니다.
9. 첫 클러스터 상의 DB2 인스턴스와 관리자 서버를 정지 합니다. 예를 들어:
```
devel0# su - db2inst1
devel0# db2stop
devel0# exit
devel0# su - db2as
devel0# db2admin stop
devel0# exit
```
10. 다음과 같이 첫 클러스터 시스템으로부터 DB2와 관리 데이터 파티션을 마운트 해제하십시오:
```
devel0# umount /db2home
```
11. 두번째 클러스터 시스템인 `devel1`에 DB2 인스턴스와 관리자 데이터 파티션을 마운트시킵니다. 예를 들어:
```
devel1# mount -t ext2 /dev/sda3 /db2home
```
12. DB2 CDROM을 두번째 클러스터 시스템에 장착한 후, 원격으로 `db2server.rsp` 파일을 `/root`로 복사합니다. 예를 들어:
```
devel1# mount -t iso9660 /dev/cdrom /mnt/cdrom
devel1# rcp devel0:/root/db2server.rsp /root
```
13. 두번째 클러스터 시스템, `devel1`에 설치를 시작합니다. 예를 들어:
```
devel1# cd /mnt/cdrom/IBM/DB2
devel1# ./db2setup -d -r /root/db2server.rsp 1>/dev/null \
  2>/dev/null &
```

14. 로그 파일을 이용해 설치 상에 문제가 없었는지 확인 합니다. 설치 중 모든 단계들이 다음을 제외하고는 **SUCCESS**로 나와야 합니다:
```
DB2 Instance Creation              FAILURE
Update DBM configuration file for TCP/IP     CANCEL
Update parameter DB2COMM             CANCEL
Auto start DB2 Instance             CANCEL
DB2 Sample Database               CANCEL
Start DB2 Instance
Administration Server Creation          FAILURE
Update parameter DB2COMM            CANCEL
Start Administration Serve           CANCEL
```
15. 우선 첫 시스템 상에서 다음의 명령어를 사용하여 데이터베이스가 제대로 설치되었는지 확인하신 후 다음으로 두번째 시스템을 확인하시기 바랍니다:
```
# mount -t ext2 /dev/sda3 /db2home
# su - db2inst1
# db2start
# db2 connect to sample
# db2 select tabname from syscat.tables
# db2 connect reset
# db2stop
# exit
# umount /db2home
```
16. DB2 관리와 인스턴스 데이터 파티션에서 DB2 클러스터를 시작/정지시킬 스크립트를 만듭니다. 예를 들면:
```
# vi /db2home/ibmdb2
# chmod u+x /db2home/ibmdb2

#!/bin/sh
#
# IBM DB2 Database Cluster Start/Stop Script
#

DB2DIR=/usr/IBMdb2/V6.1

case $1 in
"start")
 $DB2DIR/instance/db2istrt
 ;;
"stop")
 $DB2DIR/instance/db2ishut
 ;;
esac
```
17. 양쪽 시스템의 `/usr/IBMdb2/V6.1/instance/db2ishut` 파일을 편집하여 데이터베이스가 멈추기 전 현재 접속되어 있는 프로그램들을 강제적으로 접속 해제하게 만듭니다. 예를 들어:
```
for DB2INST in ${DB2INSTLIST?}; do  echo "Stopping DB2 Instance "${DB2INST?}"..." >> ${LOGFILE?}  find_homedir ${DB2INST?}  INSTHOME="${USERHOME?}"  s
```
18. 클러스터 서비스가 DB2 서비스를 시작하고 정지할 수 있도록 `inittab` 파일을 편집하여 DB2 관련 줄을 주석화 시킵니다. 일반적으로 DB2 라인은 파일 마지막 줄에 위치합니다. 예:
```
# db:234:once:/etc/rc.db2 > /dev/console 2>&1 # Autostart DB2 Services
```

`cluadmin` 유틸리티를 사용해서 DB2 서비스를 만듭니다. 첫 단계에서의 IP 주소와 두번째 단계에서 만들어진 공유 파티션, 그리고 16번째 단계에서 생성한 시작/정지 스크립트를 넣어줍니다.

세번째 시스템에 DB2 클라이언트를 설치하시려면 다음과 같은 명령어를 사용합니다:

```
display# mount -t iso9660 /dev/cdrom /mnt/cdrom
display# cd /mnt/cdrom/IBM/DB2
display# ./db2setup -d -r /root/db2client.rsp
```

DB2 클라이언트를 사용하려면, 서비스의 IP 주소를 클라이언트 시스템의 `/etc/hosts` 파일에 더합니다:

```
10.1.16.182 ibmdb2.lowell.mclinux.com ibmdb2
```

그 후, 다음과 같은 항목을 클라이언트 시스템의 /etc/services에 추가하시기 바랍니다:

```
db2cdb2inst1  50000/tcp
```

클라이언트 시스템에 다음과 같은 명령어를 실행합니다:

```
# su - db2inst1
# db2 catalog tcpip node ibmdb2 remote ibmdb2 server db2cdb2inst1
# db2 catalog database sample as db2 at node ibmdb2
# db2 list node directory
# db2 list database directory
```

DB2 클라이언트 시스템에서 테스트하시려면 다음과 같은 명령어를 사용합니다:

```
# db2 connect to db2 user db2inst1 using ibmdb2
# db2 select tabname from syscat.tables
# db2 connect reset
```

6장 .

네트워크 파일 공유 서비스

이 장에서는 Red Hat Linux 어드밴스 서버가 고가용성 NFS와 삼바를 통해 네트워크 파일 공유 서비스 할 수 있도록 설정하는 방법에 대해 설명하고 있습니다.

6.1. NFS 서비스 설정하기

고가용성 네트워크 파일 시스템(NFS)은 클러스터의 큰 장점 중에 하나입니다. 클러스터된 NFS 서비스의 장점들은 다음과 같습니다:

- NFS 클라이언트에게 서버의 오류 중에도 데이터의 방해받지 않도록 제공해 줍니다.
- 계획된 서버의 관리, 즉 업데이트나 오류 복구등의 관리를 클라이언트들에게는 아무런 차이가 없이 다른 클러스터 구성원으로 NFS 서비스를 재배치시켜, 서버 관리를 쉽게 만들어 줍니다.
- Active-Active 설정을 사용해, 현재 있는 장비를 최대한 활용할 수 있습니다. Active-active에 대한 자세한 내용은 이 장 뒷 부분에 나옵니다.

6.1.1. NFS 서버의 요구 사항

고가용성 NFS 서비스를 만들려면, 각 클러스터가 가져야할 몇가지 요구 사항이 있습니다. (주의: 이 요구 사항은 NFS 클라이언트 시스템에 적용되지 않습니다.) 요구 사항들은 다음과 같습니다:

- 커널상의 NFS 서버 지원이 활성화 되어야 합니다. NFS는 모듈로나 항시 사용하도록 설정할 수 있습니다. NFS V2와 NFS V3모두 지원됩니다.
- Red Hat Linux 어드밴스 서버 2.1의 커널에는 (Mission Critical Linux Inc. 에서 개발한) 투명한 NFS 서버의 재배치를 부분이 강화되어 있습니다. 이 커널의 강화된 부분은, NFS 클라이언트가 서버의 재배치 중에 파일 핸들링 에러를 받지 않도록 되어있습니다. 만일 커널 소스에 이와 같은 것을 가지고 있지 않아도, NFS는 클러스터 중에 실행되도록 설정될 수 있습니다. 하지만 서비스가 시작하고 멈출때, 이와 같은 커널 관련 주의 문구가 나타날 것입니다.
- NFS 데몬은 모든 클러스터 서버에서 실행되어야 합니다. 이것은 NFS의 init.d 런 레벨 스크립트에서 자동으로 이루어 집니다. 예를 들어:
  `/sbin/chkconfig --level 345 nfs on`
- RPC `portmap` 포트 맵 데몬이 활성화 되어 있어야 합니다. 예를 들어:
  `/sbin/chkconfig --level 345 portmap on`

  NFS 서비스는 다음과 같은 NFS 데몬들이 실행 되지 않으면 사용할 수 없습니다: `nfsd`, `rpc.mountd`, 그리고 `rpc.statd`.
- 클러스터 된 NFS 서비스의 파일 시스템 마운트와 관련된 exports는 `/etc/fstab` 혹은 `/etc/exports`에 명시되면 안됩니다. 클러스터된 NFS 서비스는, 대신 `cluadmin`설정중에 마운트와 exports 관련 변수에 넣어질 수 있습니다.

6.1.2. NFS 서비스 설정 관련 변수 모으기

NFS 서비스 설정을 준비하면서, 파일 시스템들이 어떻게 export 될것이며 오류 복구될 것인지 계획하는 것이 중요합니다. 다음은 NFS 서비스를 설정하는데 필요한 자료들입니다:

- 서비스 이름 — 클러스터 상에서 사용될 특유의 이름.

- 우선 구성원 — 만일 하나 이상의 클러스터 구성원이 작동중일때, NFS 서버가될 구성원을 나타냅니다.
- 재배치 방침 — 서비스가 시작되었을 때, 우선 구성원이 작동중이 아니었으나, 나중에 우선 구성원이 연결되었을 때, 재배치를 하는지 않는지의 여부를 나타냅니다. 이 번수는 각 클러스터 구성원을 NFS 서버로 인식 부하를 반씩 줄이기에, 부하분산에 있어 중요한 역할을 합니다.
- *IP* 주소 — NFS 클라이언트는 파일시스템을 NFS 서버를 통해 이용하는데, 서버는 IP 주소(혹은 관련된 호스트이름)으로 이용 가능 합니다. NFS 클라이언트가 서버가 어디인지 알기 위해, 현재 서버가 실행되고 있는 클러스터 구성원의 호스트 이름을 사용해서는 안됩니다. 클러스터된 NFS서비스에는 클러스터 서버들과 다른 유동 IP 주소를 주는 것이 좋습니다. 이 유동 IP 주소는 어느 서버에서 NFS가 export 되고 있던지 그곳에 설정이 되면 됩니다. 이 방법을 따르면, NFS 클라이언트는 이 유동 IP 주소만 알면 되고, 클러스터된 NFS서버가 있는 줄은 알지 못할 것입니다. NFS 서비스의 IP 주소를 입력할때는, 관리자가 넷마스크와 브로드캐스트 주소를 입력해 달라고 할 것입니다. 만일 **None** (기본임)이 선택되면, 현재 네트워크 인터페이스에 설정된 넷마스크와 브로드캐스트와 같을 것입니다.
- 마운트 관련 자료 — 클러스터 되지 않은 파일 시스템에서는 보통 마운트 관련 자료는 `/etc/fstab`에 입력이 되었습니다. 비교해 보면, 클러스터된 파일 시스템의 마운트 지점은 `/etc/fstab`에 입력되면 안됩니다. 이것은 오직 한 클러스터 구성원만 한번에 파일 시스템을 마운트하는 것을 돕습니다. 만일 이렇게 되지 못하면, 파일 시스템 문제와 시스템 다운을 유발할 수 있습니다.
  - 장치 관련 특벌 파일 — 마운트 정보에는 디스크의 장치 특벌 파일과 파일 시스템이 마운트될 곳이 나타납니다. NFS 서비스를 설정하는 중, 이 정보를 입력해야 합니다.
  - 마운트 지점 디렉토리 — NFS 서비스는 하나 이상의 파일 시스템을 마운트할 수 있습니다. 이에 맞게 파일 시스템들은 하나의 오류 복구로 설정될 것입니다.
  - 마운트 옵션 — 마운트 관련 정보는 마운트 옵션을 포함합니다. 주의: 기본으로, 리눅스 NFS 서버는 디스크에 쓰는것이 동시에 되지 않습니다. 이것을 돕기위해 `sync` 옵션을 사용해야 합니다. `sync` 옵션을 사용하면, 성능에는 문제가 있으나, 데이타 무결성은 좀더 강화됩니다. 마운트에 관련된 옵션에 대한 자세한 정보는 mount(8)에서 보실 수 있습니다.
  - 강제적인 마운트 해제 — 마운트 관련 정보의 일부로, 강제적인 마운트 해제를 할것인지 아닌지를 정해야 합니다. 만일 강제적인 마운트 해제를 활성화 시키면, 만일 클라이언트 서버가 마운트된 파일 시스템을 가지고 있을때, 서비스가 비활성화되거나, 재배치 될 때, 마운트 해제가 가능하도록 사용중인 모든 프로그램을 죽일것 입니다.

- *Export* 정보 — 클러스터되지 않은 NFS 서비스에서는, export 정보가 일반적으로 `/etc/exports` 파일에 입력됩니다. 하지만, 클러스터된 NFS 서비스에서는 export 관련 정보를 `/etc/exports`에 넣으면 안됩니다., 하지만, 서비스 설정 중에 이 정보를 입력할 수 있는 곳이 있습니다. Export 정보는 다음과 같습니다:
  - *Export* 디렉토리 — export 디렉토리는 마운트 지점과 같이 마운트 정보가 지정 됩니다. 이 경우, 모든 파일 시스템이 NFS를 통해 이용 가능합니다. 또 다른 방법으로 마운트된 파일 시스템의 일부만 마운트 될 수도 있습니다. 한 마운트 지점의 subdirectory를 export함으로써, NFS클라이언트에 다른 이용 허가를 줄 수 있습니다.
  - *Export* 클라이언트 이름 — 이 번수는 어떤 시스템이 NFS 클라이언트로 이 파일 시스템을 사용할 수 있는지 지정합니다. 이 방법으로는, 개인 시스템이(e.g. fred) 지정될 수도, 혹은 와일드 카드로 한 그룹의 시스템이(e.g. *.wizzbang.com) 지정될 수도 있습니다. *를 클라이언트 이름 대신에 넣음으로 모든 클라이언트가 파일 시스템을 마운트할 수 있게 할 수도 있습니다.
  - *Export* 클라이언트 옵션 — 이 번수는 적정 클라이언트들에 알맞는 허가를 줄때 사용 됩니다. 예를 들어, 여기에는 **ro** (read only), 과 **rw** (read write)가 있습니다. 확실하게 Export 옵션이 정해지지 않는한, 기본 export 옵션은 `ro`,`async`,`wdelay`,`root_squash` 입니다.

  좀 더 자세히 export와 그 번수를 알고 싶으면, exports(5)를 참고하시기 바랍니다.

NFS 서비스를 설정하기 위해 `cluadmin`를 사용하시면:

- 매우 주의해서 서비스 변수를 입력하시기 바랍니다. NFS 변수와 관련된 확인 작업이 현재 매우 미진합니다.
- 거의 모든 프롬프트에서, 좀더 자세한 설명이 필요하면, [?]를 입력하면 됩니다.

6.1.3. NFS 서비스 설정 예제

NFS 서비스 설정을 보여주기 위해서 예제 설정이 이곳에 설명되어 있습니다. 이 예제는 간단하게 4명의 총무부의 home 디렉토리가 있는 하나의 디렉토리를 export 하는 설정을 하고 있습니다. NFS 클라이언트는 이 4명의 시스템으로 제한되어 있습니다.

다음 서비스 설정 변수는 다른 자세히 설명된 것들과 더불어 사용될 것입니다.


주의

cluadmin를 사용해서 NFS 서비스를 설정하기 전에, 클러스터 데몬들이 실행 중이어야 합니다.

- 서비스 이름 — **nfs_accounting**. 이 이름은 서비스의 목적인 총무부의 구성원에게만 export 되는 것을 보이고 있습니다.
- 우선 멤버 — **clu4**. 이 예제 클러스터에서는 구성원의 이름이 clu3와 clu4입니다.
- 사용자 스크립트 — 클러스터에 NFS 서비스가 지원 됩니다. 그렇기 때문에 따로 사용자 스크립트를 만들 필요가 없습니다. 이 이유때문에, 사용자 스크립트를 입력해 달라고 할때 기본 값인 **None**이 선택 될 것입니다.
- *IP* 주소 — **10.0.0.10**. 이 주소에는 clunfsacct라는 호스트 이름이 지정되어 있고, 이것을 통해 NFS 클라이언트가 파일시스템을 마운트할 것입니다. 주의 할 것은 이 주소는 두 클러스터 멤버 (clu3와 clu4)와 다릅니다. 기본 넷마스크와 브로드캐스트 주소가 사용될 것입니다.
- 마운트 정보 — /dev/sdb10, 이것은 파일 시스템이 있을 공유 저장 RAID 장치를 가르치고 있습니다. **ext3** —는 파일 시스템이 만들어 졌을 때 시스템의 유형을 지정합니다. /mnt/users/accounting — 는 파일 시스템의 마운트 지점을 애기합니다. rw,nosuid,sync —가 마운트 옵션입니다.
- *Export* 정보 - 이번 예제는, 전체 파일 시스템이 읽고 쓰기를 기본으로 총무부에 있는 네 사람에게만 이용가능할 것입니다. 이 네명이 사용하고 있는 시스템의 이름은, burke, stevens, needle 그리고 dwalsh입니다.

다음이 클러스터 상에서 사용될 IP주소와 호스트 주소입니다:

```
10.0.0.3    clu3     # cluster member

10.0.0.4    clu4     # second cluster member

10.0.0.10   clunfsacct  # floating IP address associated with accounting dept. NFS service

10.0.0.11   clunfseng  # floating IP address associated with engineering dept. NFS service
```

다음은 이번 예제를 가지고 cluadmin을 실행할 경우입니다:

```
cluadmin> service add

Service name: nfs_accounting
Preferred member [None]: clu4
Relocate when the preferred member joins the cluster (yes/no/?) \
  [no]: yes
Status check interval [0]: 30
User script (e.g., /usr/foo/script or None) [None]:
Do you want to add an IP address to the service (yes/no/?) [no]: yes

    IP Address Information

IP address: 10.0.0.10
```

```
Netmask (e.g. 255.255.255.0 or None) [None]:
Broadcast (e.g. X.Y.Z.255 or None) [None]:
Do you want to (a)dd, (m)odify, (d)elete or (s)how an IP address, or
are you (f)inished adding IP addresses [f]: f
Do you want to add a disk device to the service (yes/no/?) [no]: yes

Disk Device Information

Device special file (e.g., /dev/sdb4): /dev/sdb10
Filesystem type (e.g., ext2, ext3 or None): ext3
Mount point (e.g., /usr/mnt/service1) [None]: /mnt/users/accounting
Mount options (e.g., rw,nosuid,sync): rw,nosuid,sync
Forced unmount support (yes/no/?) [yes]:
Would you like to allow NFS access to this filesystem (yes/no/?) [no]: yes

You will now be prompted for the NFS export configuration:

Export directory name: /mnt/users/accounting

Authorized NFS clients

Export client name [*]: burke
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]: a

Export client name [*]: stevens
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]: a

Export client name [*]: needle
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]: a

Export client name [*]: dwalsh
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]: f
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS EXPORTS, or
are you (f)inished adding EXPORTS [f]:
Do you want to (a)dd, (m)odify, (d)elete or (s)how DEVICES,
or are you (f)inished adding DEVICES [f]:
Disable service (yes/no/?) [no]:
name: nfs_eng
disabled: no
preferred node: clu4
relocate: yes
user script: None
monitor interval: 30
IP address 0: 10.0.0.10
 netmask 0: None
 broadcast 0: None
device 0: /dev/sdb10
 mount point, device 0: /mnt/users/accounting
 mount fstype, device 0: ext3
 mount options, device 0: rw,nosuid,sync
 force unmount, device 0: yes
NFS export 0: /mnt/users/accounting
 Client 0: burke, rw
 Client 1: stevens, rw
```

```
 Client 2: needle, rw
 Client 3: dwalsh, rw
Add nfs_eng service as shown? (yes/no/?) yes
Added nfs_eng.
cluadmin>
```

6.1.4. NFS 클라이언트 이용

클라이언트를 위한 NFS 사용 모델은 보통때와 바뀐 것이 하나도 없습니다. 전의 예제를 따른다면, 만일 클라이언트 시스템이 고성능 NFS 서비스를 마운트하고 싶으면, `/etc/fstab` 파일에 다음과 같은 항이 있으면 됩니다:

```
clunfsacct:/mnt/users/accounting /mnt/users/ nfs bg 0 0
```

6.1.5. Active-Active NFS 설정

앞의 부분에서, 기본적인 NFS 서비스의 예제 설정을 보셨습니다. 여기서는 좀더 복잡한 NFS 서비스를 설정하는지에 대해서 얘기하고 있습니다.

이곳에는 두 개의 고성능 NFS서비스를 설정하는 방법이 나옵니다. 이 예제에서는, 클러스터에서 제공되고 있는 NFS 파일 시스템을 다른 두 팀의 사용자들이 이용하려고 할것입니다. 이 사용자들을 위해 두개의 다른 NFS 서비스가 설정될 것입니다. 각 서비스는 다른 IP 주소와 우선 클러스터 멤버를 가지게 될것입니다. 이것을 통해, 보통 작동중에, 두 클러스터 구성원이 작동중일때, 각각 다른 파일 시스템을 export하게 됩니다. 이것은 관리자에게 현재 가지고 있는 두 서버의 능력을 최대한 사용할 수 있게 해줄 것입니다. 둘 중 하나의 클러스터 멤버에 오류 발생시 (혹은 보통 관리중), 작동중인 클러스터 구성원에서 전 NFS 서비스가 실행될 것입니다.

이 예제 설정은 전에 만들어진 NFS서비스에 두번째 서비스가 덧붙여질 것입니다. 다음 서비스 설정 변수는 두번째 서비스에 관련된 것입니다:

- 서비스 이름 — **nfs_engineering**. 이 이름은 파일 시스템이 엔지니어링 부서에 있는 구성원에게 NFS export가 제공 될것을 나타내고 있습니다.
- 우선 구성원 — **clu3**. 이 예제 클러스터에서, 구성원의 이름이 clu3와 clu4입니다. 주의하실 것은 clu3가 지정되었는데 그것은 다른 서비스 (nfs_accounting)에 clu4가 지정되었기 때문입니다.
- *IP* 주소 — **10.0.0.11**. 이 주소에는 clunfseng라는 호스트 이름이 주어져 있습니다, 이 이름을 통해 NFS 클라이언트들이 파일 시스템을 마운트할 것입니다. 주의하실 것은 이 IP 주소가 두 클러스터 구성원 (clu3와 clu4)와 전혀 다르다는 것입니다. 또한 이 IP 주소가 다른 NFS 서비스 (nfs_accounting)과 또 다르 다는 것을 기억하셔야 합니다. 기본 넷마스크와 브로드캐스트가 사용될 것입니다.
- 마운트 정보 — `/dev/sdb11`, 이것은 파일 시스템이 있을 공유 저장 RAID를 나타냅니다. **ext2** — 이것은 파일 시스템이 생성되었을때 사용될 유형을 나타냅니다. `/mnt/users/engineering` — 은 파일 시스템의 마운트 지점을 나타내며, `rw,nosuid,sync` — 은 마운트 옵션입니다.
- *Export* 정보 — 이 예제를 위해서, 각 하부 디렉토리의 파일 시스템은 3명의 엔지니어링 팀 구성원에 기본해서 읽고-쓰기 (rw)로 이용될 것입니다. 팀 구성원이 사용하는 시스템의 이름은, `ferris`, `denham`, 그리고 `brown`입니다. 이 예제가 좀 더 확실한 예제가 되려면, 각 팀 구성원은 자신의 디렉토리만 이용가능할 것입니다.

다음은 `cluadmin`를 두번째 NFS 서비스 생성에 사용했을때의 결과입니다.

```
cluadmin> service add

Service name: nfs_engineering
Preferred member [None]: clu3
Relocate when the preferred member joins the cluster (yes/no/?) [no]: yes
Status check interval [0]: 30
```

```
User script (e.g., /usr/foo/script or None) [None]:
Do you want to add an IP address to the service (yes/no/?) [no]: yes

    IP Address Information

IP address: 10.0.0.11
Netmask (e.g. 255.255.255.0 or None) [None]:
Broadcast (e.g. X.Y.Z.255 or None) [None]:
Do you want to (a)dd, (m)odify, (d)elete or (s)how an IP address, or
are you (f)inished adding IP addresses [f]: f
Do you want to add a disk device to the service (yes/no/?) [no]: yes

Disk Device Information

Device special file (e.g., /dev/sdb4): /dev/sdb11
Filesystem type (e.g., ext2, ext3 or None): ext2
Mount point (e.g., /usr/mnt/service1) [None]: /mnt/users/engineering
Mount options (e.g., rw,nosuid,sync): rw,nosuid,sync
Forced unmount support (yes/no/?) [yes]:
Would you like to allow NFS access to this filesystem (yes/no/?) \
  [no]: yes

You will now be prompted for the NFS export configuration:

Export directory name: /mnt/users/engineering/ferris

Authorized NFS clients

Export client name [*]: ferris
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]: f
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS EXPORTS, or
are you (f)inished adding EXPORTS [f]: a

Export directory name: /mnt/users/engineering/denham

Authorized NFS clients

Export client name [*]: denham
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]:
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS EXPORTS, or
are you (f)inished adding EXPORTS [f]: a

Export directory name: /mnt/users/engineering/brown

Authorized NFS clients

Export client name [*]: brown
Export client options [None]: rw
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS CLIENTS, or
are you (f)inished adding CLIENTS [f]: f
Do you want to (a)dd, (m)odify, (d)elete or (s)how NFS EXPORTS, or
are you (f)inished adding EXPORTS [f]: a
Do you want to (a)dd, (m)odify, (d)elete or (s)how DEVICES, or
are you (f)inished adding DEVICES [f]:
Disable service (yes/no/?) [no]:
name: nfs_engineering
disabled: no
preferred node: clu3
```

```
relocate: yes
user script: None
monitor interval: 30
IP address 0: 10.0.0.11
 netmask 0: None
 broadcast 0: None
device 0: /dev/sdb11
 mount point, device 0: /mnt/users/engineering
 mount fstype, device 0: ext2
 mount options, device 0: rw,nosuid,sync
 force unmount, device 0: yes
NFS export 0: /mnt/users/engineering/ferris
 Client 0: ferris, rw
NFS export 0: /mnt/users/engineering/denham
 Client 0: denham, rw
NFS export 0: /mnt/users/engineering/brown
 Client 0: brown, rw
Add nfs_engineering service as shown? (yes/no/?) yes
Added nfs_engineering.
cluadmin>
```

6.1.6. NFS 경고

클러스터된 NFS서비스를 설정시, 다음과 같은 사항을 고려하시기 바랍니다.

`exportfs -r` 사용을 피해야 합니다.

‘ 클러스터 구성원을 통해 export된 NFS 파일 시스템은 기존의 `/etc/exports`과 다릅니다. 클러스터 서비스와 관련된 NFS exports는 클러스터 설정 파일에 설정되어 있습니다. (`cluadmin`에서 설정된 것처럼).

`exportfs -r` 명령어는 `/etc/exports`에 명시되지 않은 모든 export를 제거하게 되어있습니다. 이 명령어를 실행한다면, 서비스가 재시작 될때까지, NFS 서비스가 사용불가능 하게 됩니다. 이 이유에서, `exportfs -r` 의 사용을 자제해야 하는 것입니다. 만일 `exportfs -r`를 사용한후 다시 고가용성 NFS 서버를 사용하고자 한다면, NFS 클러스터 서비스를 멈추고 다시 시작해야 합니다.

NFS 파일 잠금

‘ NFS 파일 lock은 오류 복구나 서비스 재배치때 보존되지 않습니다. 이것은 리눅스 상의 NFS는 파일 locking 정보를 시스템 파일에 저장하기 때문입니다. 이 시스템 파일의 NFS locking 상태는 클러스터 전체적으로 알려지지 않습니다. 이 때문에 오류 복구시 lock이 다시 재발행되게 됩니다.

6.2. 고성능 삼바 서비스 설정하기

고성능 네트워크 파일 서비스는 클러스터의 강점 중 하나입니다. 고성능 삼바의 장점은 다음과 같습니다:

- Microsft® Windows™ 클라이언트 같은 CIFS/SMB 프로토콜을 사용하는 클라이언트에게 다양한 파일을 제공할 수 있습니다.
- 같은 파일 시스템을 동시에 네트워크 상으로 NFS와 Window기반의 클라이언트에게 제공할 수 있습니다.
- 서버의 오류 때, 빠르게 재접속 할수 있게 하여 해주며, 윈도우--기반의 클라이언트 들에게 중요한 데이타를 사용할 수 있게 해줍니다.
- 계획된 관리를 투명한 삼바 서비스의 재배치를 통해, 관리자에게 쉽게 클러스터 구성원을 고치거나 업그레이드 가능하게 해줍니다.

- Active-active 설정을 통해, 장비를 최대한 사용할 수 있게 하여 줍니다. Active-active관련된 좀더 자세한 내용은 뒤에 보실 수 있습니다.

주의

완벽한 삼바 설정의 설명은 이 문서에서 다룰 수가 없이 방대합니다. 그래서 이 문서에서는 클러스터 중에 필요한 부분만 강조 하였습니다. 삼바에 대한 좀 더 자세한 내용은 공식 *Red Hat Linux* 사용자 정의 가이드를 참조 하여 주십시요. 또한, 다음의 URL에서 필요한 삼바 설정 자료를 얻으실 수 있습니다. http://www.redhat.com/support/resources/print_file/samba.html 고성능 삼바 서비스를 설정하려면, 클러스터 되지 않은 삼바 파일 제공에 대해 아는 것이 필요하다.

6.2.1. 삼바 서버 요구사항

만일 고성능 삼바 서비스를 만드려고 하려면, 모든 클러스터 서버들이 몇가지 요구 사항을 충족시켜야 합니다. 요구 사항들이란 다음과 같습니다:

- 삼바 RPM 패키지들이 설치되어 있어야만 합니다. Red Hat Linux 어드밴스 서버 에는 다음과 같은 삼바와 관련된 패키지들이 있습니다: `samba` 그리고 `samba-common`. 여기서 눈여겨 볼것은 고성능 삼바 RPM들을 만들기 위해 평상시의 RPM을 수정하지 않았다는 것입니다.
- 삼바 데몬은 서비스 기반으로 클러스터 체제에서 시작되고 정지될 것입니다. 따라서, 삼바의 설정은 평상시의 `/etc/samba/smb.conf` 파일안에 저장되지 말아야 합니다 . 자동 시스템 시작의 삼바 데몬들인 `smbd`와 `nmbd`은 `init.d` 런레벨에서 비활성화 되어야 합니다. 예를 들어: `chkconfig --del smb`.
- 클러스터 체제에서 클러스터와 관련된 삼바의 데몬들을 적당히 멈춰주기 때문에, 시스템 관리자는 평상시에 사용되는 삼바 멈춤 스크립트 (예. `service smb stop`)를 사용하지 말아야 합니다. 이유는 이것이 모든 클러스터에 관련된 삼바 데몬들을 멈출 것이기 때문입니다.
- 클러스터된 삼바 서비스를 위한 파일 시스템들은 `/etc/fstab`에 포함되지 말아야 합니다. 클러스터된 삼바 서비스용 파일 시스템들은 `cluadmin` 설정에 명시된 변수들을 통해 입력되어야 합니다.
- 삼바 프린터 공유는 아직 자동 오류 복구되지 않습니다.

6.2.2. 삼바 운영 모델

이곳에서는 고성능 삼바를 지원하는데 기반이 되는 응용 모델에 관련된 자료를 가지고 있습니다. 이 모델을 아는 것이 클러스터 된 삼바 서비스의 설정 요구 사항을 이해하는데 큰 도움이 될 것입니다.

보통, 클러스터 되지 않은 삼바 설정 모델은 `/etc/samba/smb.conf` 파일을 수정함으로 윈도우 클라이언트들에게 어떤 파일 시스템이 보일지 지정할 수 있습니다. 또한 이 파일을 통해, 사용 허가와 다른 맵핑과 관련된 것을 지정할 수 있습니다. 단순한 시스템 모델에서는 각 `smbd`와 `nmbd` 데몬들이 `/etc/rc.d/init.d/smb` 런레벨 스크립트를 통해 자동적으로 시작됩니다.

고성능 삼바 서비스를 구현하기 위해서는, 하나의 `/etc/samba/smb.conf` 파일을 가지는 것보다, 각각 서비스 기반의 삼바 설정 파일을 가지고 있습니다. 이것들은 다음과 같이 불려집니다: `/etc/samba/smb.conf.`*`sharename`* 이곳에서, *`sharename`*은 삼바 서비스와 관련된 각 설정파일의 이름을 나타냅니다. 예를 들어 이것은 공유가 *`eng`* 혹은 *`acct`*일때, 알맞은 삼바 설정 파일은 `/etc/samba/smb.conf.`*`eng`* 와 `/etc/samba/smb.conf.`*`acct`*일 것입니다.

`smb.conf.`*`sharename`* 파일의 포맷은 기본 `smb.conf` 파일의 포맷과 같습니다. 클러스터를 위해 따로 덧붙여진 것이 아무것도 없습니다. 하지만, 좀 더 부드러운 클러스터의 동작을 위해서 `smb.conf.`*`sharename`* 내에서 꼭 필요한 것들은 몇가지 있습니다; 이것들은 뒤에서 좀 더 자세하게 다루어 질 것입니다. `cluadmin`를 통해 새로운 삼바 서비스가 생성될때는 서비스 기반 변수들과 기본 `smb.conf.`*`sharename`* 파일을 사용해 설정파일이 생성 됩니다. 이 파일은 후에 시스템 관리자가 필요한 윈도우 클라이언트 시스템과 필요한 디렉토리와 알맞은 허가 관련 정보를 넣을 수 있는 기본 틀이 될 것입니다.

시스템 관리자는 모든 클러스터 구성원에 /etc/samba/smb.conf.*sharename* 파일을 복사해 두어야 합니다. 기본 설정 후에, /etc/samba/smb.conf.*sharename*이 수정되었다면, 수정된 것을 다른 클러스터 구성원에 복사해야 합니다.

고성능의 삼바 기능을 사용하기 위해서는, 각 삼바 서비스가 클러스터 안에서 설정 되어야 합니다 (cluadmin를 통해). 이를 통해 각 클러스터 구성원 안에 **smbd**/nmbd 데몬들을 갖게 됩니다. 따라서, 만일 하나 이상의 삼바 서비스가 클러스터에 설정 된다면, 여러개의 이 데몬들이 실행 되는 것을 볼 수 있습니다. 이 삼바 데몬들, **smbd**/nmbd은 보통 우리가 사용하는 init.d 런레벨 스크립트를 통해 시작되지 않으며; 클러스터 체제속에서, 어떤 노드이던지 서비스 지원을 시작하면 시작됩니다.

한 시스템에서 여러 삼바 데몬을 실행하기 위해서, 시작된 데몬들은 각자의 잠금 디렉토리 (locking directory)를 가져야만 합니다. 따라서, 이것은 서비스에 기본한 삼바 데몬 잠금 디렉토리가 됩니다. 이 디렉토리는 /var/cache/samba/*sharename* 의 이름을 가지고 있으며, 이곳에서 *sharename*은 서비스 설정 (cluadmin)에서 지정된 삼바 공유 이름입니다. 전의 예를 따르면, 다음과 같은 잠금 디렉토리가 생길 것입니다: /var/cache/samba/*eng* 와 /var/cache/samba/*acct*.

삼바 서비스를 설정하기 위해 cluadmin를 사용하면, 자동적으로 cluadmin를 실행하는 클러스터 시스템에 /var/cache/samba/*sharename* 디렉토리가 생성됩니다. 여기서 수동으로 다른 클러스터 구성원에 잠금 디렉토리를 생성하라는 주의가 나타날 것입니다. 예를 들어: mkdir /var/cache/samba/*eng*.

6.2.3. 삼바 서비스 설정 변수들 정리하기

삼바 서비스 설정을 준비할때, 윈도우 기반 클라이언트 들에게 어떤 파일 시스템을 지원할지 설정에 관련된 정보를 정해야 합니다. 다음 정보가 NFS 서비스를 설정하기 위해 필요한 정보입니다:

- 서비스 명 — 클러스터 내에서 서비스를 구별하기 위한 이름.
- 우선 구성원 — 하나 이상의 클러스터 구성원이 연결되어 있을때, 삼바 서버가 실행되어질 서버를 명시하는 것.
- 재배치 방침 — 기본 적으로 서비스가 시작될때, 우선 구성원이 클러스터에 연결 되어 있지 않았으며, 나중에 우선 구성원이 연결되었을때, 서비스를 재배치 할지를 나타냅니다. 이 변수는 부하분산의 방법중 하나로, 부하를 클러스터 구성원 사이에 균일하게 나눌수 있습니다.
- 상태 검사 시간 — 삼바 서비스와 관련된 삼바 데몬들, smbd/nmbd이 제대로 실행되고 있는지를 얼마나 자주 (초 단위) 확인 하는지를 명시합니다. 두 데몬들이 생각치 않게 멈췄다면, 자동적으로 서비스를 재시작하기 위해 시작될 것입니다. 만일 0이 입력이 된다면, 시스템이 검사하지 않을 것입니다. 예를 들어, 만일 90초로 정한다면 매 그 시간마다 검사를 할 것입니다.
- *IP* 주소 — 윈도우 클라이언트가 파일 공유를 IP 주소와 연결되어 있는 서버를 통해 사용할 것입니다. 윈도우 클라이언트가 특정 클러스터 구성원이 삼바 서버로 사용되는 것을 알지 못하게 하려면, 클러스터 구성원의 호스트이름을 서비스가 사용되는 IP 주소로 사용하지 말아야 합니다. 클러스터된 삼바 서비스에는 클러스터 서버들의 IP 주소와 다른 유동 IP 주소가 주어질 것이며 이것을 사용하십시요. 이 유동 IP 주소는 공유를 제공하고 있는 어떤 클러스터 구성원이든지 설정이 될 것입니다. 이 방법을 따라, 윈도우 클라이언트는 유동 IP 주서만 알것이며, 클러스터가 된 삼바 서비스가 사용중이라는 것을 알지 못합니다. 삼바 서비스의 IP 주소를 입력할때, 관계된 넷마스크과 브로드캐스트의 주소도 입력하라 할 것입니다. 만일 NONE을 기본으로 선택하면, 네트워크 인터페이스에 설정된 넷마스크와 브로드캐스트와 일치할 것입니다.
- 마운트 정보 — 클러스터 되지 않은 파일 시스템에서는, 마운트 정보는 보통 /etc/fstab에 정장됩니다. 하지만, 클러스터된 파일 시스템은 /etc/fstab에 저장되면 안됩니다. 이것을 통해 한번에 한 클러스터 구성원만 파일 시스템을 마운트하게 할 수 있습니다. 그렇지 않으면, 파일 시스템의 문제를 초래하며, 시스템 전체의 문제를 일으킬 가능성이 높습니다.
  - 장치 특별 파일 — 마운트 정보는 디스크의 장치 특별 파일과 파일 시스템이 마운트될 디렉토리를 지정합니다. 삼바 서비스를 설정하며 이 정보를 입력하도록 프로그램에서 묻습니다.
  - 마운트 지점 디렉토리 — 삼바 서비스는 하나 이상의 파일시스템 마운트를 가질 수 있습니다. 이렇듯이, 파일 시스템들이 하나의 자동 오류 복구로 묶일 수 있습니다.
  - 마운트 옵션 — 마운트 정보는 또한 마운트 옵션을 지정합니다.

- 강제 마운트 해제 — 마운트 정보의 일부로, 강제 마운트 해제가 활성화 되어야 하는지 아닌지를 정해야 합니다. 강제 마운트 해제가 활성화 되면, 서비스가 비활성화 되거나 재배치 될때, 파일 시스템을 사용중인 프로그램이 정지되어 파일 시스템의 마운트 해제 가능하도록 하여줍니다.

- *Export* 정보 — 이 정보는 NFS서비스에서만 필요합니다. 만일 파일 제공을 윈도우 기반의 클라이언트에만 한다면, NFS exports에 대한 질문에 no로 대답합니다. 또한 NFS exports 변수와 삼바 공유 변수를 지정함으로 다기능 파일 제공을 할수 있도록 서비스를 설정할 수 있습니다.
- 삼바 공유 명 — 서비스를 설정하는 중, 파일 시스템을 윈도우 클라이언트들과 공유하고 싶은지 질문을 받게 될것 입니다. 만일 대답을 yes로 한다면, 삼바 공유 명을 입력해야 합니다. 이곳에 지명한 이름에 따라, 알맞는 `/etc/samba/smb.conf.`*`sharename`* 파일명과 잠금 디렉토리 `/var/cache/samba/`*`sharename`* 가 생성됩니다. 기본적으로 `smb.conf.`*`sharename`* 에 지정한 윈도우 공유 이름이 이 변수와 맞습니다. 서비스당 최대한 한개의 삼바 설정을 할 수 있으며, 첫 장치와 지정되어야 합니다. 예를 들어, 여러개의 디스크 장치 (그리고 적당한 파일 시스템 마운트 가지고 있다면)를 한 서비스 안에 가지고 있다면, 서비스에 하나의 *sharename* 공유 이름을 지정할 수 있습니다. 그리고 `/etc/samab/smb.conf.`*`sharename`* 파일안에, 여러개의 삼바 공유를 여러 장치중의 디렉토리와 지정할 수 잇습니다. 삼바 공유 서비스를 비활성화 시키려면, 공유 이름을 **None**으로 지정할 수 있습니다.

삼바 서비스를 설정하려고 `cluadmin`를 실행할때:

- 바른 변수를 입력하도록 신경을 써 주십시요. 삼바 변수와 관련된 확인 로직이 현재 그리 똑똑하지 못합니다.
- 거의 모든 질문에, [?]를 입력함으로, 좀 더 상세한 도움을 얻을 수 있습니다.
- `cluadmin`를 사용해 삼바 서비스를 설정한 후, `/etc/samba/smb.conf.`*`sharename`*를 각 서비스, 클라이언트, 허가에 맞게 조절하는 것을 잊지 마십시요.
- `smb.conf.`*`sharename`* 파일을 다른 클러스터 구성원에 복사하는 것을 잊지 마십시요.
- 명시된 단계에 따라, 다른 클러스터 구성원에 삼바 데몬의 잠금 디렉토리를 생성하여 주십시요, 예를 들어: `mkdir /var/cache/samba/`*`acct`*.
- 만일 삼바 서비스를 제거하신다면, `/etc/samba/smb.conf/`*`sharename`* 파일을 제거하시는 것을 잊지 마십시요. `cluadmin` 프로그램은 혹시 나중에 있을지 모르는 사이트 관련 설정 변수를 보존하기 위해 자동적으로 이 파일을 제거하지 않습니다.

6.2.4. 삼바 서비스 설정 예제

삼바 서비스의 설정 단계를 제대로 나타내기 위해 이 곳에서는 예제 설정을 하고 있습니다. 이 예제는 하나의 네멍의 총무부 사람들의 홈 디렉토리를 삼바 공유로 설정하는 것을 보여주고 있습니다. 총무부 사람들은 이것을 그들의 윈도우 기반의 시스템에서 사용할 것입니다.

다음은 서비스 변수와 약간은 설명이 입니다.

- 서비스 명 — `samba_acct`. 이 이름은 서비스의 주 목적인 총무부 사람들에게 export 하는 것을 표현하기 위해 선택 되었습니다.
- 우선 구성원 — **clu4**. 이 예제 클러스터에는, 구성원이 clu3과 clu4가 있습니다.
- 사용자 스크립트 — 클러스터 체제에 삼바 서비스의 지원이 포함 되어 있습니다. 그래서, 삼바 서비스를 설정할때는 사용자 스크립트를 정하지 않아도 됩니다. 이 이유에서, 사용자 스크립트를 원하게 되면, 기본값인 **None**을 선택 하십시요.
- 검사 시간단위 — **90** 초.
- *IP* 주소 — **10.0.0.10**. 여기에는 cluacct라는 호스트 이름이 연결되어 있으며, 이 이름으로, 윈도우 기반의 클라이언트들이 공유를 사용할 것입니다. 기억할 것은, 이 IP 주소는 모든 클러스터 구성원 (clu3와 clu4)와 다르다는 것입니다. 기본 넷마스크와 브로드캐스트가 사용될 것입니다.

- 마운트 정보 — /dev/sdb10 이것은 파일 시스템이 있을 RAID 장치 안의 파티션을 나타냅니다. **ext2** 는 파일 시스템 유형을 나타내며, 파일 시스템이 형성될때 지정되었습니다. /mnt/users/accounting은 파일 시스템 마운트 지점을 나타냅니다. rw,nosuid,sync 은 마운트 옵션입니다.
- *Export* 정보 — 이 예제의 간단함을 위하여, 파일 시스템이 NFS로 export 되지 않습니다.
- 공유 명 — acct. 이 공유 명은 윈도우 기반의 클라이언트들이 공유할 이름입니다. 예를 들어, e.g. \\10.0.0.10\acct.

다음이 클러스터 에서 사용되는 IP 주소와 관련된 호스트 이름이 입력된 /etc/hosts를 출력한 것입니다:

```
10.0.0.3    clu3     # cluster member10.0.0.4    clu4     # second cluster member10.0.0.10    cluacct  # floating IP address associated with accounting team NFS s
```

다음은 cluadmin을 사용하여 이 예제의 삼바 서비스를 설정하는 것입니다:

```
Service name: samba_acct
Preferred member [None]: clu4
Relocate when the preferred member joins the cluster (yes/no/?) [no]: yes
User script (e.g., /usr/foo/script or None) [None]:
Status check interval [0]: 90
Do you want to add an IP address to the service (yes/no/?) [no]: yes

    IP Address Information

IP address: 10.0.0.10
Netmask (e.g. 255.255.255.0 or None) [None]:
Broadcast (e.g. X.Y.Z.255 or None) [None]:
Do you want to (a)dd, (m)odify, (d)elete or (s)how an IP address, or
are you (f)inished adding IP addresses [f]:
Do you want to add a disk device to the service (yes/no/?) [no]: yes

Disk Device Information

Device special file (e.g., /dev/sdb4): /dev/sdb12
Filesystem type (e.g., ext2, ext3 or None): ext2
Mount point (e.g., /usr/mnt/service1) [None]: /mnt/users/accounting
Mount options (e.g., rw,nosuid,sync): rw,nosuid,sync
Forced unmount support (yes/no/?) [yes]:
Would you like to allow NFS access to this filesystem (yes/no/?)\
  [no]: no
Would you like to share to Windows clients (yes/no/?) [no]: yes

You will now be prompted for the Samba configuration:
Samba share name: acct

The samba config file /etc/samba/smb.conf.acct does not exist.

Would you like a default config file created (yes/no/?) [no]: yes

Successfully created daemon lock directory /var/cache/samba/acct.
Please run 'mkdir /var/cache/samba/acct' on the other cluster member.

Successfully created /etc/samba/smb.conf.acct.
Please remember to make necessary customizations and then copy the file
over to the other cluster member.

Do you want to (a)dd, (m)odify, (d)elete or (s)how DEVICES, or
are you (f)inished adding DEVICES [f]: f
name: samba_acct
preferred node: clu4
relocate: yes
user script: None
monitor interval: 90
```

```
IP address 0: 10.0.0.10
 netmask 0: None
 broadcast 0: None
device 0: /dev/sdb12
 mount point, device 0: /mnt/users/accounting
 mount fstype, device 0: ext2
 mount options, device 0: rw,nosuid,sync
 force unmount, device 0: yes
 samba share, device 0: acct
Add samba_acct service as shown? (yes/no/?) yes
```

서비스를 설정하기 위해 위처럼 cluadmin을 실행한 후에, 기억할 것은:

- /etc/samba/smb.conf.*sharename* 를 알맞게 사용자화 합니다.
- /etc/samba/smb.conf.*sharename* 파일을 각 클러스터 구성원에 복사합니다.
- 알맞은 잠금 디렉토리를 다른 클러스터 구성원에 다음과 같이 설치합니다, e.g. mkdir /var/cache/samba/*acct*

6.2.5. smb.conf.*sharename* 파일 필드

이 곳에서는 고성능 삼바 서비스의 정확한 실행을 위해 필요한 smb.conf.*sharename* 파일 안에 있는 필드들을 설명하고 있습니다. 이 책자안에 삼바 설정에 필요한 모든 설명을 하는 것은 불가능 합니다. 클러스터를 지원하기 위해 따로 필드 이름이 더해지지 않았으며, 파일 유형은 보통 삼바의 규정을 따르고 있습니다.

밑에 보여지는 것이 서비스 기반으로 cluadmin를 통해 자동적으로 생성된 smb.conf.*sharename* 파일의 예제입니다. 이 예제 파일은 위의 cluadmin의 설정과 동일합니다.

```
# Template samba service configuration file - please modify to specify
# subdirectories and client access permissions.
# Remember to copy this file over to other cluster member, and create
# the daemon lock directory /var/cache/samba/acct.
#
# From a cluster perspective, the key fields are:
# lock directory - must be unique per samba service.
# bind interfaces only - must be present set to yes.
# interfaces - must be set to service floating IP address.
# path - must be the service mountpoint or subdirectory thereof.
# Refer to the cluster documentation for details.

[global]
    workgroup = RHCLUSTER
    lock directory = /var/cache/samba/acct
    log file = /var/log/samba/%m.log
    encrypt passwords = yes
    bind interfaces only = yes
    interfaces = 10.0.0.10

[acct]
    comment = High Availability Samba Service
    browsable = yes
    writable = no
    public = yes
    path = /mnt/service12
```

다음에 설명된 필드들이 /etc/samba/smb.conf.*sharename* 파일 안에서 클러스터화 하는 데 가장 많이 쓰이는 필드들입니다. 이 예제에서, 파일의 이름은 /etc/samba/smb.conf.*acct* 이며, cluadmin 실행 중에 지정된 공유 이름인 *acct*과 맞습니다. 클러스터와 관련된 필드는 다음과 같습니다. 다른 필드들은 보통 삼바 규칙을 따르고 있으며, 알맞게 사용자화 될 수 있습니다.

전체적인 변수

‘ 이 변수들은 smb.conf.*sharename* 파일안에 사용된 모든 공유에 사용됩니다. 주의할 것은 하나 이상의 공유를 이 파일안에 지정할 수 있으며, 서비스의 파일 시스템 마운트 안에 디렉토리들이 지정 되어야 합니다.

잠금 디렉토리

‘ 삼바 데몬인 smbd/nmbd 파일들이 자신들의 잠금 파일들을 저장할 디렉토리를 나타냅니다. 이것은 /var/cache/samba/*sharename* 이 규칙을 따라야 하며, 여기서 *sharename*은 cluadmin에 주어진 변수에 따라 달라질 수 있습니다. 잠금 디렉토리는 서비스 기반의 smbd/nmbd을 갖기위해 꼭 필요합니다.

bind 인터페이스 만

‘ 이 변수는 smbd/nmbd가 유동 IP주소와 관련된 클러스터된 삼바 서비스를 하나로 맞추기 위해 꼭 필요하기에 **yes**를 선택하여 줍니다.

인터페이스들

‘ 삼바 서비스와 연관된 IP 주소를 지정합니다. 만일 서비스안에 넷마스크가 지정되어 있다면, 다음 예처럼 보일 것입니다: *interfaces = 10.0.0.10/255.255.254.0*

공유 관련 변수

‘ 삼바 공유와 관련된 변수들을 지정합니다.

쓰기가능

‘ 기본적으로, 공유 사용의 허가는 쓰기-불가능입니다. 이 변수를 정함으로, 사용자의 필요에 알맞게 정할 수 있습니다.

경로

‘ 서비스 설정에 지정된 첫 파일 시스템 마운트 지점을 기본으로 하고 있습니다. 이것은 윈도우 클라이언트들과 공유하기에 알맞게 디렉토리나 내부 디렉토리로 수정되어야 합니다.

6.2.6. 윈도우 클라이언트의 삼바 공유 사용

윈도우 클라이언트들은 공유를 사용할때, 고성능 클러스터에 의해 제공되는 것에 대해 거의 아무것도 느끼지 못합니다. 윈도우 클라이언트에서 바라볼때, 오직 충족시켜야 하는 것은 삼바 공유를 cluadmin, e.g. 10.0.0.10. 에서 지정된 유동 IP 주소 (혹은 연결된 호스트명)으로 한다는 것입니다. 윈도우 클라이언트는 절대로 클러스터 구성원 (e.g. clu3 혹은 clu4)의 시스템 IP 주소로 공유를 사용해서는 안됩니다.

클러스터 환경에서 정해진 인증 방법에 따라 다르지만, 윈도우 계정 정보를 클러스터 서버들에 만들기 위해 smbpasswd를 사용해야 할 수도 있습니다. 이 계정들을 만들때, 모든 클러스터 구성원에 동일한 계정 정보를 넣어주어야 합니다. 이것은 모든 클러스터 구성원에 동시에 smbpasswd를 실행하여 할 수도 있으며, 혹은 먼저 한 시스템에서 실행 한후, 결과물인 /etc/samba/smbpasswd파일을 다른 시스템에 복사할 수도 있습니다. 예를 들어, sarge라는 이름의 윈도우 클라이언트 시스템을 활성화 시키려면, 다음과 같은 명령어를 모든 클러스터 구성원에 실행 시키시고, 같은 사용자명과 비밀번호를 사용하도록 하십시요:

```
smbpasswd -a sarge
```

윈도우 클라이언트에서, 삼바 공유는 평상시의 방법대로 사용될 수 있습니다. 예를 들어, 메인 태스크 바에서 **Start**를 클릭하시고, **Run**을 선택하십시요. 이를 통해 삼바 공유 이름을 지정할 수 있는 대화창이 나타날 것입니다. 예를 들어:\\10.0.0.10\acct 혹은 \\cluacct\acct. 삼바 공유를 윈도우 클라이언트에서

사용하려면, **Map Network Drive** 기능을 사용할 수도 있습니다. 주의 할 것은 호스트명이 유동 IP를 지정하게 하는 것입니다. 다음의 호스트명/IP 주소는 위의 `/etc/hosts`에서 나온 것입니다; 그 곳에서 고성능 클러스터 공유를 지시하고 있는 것은 `\\cluacct\acct` 입니다. 공유는 클러스터 서버의 이름으로는 사용할 수 없습니다. 예를 들어, 공유를 `\\clu3\acct` 혹은 `\\clu4\acct`로 사용하지 마십시요. 만일 공유가 잘못되어 클러스터 명으로 지시된다면 (예. `\\clu3\acct`), `clu3`에 의해 공유가 될때만, 윈도우 클라이언트가 공유를 사용할 수 있습니다, 그렇기에 고성능의 효과를 볼수 없게 됩니다.

NFS 프로토콜과는 다르게, 윈도우 기반의 CIFS/SMB 프로토콜은 좀더 안정적입니다. 그렇기에, 윈도우의 환경에서는, 삼바 서버에서의 즉각적인 반응이 없을 경우 적당한 행동을 취하는 것은 각 프로그램의 엉덕입니다. 계획된 서비스 재배치 혹은 진짜 자동 오류 복구 시나리오에서는, 윈도우 클라이언트가 삼바 서버로 부터 즉각적인 반응을 받지 못하며, 어느 정도의 시간이 필요하게 됩니다. 윈도우 프로그램은 이 시간동안 재시도를 하지 않을 것입니다.

제대로 된 프로그램은 알맞게 서비스의 응답을 위해 재시도 할것이며, 이를 통해 윈도우 클라이언트들은 서비스 재배치나 자동 오류 복구를 눈치채지 못할 것입니다. 거기에 빗대어, 제대로 되지 못한 프로그램은, 자동 오류 복구나 서비스 재배치에 공유를 사용하지 못한다는 에러 메세지를 보이게 될 것입니다. 자동 오류 복구 동안이나 서비스 재배치 동안 삼바 공유를 제대로 사용하지 못하게 된다면, 프로그램에서 재시도 하거나, 프로그램을 재실행 하는 것이 필요합니다.

윈도우 기반의 클라이언트들의 자동 오류 복구에 대한 반응은 어떤 윈도우가 설치되어 있는 지에 따라 다릅니다. 예를 들어, 윈도우 98 기반의 시스템은 "The network path was not found"라는 에러 메세지를 나타낼 것이며, 윈도우 2000 기반은 같은 환경에서 자동으로 재시도 합니다.

7장 .
아파치 서비스

이 장에서는 Red Hat Linux 어드밴스 서버에 있는 아파치 서버를 고성능 (highly available)으로 만들기 위해 필요한 설정 방법들을 포함하고 있습니다.

7.1. 아파치 서비스 설정하기

이 곳에서는 아파치 웹 서버를 고장 방지 (fail over) 클러스터 서비스로 설정하는 예를 제시하고 있습니다. 비록 실제로 서비스에 사용되는 변수들은 특정 설정에 따라 다르지만, 이 예를 보시면 특정 환경에 대한 서비스를 설정하는데 도움이 될 것입니다.

아파치 서비스를 설정하시려면, 양쪽 클러스터 시스템들을 아파치 서버로 설정하셔야 합니다. 클러스터 소프트웨어는 한번에 한 시스템에서만 아파치가 운영될 수 있도록 합니다. 아파치 설정은 아파치 **RPM**을 양쪽 클러스터 멤버에 설치하는 것과 웹 사이트를 저장할 공유 파일 시스템을 설정하는 작업으로 구성됩니다.

아파치 소프트웨어를 클러스터 시스템에 설치시, 클러스터 시스템이 부팅할때 자동으로 시작하지 않도록 다음과 같은 명령어: `chkconfig --del httpd`를 설정때 사용해 주십시요. 시스템이 시작하면서 httpd를 생성시키는 것 보다, 클러스터 자체에서 아파치 서비스에 알맞는 클러스터 서버에서 시작할 수 있게 해주는 것이 좋습니다. 이것을 통해 한번에 오직 한 클러스터 구성원에만 알맞는 IP 주소와 파일 시스템을 마운트할 수 있습니다.

아파치 서비스를 추가할 경우, "유동" IP 주소가 할당되어야 합니다. 클러스터 구조에서 이 IP주소를 현재 아파치 서비스가 실행되고 있는 네트워크 인터페이스에 할당할 것입니다. 이 IP 주소는 아파치 서버를 이용하려는 HTTP 클라이언트에게 아파치 소프트웨어가 실행되고 있는 클러스터 시스템을 이용할 수 있도록 도울 것입니다.

웹페이지의 내용을 포함하는 파일 시스템은 클러스터 시스템이 시작할 때, 자동적으로 공유 디스크에 마운트되면 안됩니다. 그 대신, 클러스터 시스템들의 아파치 서비스가 시작되고 끝날때 마운트되거나 마운트해제되어야 합니다. 이를 통해, 데이터 상에 무리를 일으킬 수 있는 여럿의 클러스터 시스템이 동시에 같은 데이터를 사용하려는 경우를 막을 수가 있습니다. 따라서 이 파일 시스템을 `/etc/fstab`에 포함 시키지 마십시요.

아파치 서비스를 설정하는 것은 다음과 같은 4 단계로 나눌수 있습니다:

1. 서비스를 위한 공유 파일 시스템을 설정. 이 파일 시스템은 웹 사이트의 내용을 저장하는데 사용됩니다.
2. 양쪽 클러스터 시스템에 아파치 소프트웨어 설치.
3. 양쪽 클러스터 시스템에 아파치 소프트웨어 설정.
4. 클러스터 데이터베이스에 서비스 추가.

아파치 서비스를 위해 공유 파일 시스템을 설정하려면, 클러스터 시스템 상에서 root 사용자로 로그인 하신 후 다음과 같은 작업을 수행하시기 바랍니다:

1. 공유 디스크 상에서 상호 대화식 **fdisk** 유틸리티를 사용해서 아파치 문서의 기본 디렉토리로 사용될 파티션을 만듭니다. 기억하실 것은, 다른 디스크 파티션에 여러 문서 기본 디렉토리를 만들 수 있다는 것입니다. 자세한 내용은 2.4.4.4 절을 참조 하십시요.
2. `mkfs` 명령어를 사용해 바로 전단계에서 만드신 파티션에 ext2 파일 시스템을 만드실 수 있습니다. 드라이브 글자와 파티션 번호를 지정 하십시요. 예를 들어:
   `mkfs /dev/sde3`
3. 웹 내용을 저장할 파일 시스템을 아파치 문서 기본 디렉토리에 마운트하십시요. 예를 들어:
   `mount /dev/sde3 /var/www/html`

   이 마운트 정보를 `/etc/fstab` 파일에 저장하지 마십시요. 그 이유는 클러스터 소프트웨어는 서비스에 사용되는 파일 시스템만 마운트/마운트 해제할 수 있기 때문입니다.

4. 필요한 파일들을 모두 문서 기본 디렉토리로 복사하십시요.
5. 만일, CGI 파일들이나 혹은 그와 같이 다른 디렉토리나 다른 파티션에 있어야 하는 파일들이 있다면, 전과 같은 단계들을 필요한 만큼 반복하십시요.

아파치가 양쪽 클러스터 시스템에 꼭 설치되어 있어야 합니다. 기억하실 것은 시스템에 문제가 생겼을 시에, 자동 복구가 되려면, 기본 아파치 서버 설정이 양쪽 클러스터 시스템에 같아야 한다는 것입니다. 다음의 예는 삼자 프로그램이나 어떤 모듈 혹은 특정 맞춤이 없는 기본 아파치 웹 서버의 설치를 보여줍니다. 만일 아파치에 모듈들이나, 더 좋은 성능을 위해 조율을 하시려면, 아파치 설치 디렉토리에 있는 문서들을 참조 하시던지, 아파치 웹 사이트, http://httpd.apache.org/docs-project/를 참조 하십시요.

양 클러스터 시스템에, 아파치 RPM들을 설치 하십시요. 예를 들면:

```
rpm -Uvh apache-1.3.20-16.i386.rpm
```

클러스터 시스템을 아파치 서버로 설정하려면, 아파치 설정 파일인, `httpd.conf`을 조율하십시요. 그리고 아파치 서비스를 시작하고 정지시킬 스크립트를 만드십시요. 그 후, 그 파일들을 다른 클러스터 시스템에 복사하십시요. 서버 이상시, 자동 복구되려면, 양쪽 클러스터 시스템에 있는 파일들이 정확히 일치해야 합니다.

한 시스템에, 다음과 같은 작업을 수행하십시요:

1. 아파치의 설정 파일인 `/etc/httpd/conf/httpd.conf`을 사용자의 필요에 따라 설정하십시요. 예를 들어:
   - HTML파일들이 있을 디렉토리를 지정해 주십시요. 클러스터 데이터베이스에 아파치 서비스를 더하실 때, 이 마운트 지점을 명시하셔야 합니다. 하지만, 이것은 기본 설정인 `/var/www/html`과 다른 경우에만 바뀌 주시면 됩니다. 예를 들어:
     ```
     DocumentRoot "/mnt/apacheservice/html"
     ```
   - 만일 스크립트 디렉토리가 기본 위치인 곳에 있지 않는다면, CGI 프로그램이 상주할 디렉토리를 지정해 주어야 합니다. 예를 들어:
     ```
     ScriptAlias /cgi-bin/ "/mnt/apacheservice/cgi-bin/"
     ```
   - 전 단계에서 사용된 경로를 지정하시고, 알맞는 허가를 그 디렉토리에 주어야 합니다. 예를 들어:
     ```
     <Directory mnt/apacheservice/cgi-bin">
     AllowOverride None
     Options None
     Order allow,deny
     Allow from all
     </Directory>
     ```
     또한 아파치를 손보거나, 다른 모듈들의 기능을 향상시키기 위해 다른 일들을 해야할 경우도 있습니다. 다른 옵션을 설정하는데 있어 보다 자세한 정보를 원하신다면, 아파치 웹사이트에 있는 아파치 프로젝트 문서, http://httpd.apache.org/docs-project/를 참조하시기 바랍니다.

2. 기본 아파치 시작 스크립트인, `/etc/rc.d/init.d/httpd`이 클러스터 상에서도 아파치 서비스를 활성 클러스터 멤버에서 시작하고 정지하는데 사용됩니다. 그와 같이, 서비스를 설정시 **사용자 스크립트**가 요구된다면 이 스크립트를 지정해 주십시요.

주의

출시 버전에 따라, 기본 아파치 서비스 스크립트인 `/etc/rc.d/init.d/httpd`을 사용한 `service httpd status` 명령어가 현 httpd의 상태를 보고하지 못할 수도 있습니다. 이것이 클러스터가 서비스를 관찰하는 것 (서비스의 감시 주기를 서비스 설정때 설정하였다면)을 방해할 수도 있습니다. Status 행이 다음과 같이 나타날 것 입니다:

```
status)
   status $httpd
   ;;
```

만일 그렇다면, 아파치의 서비스를 제대로 모니터할 수 있도록 다음과 같은 줄을 Status 행에 더해주십시요:

```
status)
```

```
        status $httpd
        RETVAL=$?
        ;;
```

아파치 서비스가 클러스터 데이터베이스에 더해지기 전에, 아파치 디렉토리들이 마운트 해제되었는지 확인하여 주십시요. 그 후, 클러스터 시스템 중 하나에 서비스를 더해 주십시요. 아파치 서비스를 실행 중인 클러스터 시스템 상의 네트워크 인터페이스에 연결할 IP 주소를 지정하시기 바랍니다.

다음은 아파치 서비스를 추가하기 위해 cluadmin를 사용하는 예입니다.

```
cluadmin> service add apache

 The user interface will prompt you for information about the service.
 Not all information is required for all services.

 Enter a question mark (?) at a prompt to obtain help.

 Enter a colon (:) and a single-character command at a prompt to do
 one of the following:

 c - Cancel and return to the top-level cluadmin command
 r - Restart to the initial prompt while keeping previous responses
 p - Proceed with the next prompt

Preferred member [None]: devel0
Relocate when the preferred member joins the cluster (yes/no/?) \
 [no]: yes
User script (e.g., /usr/foo/script or None) [None]: \
/etc/rc.d/init.d/httpd

Do you want to add an IP address to the service (yes/no/?): yes

    IP Address Information

IP address: 10.1.16.150
Netmask (e.g. 255.255.255.0 or None) [None]: 255.255.255.0
Broadcast (e.g. X.Y.Z.255 or None) [None]: 10.1.16.255

Do you want to (a)dd, (m)odify, (d)elete or (s)how an IP address,
 or are you (f)inished adding IP addresses: f

Do you want to add a disk device to the service (yes/no/?): yes

    Disk Device Information

Device special file (e.g., /dev/sda1): /dev/sdb3
Filesystem type (e.g., ext2, reiserfs, ext3 or None): ext3
Mount point (e.g., /usr/mnt/service1 or None) [None]: /var/www/html
Mount options (e.g., rw, nosuid): rw
Forced unmount support (yes/no/?) [no]: yes

Do you want to (a)dd, (m)odify, (d)elete or (s)how devices,
 or are you (f)inished adding device information: f

Disable service (yes/no/?) [no]: no

name: apache
disabled: no
preferred node: node1
relocate: yes
user script: /etc/rc.d/init/httpd
IP address 0: 10.1.16.150
 netmask 0: 255.255.255.0
```

```
 broadcast 0: 10.1.16.255
device 0: /dev/sde3
 mount point, device 0: /var/www/html
 mount fstype, device 0: ext3
 mount options, device 0: rw,sync
 force unmount, device 0: yes
 owner, device 0: nobody
 group, device 0: nobody
Add apache service as shown? (yes/no/?) y

Added apache.
cluadmin>
```

주의

고성능 아파치 서비스가 설정된 클러스터에서는 **Red Hat 클러스터 관리자** GUI를 사용할 수 없습니다. 자세한 내용은 9 장를 참조하시기 바랍니다.

8장 .
클러스터 관리하기

다음은 클러스터가 설치되고 설정된 후에 적용되는 여러가지 관리 작업에 대해 자세히 설명하고 있습니다.

8.1. 클러스터와 서비스 상태 보이기

클러스터와 서비스의 상태를 감시하는 것은 클러스터 환경에서 문제를 알아내고 해결하는데 큰 도움이 됩니다. 다음은 클러스터의 상태를 보여주는데 도움이 될 것입니다:

- `clustat` 명령
- 로그 파일 메시지
- 클러스터 감시 GUI

주의 하실 것은 상태는 항상, 클러스터 시스템에 관리 프로그램이 실행중인 상태를 나타냅니다. 좀 더 자세한 클러스터 상태를 얻으시려면, 모든 클러스터 시스템에 관리 프로그램을 실행하시기 바랍니다.

클러스터와 서비스 상태에는 다음과 같은 정보가 포함되어 있습니다:

- 클러스터 구성원 시스템 상태
- 전원 스위치 상태
- Heartbeat 채널 상태
- 서비스 상태, 어느 클러스터 시스템이 서비스를 실행 중인지 혹은 주인 시스템인지
- 클러스터 시스템의 서비스 감시 상태

다음 표에서는 `clustat`와 클러스터 GUI를 통해 보여지는 정보를 어떻게 분석하는지에 대해 나와 있습니다.

구성원 상태	설명
UP	구성원 시스템이 다른 시스템과 통신하고 있으며, quorum 파티션을 사용하는데 문제가 없습니다.
DOWN	구성원 시스템이 다른 구성원 시스템과 통신할 수 없습니다.

표 8-1. 구성원 상태

전원 스위치 상태	설명
OK	전원 스위치가 제대로 작동하고 있습니다.
Wrn	전원 스위치의 상태를 알 수 없습니다.
Err	전원 스위치 상태 파악에 실패나 문제가 생겼습니다.
Good	전원 스위치가 제대로 작동하고 있습니다.
Unknown	다른 클러스터 구성원이 **DOWN**입니다.
Timeout	전원 스위치가 전원 데몬의 명령에 반응하지 않습니다, 이것은 병렬 케이블의 문제일 수 있습니다.

전원 스위치 상태	설명
Error	전원 스위치 상태 파악에 실패나 문제가 생겼습니다.
None	클러스터 설정에 전원 스위치가 포함되어 있지 않습니다.
Initializing	스위치가 초기화 하는 중에 있어서, 확실한 상태 파악이 되지 않았습니다.

표 8-2. 전원 스위치 상태

Heartbeat 채널 상태	설명
OK	Heartbeat 채널이 제대로 작동 중 입니다.
Wrn	채널의 상태를 알 수 없습니다.
Err	채널의 상태를 점검하는데 실패하거나, 문제가 있습니다.
ONLINE	Heartbeat 채널이 제대로 작동 중입니다.
OFFLINE	다른 클러스터 구성원은 **UP**으로 보이나, 현 채널에서의 heartbeat 요청에 답하지 않습니다.
UNKNOWN	이 채널상의 다른 클러스터 구성원의 상태를 파악할 수 없습니다, 이것은 시스템이 **DOWN**이거나, 클러스터 대몬이 현재 작동 중이지 않을 수 있습니다.

표 8-3. Heartbeat 채널 상태

서비스 상태	설명
running	서비스 관련 자원들이 설정 되었으며, 서비스의 주 시스템에서 사용 가능합니다. 상태 중에서 **running** 상태는 계속적인 것으로써, 이 상태에서, 서비스는 **stopping** 상태로 바뀔 수 있습니다 (예를 들어, 만일 우선 구성원이 클러스터에 연결되었을때)
disabled	서비스가 비활성화 되었습니다, 또한 지명된 주 클러스터가 없습니다. **disabled** 상태는 지속적인 것으로써, 이 상태에서, 서비스는 **starting**로 바뀔 수 도 있습니다 (만일 사용자가 서비스를 시작한다면).
starting	서비스가 시작되고 있습니다. **starting**은 지속적인 상태입니다. 서비스가 확실하게 성공적으로 시작되거나 실패될 때까지 서비스는 **starting** 상태로 지속될 것입니다. 이 상태에서, 서비스는 **running** 상태로 (서비스가 성공적으로 시작되었다면), **stopped** 상태로 (만일 서비스 시작에 실패 하였다면), 혹은 **error** 상태로 (만일 서비스 자원을 확인 할 수 없다면) 바뀔것 입니다.
stopping	서비스가 멈추고 있는 중입니다. **stopping** 상태는 지속적인 상태입니다. 서비스의 멈춤이 성공적이거나 실패하지 않을 경우 서비스는 **stopping** 상태로 지속 될 것입니다. 이 상태에서, 서비스는 **stopped** 상태로 (만일 성공적으로 멈춰졌다면), **running** 상태로 (만일 서비스의 멈춤이 실패하고, 서비스가 다시 시작될 수 있다면) 바뀔 것입니다.
stopped	서비스가 어떤 클러스터에서도 사용되고 있지 않으며, 주 클러스터 시스템이 지정되지 않았고, 어떤 클러스터 시스템에도 자원이 설정되지 않았습니다. **stopped** 상태는 지속적인 상태입니다. 이 상태에서, 서비스는 **disabled** 상태로, (만일 사용자가 서비스의 비활성화를 원하는 경우), 혹은 **starting** 상태로 (만일 주 구성원이 클러스터에 연결되었을 경우) 바뀔 것입니다.

표 8-4. 서비스 상태

현재 클러스터 시스템의 상태를 한눈에 보려면, `clustat`를 사용하십시요. 예를 들어:

```
clustat
Cluster Status Monitor (Fileserver Test Cluster)
07:46:05
Cluster alias: clu1alias.boston.redhat.com

====================Member Status=======================
 Member     Status   Node Id  Power Switch
 -------------- ---------- ---------- ------------
 clu1     Up    0     Good
 clu2     Up    1     Good

==================Heartbeat Status====================
 Name            Type   Status
 ------------------------------ ---------- ------------
 clu1    <-->clu2    network  ONLINE

==================Service Status======================
                 Last      Monitor
Restart
Service    Status  Owner     Transition   Interval Count

 ------------ -------- ------------ --------------- ------------
 nfs1    started clu1    16:07:42 Feb 27 15   0
 nfs2    started clu2    00:03:52 Feb 28 2    0
 nfs3    started clu1    07:43:54 Feb 28 90   0
```

클러스터를 감시하며, 일정 시간 단위로 상태를 보이려면, `clustat`에 `-i` *`time`* 명령어 옵션을 사용하시며, 이곳에서 *`time`*은 각 상태 보고의 초단위를 나타냅니다.

8.2. 클러스터 소프트웨어 시작과 정지

System V인 `/etc/rc.d/init` 디렉토리에 있는 `cluster start` 명령어를 사용하여 클러스터 소프트웨어를 시작합니다. 예를 들어:

```
/sbin/service cluster start
```

System V인 `/etc/rc.d/init` 디렉토리에 있는 `cluster stop` 명령어를 사용하여 클러스터 소프트웨어를 정지합니다. 예를 들어:

```
/sbin/service cluster stop
```

이전 명령어는 클러스터 시스템의 서비스가 다른 클러스터 시스템으로 페일오버 (failover) 되도록 합니다.

8.3. 클러스터 구성원 제거하기

클러스터에서 잠시 구성원을 제거해야할 경우가 있습니다. 예를 들어, 클러스터 시스템에 하드웨어 문제가 생길경우, 시스템에 관리를 하기 위해 재부팅을 하지만, 클러스터에 연결하지 않도록 해야 합니다.

`/sbin/chkconfig` 명령어를 사용하여, 부팅을 하지만, 클러스터에 연결하지 않도록 할 수 있습니다. 예를 들어:

```
/sbin/chkconfig --del cluster
```

만일 시스템이 클러스터에 다시 연결할 수 있으면, 다음과 같은 명령어를 사용합니다:

```
/sbin/chkconfig --add cluster
```

그 후 시스템을 재부팅하던지, System V 디렉토리인 `init` 안에 있는 `cluster start`명령어를 사용합니다. 예를 들어:

```
/sbin/service cluster start
```

8.4. 클러스터 설정 수정하기

언젠가 클러스터 설정을 편집해야 하는 경우가 생길 수 있습니다. 예를 들어, `/etc/cluster.conf`에 있는 클러스터 데이터베이스, heartbeat 채널 혹은 quorum 파티션의 내용을 수정해야 할 수도 있습니다.

클러스터의 설정을 수정하려면, **cluconfig**과 `cluadmin`를 사용하십시요. 절대 `cluster.conf`를 직접 수정하지 마십시요. 클러스터 설정을 수정하려면, 8.2 절에 나와있는데로, 클러스터 소프트웨어를 정지시킨 후 하십시요.

그 후, **cluconfig**를 시작하고, 지정된 곳에 필요한 설정을 수정하십시요. 프로그램을 통해 수정이 끝난 후 에는 클러스터 소프트웨어를 다시 시작하십시요.

8.5. 클러스터 데이터베이스 백업하기와 재복구하기

클러스터 데이터베이스는 정기적으로 백업하는 것이 좋습니다, 특히 클러스터 설정에 어느 정도의 수정을 하기 전에는 필히 하는 것이 필요합니다.

클러스터 데이터베이스를 `/etc/cluster.conf.bak` 파일로 백업하려면, `cluadmin`를 시작하고, `cluster backup` 명령어를 지정하십시요. 예를 들어:

```
cluadmin> cluster backup
```

또한 클러스터 데이터베이스를 다른 파일 이름으로 `cluadmin`와 `cluster saveas` *`filename`*를 사용해서 백업할 수 있습니다.

클러스터 데이터베이스를 재복구 하려면, 다음과 같은 단계를 따르시면 됩니다:

1. System V의 `init` 디렉토리에 있는 `cluster stop`를 사용해서 클러스터 소프트웨어를 정지 하십시요. 예를 들어:
   ```
   /sbin/service cluster stop
   ```
   위의 명령어를 통해 클러스터 시스템의 서비스들이 다른 클러스터 시스템으로 오류 복구될 것입니다.
2. 나머지 클러스터 시스템에, `cluadmin`을 시작하고 데이터베이스를 재복구합니다. `cluster restore` 명령어를 사용해서, `/etc/cluster.conf.bak` 파일로 부터 재복구 할 수 있습니다. 데이터베이스를 다른 파일로 부터 재복구 하려면, `cluster restorefrom` *`file_name`*를 사용하십시요.

   클러스터가 모든 활성화 중인 서비스를 비활성화 시키고, 서비스를 모두 제거한 후, 데이터베이스를 재복구합니다.
3. 멈춰진 시스템의 클러스터 소프트웨어를 재시작하려면, System V 디렉토리인 `init`에 있는 `cluster start`를 사용하면 됩니다. 예를 들어:
   ```
   /sbin/service cluster start
   ```
4. 각 클러스터 서비스는 `cluadmin`를 사용해 재시작할 수 있으며, `service enable` *`service_name`*를 통해서 원하는 서비스를 재시작할 수 있습니다.

8.6. 클러스터 이벤트 기록 수정하기

clupowerd, cluquorumd, cluhbd, 그리고, clusvcmgrd데몬들을 통해 기록되는 이벤트의 보안 수준을 수정할 수 있습니다. 이를 통해서 클러스터 시스템에 있는 데몬들은 동일한 보안 수준에서 메시지를 기록하게 할 수있습니다.

모든 클러스터 시스템들 데몬의 기록 수준을 번경하기 위해서는, cluadmin의 cluster loglevel를 사용하여, 데몬 이름과 보안 수준을 지정하시면 됩니다. 보안 수준은 수준 이름이나, 보안 수준에 알맞은 번호를 사용해 지정할 수 있습니다. 보안 수준은 다음과 같이 0 부터 7의 숫자로 나타낼 수 있습니다:

```
0 -- emerg
1 -- alert
2 -- crit
3 -- err
4 -- warning
5 -- notice
6 -- info
7 -- debug
```

기억하실 것은 클러스터는 지정된 보안 수준과 그 보다 높은 수준의 메세지들을 기록하도록 되어 있습니다. 예를 들어, 만일 quorum 데몬의 메시지의 보안 수준이 2 (**crit**)로 되어 있다면, 클러스터는 **crit**, **alert**, 그리고 **emerg** 보안 수준의 메세지들을 기록할 것입니다. 주의하실 것은, 만일 보안 수준을 낮은 수준인 7 (**debug**)로 둔다면, 시간이 흐르면서, 로그파일이 매우 커지게 됩니다.

다음의 예는 **cluquorumd**데몬이 모든 보안 수준의 메세지들을 기록하도록 지정하는 방법을 보여줍니다:

```
 cluadmin
cluadmin> cluster loglevel cluquorumd 7
cluadmin>
```

8.7. 클러스터 소프트웨어 업데이트하기

Red Hat 클러스터 관리자를 업그레이드 하시기 전에, 2.3.1 절에서 설명된 필요한 모든 소프트웨어가 설치되어 있는지 확인해 주십시오. 클러스터 소프트웨어는 현재 클러스터 데이터베이스를 보존하면서 업데이트가 가능합니다. 클러스터 소프트웨어를 한 시스템에 업데이트 하는데는 10 분에서 20 분 정도가 소요됩니다.

클러스터 소프트웨어를 서비스 정지시간을 최소화하면서 업데이트를 하시려면, 다음과 같은 방법을 따르십시요:

1. 업데이트가 필요한 클러스터 시스템에, cluadmin를 실행하고, 현재의 클러스터 데이터베이스를 백업합니다. 이를 통해서 현재 있는 클러스터 설정 데이터베이스를 보존할 수 있습니다. 예를 들어, cluadmin> 프롬트에서 다음과 같은 명령어를 사용하십시요:
   cluster backup
2. System V의 init 디렉토리에 있는 cluster stop를 사용하여, 업데이트 하려는 첫 클러스터 시스템을 정지 시키십시요. 예를 들어:
   /sbin/service cluster stop
3. 첫 클러스터 시스템에 최신 클러스터 소프트웨어를 설치합니다. 그렇지만, **cluconfig** 프로그램이 현재의 클러스터 데이터베이스를 사용하겠냐는 질문에 **yes**를 선택하십시요.
4. 두번째 클러스터 시스템의 System V init 디렉토리에 있는 cluster stop을 사용하여, 클러스터 소프트웨어를 정지시키십시요. 이때는 어떤 서비스도 활성화 중이지 않습니다.
5. **cluconfig**를 처음 업데이트된 시스템에서 실행합니다. 현재 클러스터 데이터베이스를 사용하겠냐는 질문에, **yes**를 선택하십시요. 클러스터 설정은 현재 설정에 알맞는 기본 설정 변수들을 보일 것입니다. 만일 수정할 필요가 없다면, [Enter]를 눌러서, 현재의 변수들을 지정합니다.
6. 업데이트된 첫 시스템의 System V의 init 디렉토리에 있는 cluster start 명령어를 사용해 클러스터 시스템을 시작합니다. 이를 통해 서비스들이 활성화 될 수도 있습니다. 예를 들어:

```
/sbin/service cluster restart
```

7. 업데이트 하려는 두번째 클러스터 시스템에 다음과 같은 명령어를 통해서 최신 프로그램으로 설치합니다:
```
rpm --upgrade clumanager-x.rpm
```
*x*를 현재 최신인 **Red Hat 클러스터 관리자**의 버전 번호로 바꿉니다.

8. 두번째 업데이트된 클러스터 시스템에 `/sbin/cluconfig --init=raw_file`를 실행 하십시요. 여기서 *raw_file*은 기본 quorum 파티션을 지정합니다. 이 스크립트는 첫 클러스터 시스템에 지정되어 있는 내용을 기본으로 사용합니다. 예를 들어:
```
cluconfig --init=/dev/raw/raw1
```

9. 두번째 시스템에 있는 System V `init` 디렉토리에 있는 `cluster start`명령어를 통해 클러스터 소프트웨어를 시작합니다. 예를 들어:
```
/sbin/service cluster start
```

8.8. 클러스터 데이터베이스 재시작하기

`cluadmin`을 시작 후, `cluster reload` 명령어를 사용하여, 클러스터의 데이터베이스를 다시 불러들입니다. 예를 들어:

```
cluadmin> cluster reload
```

8.9. 클러스터 이름 수정하기

`cluadmin`를 시작한 후, `cluster name` *cluster_name*를 이용해 클러스터의 이름을 지정합니다. 클러스터 이름은 `clustat`에서 사용됩니다. 예를 들어:

```
cluadmin> cluster name Accounting Team Fileserver
Accounting Team Fileserver
```

8.10. 클러스터 재초기화하기

아주 가끔, 클러스터 시스템, 서비스, 데이터베이스를 재초기화하려 할 때가 있습니다. 클러스터를 재초기화하기 전에 필히 클러스터 데이터베이스를 백업하십시요. 8.5 절를 참조하십시요.

클러스터의 재초기화를 완벽히 하려면, 다음과 같이 하십시요:

1. 현재 실행중인 클러스터 서비스를 비활성화 시키십시요.
2. 모든 클러스터 시스템의 클러스터 데몬들을 System V `init`디렉토리에 있는 `cluster stop command`를 사용해 정지시킵니다. 예를 들어:
```
/sbin/service cluster stop
```
3. 클러스터 소프트웨어를 설치합니다. 3.1 절를 참조하십시요.
4. 한 클러스터 시스템에, **cluconfig**를 실행합니다. 현재 클러스터 데이터베이스를 사용하겠냐는 질문에 대한 대답으로 **no**를 선택합니다. 이것은 quorum 파티션에 있는 모든 클러스터 데이터베이스를 삭제할 것입니다.
5. **cluconfig**를 끝낸후, 계속 프로그램을 따라 다른 시스템에서도 `cluconfig`를 실행합니다. 예를 들어:
```
/sbin/cluconfig --init=/dev/raw/raw1
```
6. System V의 `init` 디렉토리에 있는 `cluster start`를 사용해서 클러스터 데몬들을 시작합니다. 예를 들어:
```
/sbin/service cluster start
```

8.11. 클러스터 소프트웨어 비활성화 시키기

가끔은 구성원상에 클러스터 소프트웨어를 비활성화 시켜야 할때가 있습니다. 예를 들어, 만일, 클러스터 시스템에 하드웨어 문제가 생기거나, 관리자가 시스템을 재시작해야 할경우, 그러나 시스템이 클러스터에 연결하지 않아야 할 경우에 비활성화 시켜야 합니다.

클러스터 시스템이 클러스터에 연결하지 않고 시작하려면, `/sbin/chkconfig`를 사용하면 됩니다. 예를 들어:

```
/sbin/chkconfig --del cluster
```

클러스터에 시스템이 다시 연결하는 것을 원한다면, 다음과 같은 명령어를 사용하면 됩니다:

```
/sbin/chkconfig --add cluster
```

다음으로 컴퓨터를 재부팅하시거나 System V `init` 디렉토리에 있는 `cluster start` 명령을 실행하시기 바랍니다. 예를 들어:

```
/sbin/service cluster start
```

8.12. 클러스터의 문제를 진단하고 해결하기

클러스터의 문제를 제대로 진단하기 위해서는 이벤트 기록이 활성화되어야 합니다. 디볼어, 클러스터에 문제가 생길 경우, 클러스터 데몬들의 기록을 **debug** 수준으로 지정하십시요. 이것을 통해 보다 자세한 메세지들을 얻을 수 있으며, 문제를 해결하는데 도움이 될 수 있습니다. 일단 문제가 해결됐다면, debug 수준을 **info**로 변경하여 로그 메시지 파일이 필요 이상으로 커지는 것을 방지하시기 바랍니다.

만일 `cluadmin`를 실행하는데 (예, 서비스를 활성화 시키는데) 문제가 생긴다면, **clusvcmgrd**의 보안 수준을 **debug**로 설정하십시오. 이것을 통해 `cluadmin` 실행하면서 생기는 문제에 대한 메세지들을 볼 수 있습니다. 자세한 내용은 8.6 절을 참조하십시요.

클러스터에 생긴 문제는 표 8-5을 참조하시기 바랍니다.

문제	증상	해결
SCSI 버스가 종료되지 않았을때	SCSI 에러가 로그파일에 나타난다	각 SCSI 버스는 시작과 끝에만 종료되어야 한다. 버스 설정을 확인한 후, 호스트 버스 아답터, RAID 콘트롤러 그리고 저장 기기에 종료를 활성화/비활성화 해야 합니다. 핫 플러그인을 지원하려면, 외장 종료가 SCSI 버스에 있어야 한다. 덧붙여, 어떤 장치도 SCSI 버스에 장착된 스텁이 0.1미터 보다 길지 않아야 합니다. 다른 유형의 SCSI 버스 종료하는 방법에 대한 자세한 정보는 2.4.4 절과 A.3 절을 참조하시기 바랍니다.
SCSI 버스의 길이가 최대길이 보다 길 경우	로그 파일에 SCSI 에러가 나타납니다	A.4 절에 나온 것처럼, 각 SCSI 버스는 길이의 한계를 따라야 합니다. 덧붙여, single-ended 장치가 LVD SCSI 장치에 연결되지 않아야 합니다, 이유는 전체 버스가 single-end 버스로 옮겨질 것이며, 다른 버스보다 더 명확한 길이의 한계가 있기 때문입니다.

문제	증상	해결
SCSI의 지정 번호가 유일하지 않습니다.	로그 파일에 SCSI 에러가 있습니다.	SCSI 버스에 있는 각 장치에는 특유의 지정 번호가 주어져야 합니다. A.5 절에 좀 더 자세한 내용이 있습니다.
완료 전에 SCSI 명령들이 시간완료됩니다	로그 파일에 SCSI 에러가 있습니다	SCSI 버스에서 순서가 있는 경우, 낮은 순서의 장치가 어느 정도 시간동안 잠금되는 경우가 있습니다. 따라서 디스크 같은 낮은 순서의 저장 장치를 시간 완료될 수 있습니다. 일부 작업의 경우 호스트 버스 아답터에 순서가 낮은 SCSI 지정 번호를 지정함으로 이 문제를 피할 수 있습니다. 보다 많은 정보는 A.5 절을 참조하시기 바랍니다.
장착된 quorum 파티션	로그 파일에 quorum 파티션중 하나에 checksum 에러가 있다는 메세지가 있습니다	Quorum 파티션 원 장치가 클러스터 상태 관련 자료만 가지고 있음을 확실히 합니다. 이 quorum 파티션은 클러스터 서비스나 혹은 클러스터 아닌 목적으로 사용될 수 없으며 파일 시스템을 가질 수 없습니다. 자세한 내용은 2.4.4.3 절를 참조 하십시요. 이 메세지들은 quorum 파티션의 기본인 블럭 장치 특별 파일이 잘못되어 클러스터와 관련 없는 이유에 사용되고 있음을 나타낼 수도 있습니다.
서비스 파일 시스템에 문제가 있습니다.	비활성화된 서비스가 활성화되지 않습니다	수동으로 `fsck`을 실행한 후 서비스를 활성화 합니다. 기억해야 할 것은 클러스터 체제에서 기본적으로 `fsck` 명령을 `-p` 옵션과 더불어 자동으로 파일 시스템의 문제를 풀기 위해 사용합니다. 다른 파일 시스템 관련 에러들은 다른 옵션을 지정한 후에 사용할 수 있습니다.
Quorum 파티션이 제대로 설정되지 않습니다.	로그파일의 메세지가 quorum 파티션을 사용할 수 없다고 합니다	Quorum 파티션을 사용할 수 있는지 없는지 여부를 `cludiskutil -t` 를 통해 확인합니다. 만일 명령이 된다면, 모든 클러스터 시스템에 `cludiskutil -p`를 실행 합니다. 만일 시스템들의 결과 출력이 다를 시에는 quorum 파티션이 서로 다른 장치의 다른 지점을 지시하고 있습니다. 원 장치가 존재하는지 확인하고, `/etc/sysconfig/rawdevices` 파일이 확실한지 확인합니다. 2.4.4.3 절에서 좀더 자세한 내용을 확인 하십시요. 이 메세지는 또한 **cluconfig** 중 quorum 파티션의 초기화를 할때, **yes**를 선택하지 않은 것을 의미합니다. 이것을 고치려면, 다시 프로그램을 실행하십시요.

문제	증상	해결
클러스터 서비스 실행 실패	이 메세지는 콘솔이나 로그 파일에 서비스의 실행에 실패했음을 알립니다.	서비스의 실행의 실패에는 많은 이유가 있을 수 있습니다 (예를 들어, 서비스의 멈춤과 시작). 문제의 이유를 알기 위해서는, 클러스터 데몬의 심각성 레벨을 **debug**로 내려 좀 더 자세한 메세지를 얻습니다. 그 후, 실행을 시도해 보며, 로그 파일을 확인 합니다. 좀 더 자세한 내용은 8.6 절을 확인합니다.
파일 시스템을 마운트하지 못해 클러스터 서비스 멈추기에 실패합니다	실행에 실패했다는 메세지가 콘솔이나 로그파일에 나타납니다	`fuser`와 `ps` 명령어를 통해, 파일 시스템을 사용 중인 프로세스들을 확인합니다. `kill` 명령어를 사용해 프로세스를 중지시킵니다. `lsof -t` *`file_system`* 명령어를 통해 파일 시스템을 사용중인 프로세스들의 번호를 얻습니다. 만일 필요하다면, 그 결과를 `kill` 명령어와 연결할 수도 있습니다. 이 문제를 피하려면, 클러스터와 관련된 프로세스들만 공유 저장 데이타를 사용할 수 있게 합니다. 덧붙여서, 서비스를 수정하고 파일 시스템을 강제로 마운트 해제할 수 있도록 합니다. 이것을 통해 만일 프로그램이나 사용자가 파일 시스템을 사용중이더라도 마운트 해제할 수 있습니다.
클러스터 데이터베이스에 잘못된 항목이 있습니다	클러스터 동작에 문제가 있습니다	서비스의 설정을 확인하며, 수정하는데 `cluadmin`이 사용될 수 있습니다. 덧붙여서, `cluadmin`은 클러스터의 변수를 수정하는데도 사용되기도 합니다.
클러스터 데이터베이스나 `/etc/hosts` 파일에 잘못된 이더넷 heartbeat 내용이 있습니다	인터페이스가 사용중이면서도 클러스터 상태에는 이더넷 heartbeat 채널이 **OFFLINE**인 것으로 표시가 되어있습니다.	8.4 절에서 명시된 것처럼 **cluconfig** 를 사용하여 클러스터 설정을 확인하고 수정합니다. 덧붙에서, `ping` 명령어를 사용해서, 클러스터에 사용중인 모든 네트워크 인터페이스에 패켓을 보내보도록 합니다.
전원 스위치와 연결된 케이블이 빠져있습니다	전원 스위치의 상태가 **Timeout**입니다	벙럴 케이블의 연결을 확인합니다
전원 스위치의 벙럴 포트가 클러스터 데이터베이스에 잘못 기재되어 있습니다	전원 스위치의 상태가 문제를 나타내고 있습니다	8.4 절에서 명시된 것처럼 **cluconfig** 를 사용하여 클러스터 설정을 확인하고 수정하여 문제를 해결합니다.
Heartbeat 채널의 문제	Heartbeat 채널의 상태가 **OFFLINE** 입니다	8.4 절에서 명시된 것처럼 **cluconfig** 를 사용하여 클러스터 설정을 확인하고 수정하여 문제를 풉니다. 각 hearbeat 채널에 올바른 유형의 케이블이 사용되었는지 확인합니다. `ping` 명령어를 각 클러스터 시스템 네트워크 인터페이스의 이더넷 heartbeat 채널에 실행합니다.

표 8-5. 클러스터의 문제 진단과 개선

9장 .

Red Hat 클러스터 관리자의 GUI를 사용하여 설정하기

Red Hat 클러스터 관리자에는 관리자가 손쉽게 클러스터의 상태를 관리할수 있는 그래픽 인터페이스를 포함하고 있습니다. 하지만, GUI를 통해서 설정을 바꾸거나 클러스터를 관리할 수는 없습니다.

9.1. JRE 설정하기

Red Hat 클러스터 관리자의 GUI는 웹을 기반한 원격 관리를 위해 클러스터의 구성원에서나, 클러스터의 구성원이 아닌 컴퓨터에서 실행될수 있습니다. GUI 자체는 웹브라우저에서도 실행이 가능하도록 자바 애플렛으로 구성되어 있습니다. 이 이유 때문에, GUI가 실행될 시스템에 자바 런타임 환경 (JRE)이 설치, 브라우저의 플러그-인으로 설정 되어야 합니다. 클러스터 관리 프로그램 GUI는 IBM의 JRE나 Sun의 JRE를 사용해서 실행할 수 있습니다.

경고

Red Hat Linux 어드밴스 서버에는 IBM JRE가 기본으로 설치되어 있습니다. Sun의 JRE를 설치하시고 사용하는 것은 지원되지 않습니다. 9.1.2 절 에 제공된 자료는 Sun의 JRE를 사용하시고자 하는 사용자에게 편의를 제공하고자 하는 목적으로 제공된 것입니다.

9.1.1. IBM JRE 설정하기

IBM JRE는 `IBMJava2-JRE-1.3.<version>` 을 통해 자동적으로 설치되어 있습니다. (이곳에서 *<version>* 는 현 IBM JRE의 버전 번호입니다.) 이 패키지는 /opt/IBMJava2-13/에 JRE를 설치할 것입니다.

RPM 설치 중에, 자동적으로 **모질라** 웹 브라우져에 필요한 플러그-인을 설치합니다.

넷스케이프 네비게이터 버전 4.x에서 IBM JRE를 활성화 시키시려면, JRE에 같이 첨부되어 있는 문서를 참고하시기 바랍니다. 예를 들어, IBM JRE v.1.3.1-3의 `/opt/IBMJava2-131/docs/README-EN.JRE.HTML`에 명시되어 있는 다음과 같은 명령어를 사용하시기 바랍니다:

```
cd /usr/lib/netscape/plugins
ln -s /opt/IBMJava2-131/jre/bin/javaplugin.so
```

9.1.2. Sun JRE 설정하기

만일 클러스터 구성원이 아닌 서버에 클러스터 GUI를 설치해야 할경우, JRE를 다운로드 받으신 후 설치를 해야할 것입니다. JRE는 Sun의 java.sun.com 사이트에서 구하실 수 있습니다. 예를 들어 이 책이 출판될 시에 해당 웹페이지는 http://java.sun.com/j2se/1.3/jre/download-linux.html 이었습니다.

JRE를 받으신 후, 다운로드 받으신 프로그램을 실행하십시요 (예를 들어, `j2re-1_3_1_02-linux-i386-rpm.bin`) 그리고 License agreement에 동의해 주십시요. 이것을 통해, RPM으로 설치할수 있는 JRE의 RPM, `jre-1.3.1_02.i386.rpm`이 생성될 것입니다.

JRE를 설치한 후, GUI 애플릿을 사용하시고자 하는 브라우져에 자바 지원을 활성화 시키십시요. 자바 지원을 활성화 시키는 방법은 각 브라우져와 브라우져 버전에 따라 다릅니다. 자바 플러그 인을 찾으신 JRE 다운로드 페이지에서 더 자세한 정보를 얻으십시요.

예를 들어, 버전 4의 **Netscape Navigator/Communicator** 에서 자바를 사용하시려면 다음 같은 내용을 `~/.bash_profile` 파일 안에 더하십시요:

```
export NPX_PLUGIN_PATH=/usr/java/jre1.3.1_02/plugin/i386/ns4:/usr/lib/netscape/plugins
```

특정 디렉토리 경로는 다를 수 있습니다. 또한 주의하실것은, 버전 6의 Netscape Communicator에서는 JRE의 설치 방법이 다르다는 것입니다.

다음 예제는 모질라 브라우저에서, 자바 플러그 인을 활성화 시키는데 필요한 설정 방법입니다:

```
ln -s /usr/java/jre1.3.1_02/plugin/i386/ns600/libjavaplugin_oji.so \ /usr/lib/mozilla/plugins/
```

9.2. 클러스터 모니터링 매개 변수 설정하기

cluconfig이 클러스터를 설정하기 위해 사용될때, **Cluster Manager**의 GUI와 관련이 있는 설정 자료들이 나타날 것입니다.

첫 GUI는 클러스터의 별칭을 설정할 것인지 아닌지를 묻는 창이 나타날 것입니다. 예를 들어:

```
Enter IP address for cluster alias [NONE]: 172.16.33.128
```

클러스터 별칭은 어떤 클러스터 멤버들에게서든 활성화될 수 있는 유동 IP 주소들로 이루어져 있습니다. 이것을 위해 예를 들어, IP주소가 172.16.33.128로 설정되었습니다. 클러스터 구성원을 모니터하기 위해서는 브라우저 상에서 이 IP 주소(혹은 관련된 호스트 이름을 사용하는 것이) 용이합니다. 만일 별칭을 만들지 않을 경우, GUI를 사용하여 클러스터의 상태를 알아 보기 위해서는 각각 클러스터 멤버에게 필요한 주소를 주는 것이 필요합니다. 별칭을 사용할 경우 좋은 점은, 클러스터 구성원 중 하나라도 온라인인 경우 GUI가 계속적으로 모니터할 것입니다.

두번째로 **cluconfig**에서 물어볼 GUI와 관련된 변수는 원격 감시를 허용할 것인지 아닌지를 묻는 것입니다. 예를 들어:

```
Do you wish to enable monitoring, both locally and remotely, via \
  the Cluster GUI? yes/no [yes]:
```

이 곳에 **no**를 선택할 경우, 클러스터의 GUI를 완전히 비활성화 합니다.

9.3. 웹 서버 활성화 시키기

Cluster Manager의 GUI 사용을 활성화 시키려면, 모든 클러스터 멤버들이 웹서버를 실행하고 있어야 합니다. 예를 들어, 아파치 웹서버를 사용하려면, HTTP 데몬이 실행되고 있어야 합니다.


주목

Cluster Manager GUI를 사용하실 계획이라면, 7 장에서 설명된 것처럼 고성능 아파치 서비스를 설정하실 수 없습니다. 이렇게 서비스에 제약을 받는 이유는 고성능 아파치 서비스는 웹 서버가 한번에 한 개의 클러스터 구성원에서만 작동되기 때문입니다.

Cluster Manager가 제대로 작동하기 위해서는 클러스터 디렉토리와 관련된 웹 문서들이 설치되어 있는 아파치 문서 루트의 기본 설정인 `/var/www/html`으로 되어 있어야 합니다.

9.4. Red Hat 클러스터 관리자 GUI 시작하기

자바 플러그 인을 설정한후, 브라우저에 적절한 URL을 입력하여 **Cluster Manager** GUI를 시작할수 있습니다. GUI의 URL은 "/Cluster" 다음에 클러스터 멤버의 이름이나 클러스터 별칭을 입력하도록 되어있습니다. 전 예를 따른다면, URL이 **`http://clu2alias/Cluster`**일 것입니다. GUI 애플릿이 시작되면, 스플래시 화면이 오른쪽에 나타나며, 왼쪽에는 Tree view가 보일 것입니다. 클러스터 관리를 시작하려면, **Clusters**를 Tree View (유형별로 보기) 상에서 두번 클릭하고, 그 뒤에 (**cluconfig**에서 설정한데로) 클러스터의 이름들이 보일 것입니다.

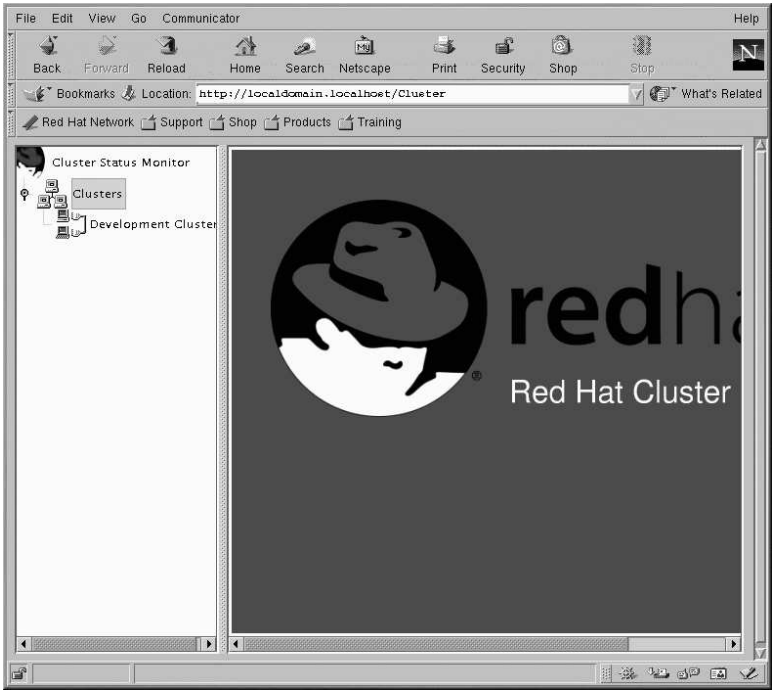

그림 9-1. Red Hat 클러스터 관리자 GUI 스플래시 화면

Tree View상에서 클러스터의 이름을 두번 클릭함으로서, GUI상의 오른쪽에 그림 9-2에 볼수 있는것 처럼 클러스터 관련된 통계가 나타날 것입니다. 이 통계는 각 구성원에서 실행되고 있는 서비스와 heartbeat 채널의 상태 등과 같은 클러스터 구성원의 상태를 보여줍니다.

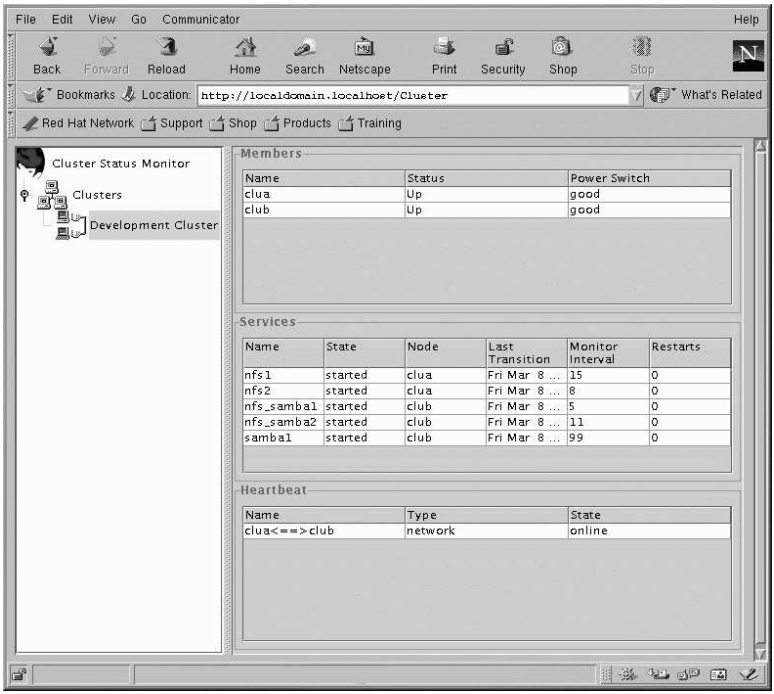

그림 9-2. Red Hat 클러스터 관리자 GUI 화면

기본적으로, 클러스터 통계는 매 5초 마다 갱신됩니다. Tree에 있는 클러스터 이름에 오른쪽 마우스 클릭을 하시면, 갱신되는 시간을 바꿀수 있는 창을 보실 수 있습니다.

9.4.1. 설정 항목 보기

클러스터 모니터링을 시작한 후, 클러스터 상태 항목을 두번 클릭 함으로 자세한 설정 자료를 얻을 수 있습니다. 전의 예를 본다면, **nfs_samba1** 서비스에 두번 클릭하면, 그림 9-3에 서비스에 관련된 자료가 나타날 것입니다:

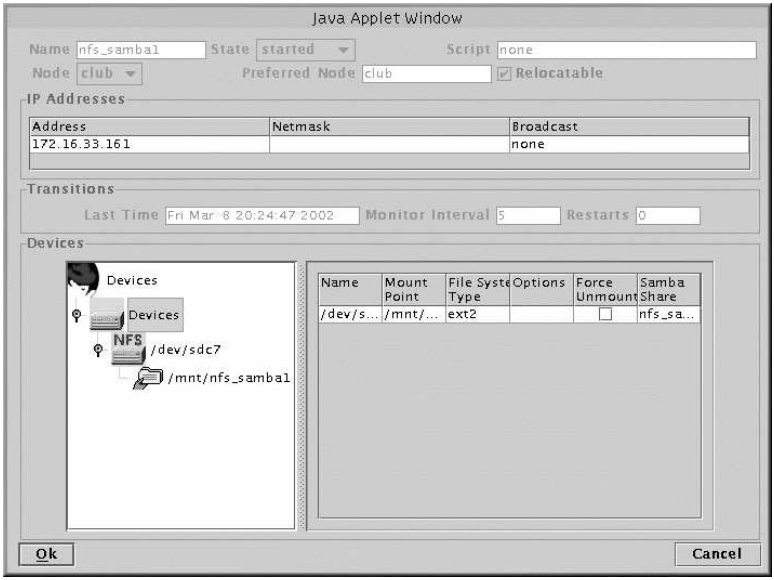

그림 9-3. Red Hat 클러스터 관리자 GUI 설정 자료 화면

그림 9-3에서 각 장치 번수에 클릭하시면 보다 자세한 장치 정보를 보실 수 있습니다.

클러스터 서비스에 관련된 자세한 설정 정보를 얻을 수 있을 뿐만 아니라 또한 GUI 상에 적절한 곳을 두번 클릭함으로서 개별 클러스터 구성원과 heartbeat 채널의 설정을 보실 수도 있습니다.

부록 A.

보조 하드웨어 정보

다음에서는 클러스터 하드웨어 설정을 도울 수 있는 정보가 있습니다. 몇 경우에는, 정보가 제조사에 따라 다를 수도 있습니다.

A.1. 전원 스위치 설정하기

A.1.1. RPS-10 전원 스위치 설정하기

만일 클러스터의 일부로 RPS-10 기종이 사용되었다면, 다음을 확실히 해주십시요:

- 모든 전원 스위치의 로타리 주소를 0으로 해주십시요. 스위치가 제대로 위치되어 있으며, 설정 중이지 않게 하십시요.
- 모든 전원 스위치의 4개 설정 스위치를 다음과 같이 옮기십시요:

스위치	기능	Up Position	Down Position
1	Data rate		X
2	Toggle delay		X
3	Power up default	X	
4	Unused		X

표 A-1. RPS-10 전원 스위치 설정하기

- `/etc/cluster.conf` 파일에 지정된 병렬 포트 장치 특벌 파일 (예를 들어, `/dev/ttyS1`)에 연결된 병렬 포트에 전원 스위치의 병렬 케이블이 연결되어 있는지 확인하십시요.
- 각 클러스터 시스템의 전원 케이블을 전원 스위치에 연결합니다.
- 다른 클러스터 시스템에 전원을 공급하는 전원 스위치의 병렬 포트와 각 클러스터 시스템은 Null 모뎀 케이블로 연결합니다.

그림 A-1 은 RPS-10 계열 전원 스위치의 설정의 예를 보여주고 있습니다.

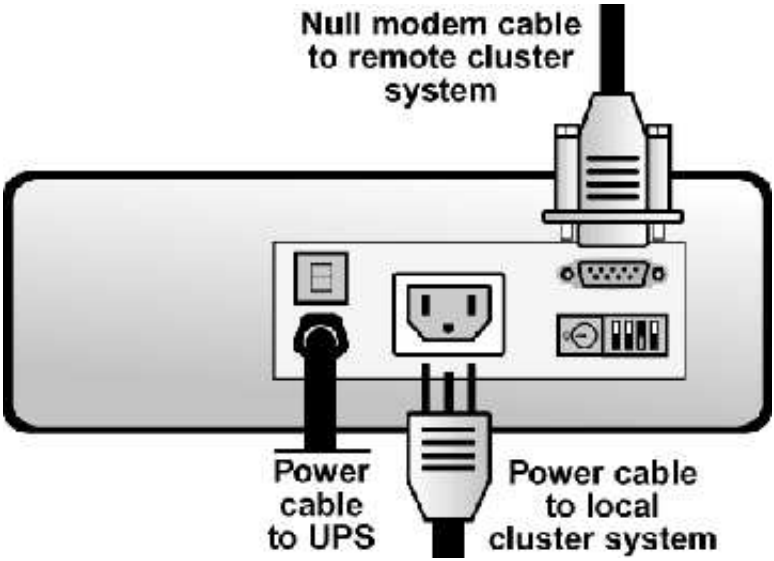

그림 A-1. RPS-10 전원 스위치 하드웨어 설정

RPS-10에 대한 더 자세한 내용은 제조사에서 공급한 문서를 참조하십시요. 기억하실 것은 제조사에서 공급하는 내용의 극히 일부일 뿐입니다.

A.1.2. WTI NPS 전원 스위치 설정하기

WTI NPS-115 와 NPS-230은 네트워크 기반의 장치입니다. 중요한 것은 이것은 네트워크 연결이 가능한 전원 스트립으로 연결될 장치 하나 하나에 전원 사이클링이 가능합니다. 한 클러스터 안에서는 NPS 하나가 필요합니다 (클러스터 구성원 하나에 스위치 하나가 요구되는 RPS-10모델과는 다릅니다).

주의하실 것은 클러스터 소프트웨어를 통해 각 클러스터 구성원 시스템이 맞는 NPS 전원 스위치에 연결되었는지 확인할 방법이 없기 때문에 설정을 확인하시기 바랍니다. 만일 문제가 생길 경우, 클러스터 소프트웨어가 전원 사이클의 성공 여부를 잘못 판단할 수 있습니다.

NPS 스위치를 설정하려면 다음과 같은 설정을 따라야 합니다.

전원 스위치를 설정시:

- **System Password**를 정합니다 (**General Parameters** 에 있습니다). 주의: 이 비밀 번호는 일반 텍스트로 클러스터 설정 파일에 저장되기에, 시스템의 비밀번호와 다른 비밀 번호를 선택하십시요. (하지만 `/etc/cluster.conf`은 root만 읽을 수 있는 파일 권한을 가지고 있습니다.)
- **Plug Parameters**에 비밀번호를 정하지 마십시요.
- 플러그 번수에 시스템 이름을 지정해 주십시요, (예를 들어, *`clu1`* 를 plug 1에, *`clu2`* 를 plug 2에 — 이것을 클러스터 이름이라고 가정한 것입니다).

전원 스위치의 번수를 **cluconfig**를 통해 지정합니다:

- **WTI_NPS**의 스위치 유형을 지정합니다.

- NPS 스위치에 지정한 비밀번호를 입력합니다 (첫 단계를 보십시요).
- 플러그/포트 번호를 물을때, 세번째 단계와 같은 이름을 적어 줍니다.

주의

브로드캐스트나 멀티-캐스트 패킷이 많은 네트워크 상에서는 NPS 전원 스위치가 응답을 안할 수도 있습니다. 이런 상황에서는 전원 스위치를 개별 서브넷으로 옮기는 것이 필요할 수도 있습니다.

NPS-115 전원 스위치는 클러스터 구성원의 전원 사이클과 이중 전원 지원을 할 수 있습니다. NPS-115는 2 그룹의 전원 공급원을 가지고 있으며, 각 그룹은 독립적으로 전원 공급을 하며, 4개의 플러그를 가지고 있습니다. 각 NPS-115의 전원 플러그는 다른 전원 공급원에 연결이 됩니다 (추정하기로는 다른 UPS를 사용할 것입니다). 이중 전원 지원을 가진 클러스터 구성원은 각각의 전원 코드를 서로 다른 공급원에 연결하십시요. 그후, NPS-115, 지정된 포트를 설정할때, 간단하게 각 공급원에 클러스터 구성원과 알맞은 이름을 지정하십시요. 예를 들어, 클러스터 구성원이 *clu3* 과 *clu4*이 있을때, *clu3*이 공급원 1과 5, 그리고 *clu4*은 공급원 2와 6에 연결되었을 때:

```
Plug | Name     | Status | Boot Delay | Password    | Default |
-----+----------------+---------+------------+-----------------+---------+

 1  | clu3     |  ON  | 5 sec | (undefined)   |  ON  |
 2  | clu4     |  ON  | 5 sec | (undefined)   |  ON  |
 3  | (undefined)  |  ON  | 5 sec | (undefined)   |  ON  |
 4  | (undefined)  |  ON  | 5 sec | (undefined)   |  ON  |
 5  | clu3     |  ON  | 5 sec | (undefined)   |  ON  |
 6  | clu4     |  ON  | 5 sec | (undefined)   |  ON  |
 7  | (undefined)  |  ON  | 5 sec | (undefined)   |  ON  |
 8  | (undefined)  |  ON  | 5 sec | (undefined)   |  ON  |
-----+----------------+---------+------------+-----------------+---------+
```

여러 공급원에 같은 이름을 지정함으로, 전원 사이클 명령에, 같은 이름을 가지고 있는 공급원이 전원 사이클 될것입니다. 이와 같이, 이중 전원 지원을 가지고 있는 클러스터 구성원은 전원 사이클 될 수 있습니다. 이런 이중 설정에서, **cluconfig**에서 사용되는 번수는 위에 나타난 보통 설정과 동일합니다.

A.1.3. Baytech 전원 스위치 설정하기

다음은 RPC-3 와 RPC-5 전원 스위치에 해당되는 것입니다.

Baytech 전원 스위치는 네트워크 기반의 장치입니다. 중요한 것은 이것은 네트워크 연결이 가능한 전원 스트립으로 연결될 장치 하나 하나에 전원 사이클링이 가능합니다. 한 클러스터 안에서는 NPS 하나가 필요합니다 (클러스터 구성원 하나에 스위치 하나가 요구되는 RPS-10 모델과는 다릅니다).

주의할 것은 클러스터 소프트웨어를 통해 각 클러스터 구성원 시스템이 맞는 Baytech 전원 스위치에 연결되었는지 확인할 방법이 없기 때문에 설정을 확인하시기 바랍니다. 만일 문제가 생길경우, 클러스터 소프트웨어가 전원 사이클의 성공 여부를 잘못 판단할 수 있습니다.

주의

제조사에서 출시된것 같이, Baytech 스위치의 공급원은 모두 꺼져 있습니다. 클러스터 구성원이 플러그 된 공급원에 전원을 넣으려면, Baytech의 설정 메뉴를 메인 메뉴에서 시작하고, **Outlet Control**을 선택하십시요. 여기서 부터, 개개의 공급원을 사용할 수 있습니다, 예를 들어, **on 1**, **on 2**, etc.

Baytech 스위치를 설정할때는, 다음과 같은 단계를 따라야 합니다.

Baytech 전원 스위치를 설정시:

1. 빙렬 연결을 사용할때, IP 주소 관련된 번수를 지정합니다.
2. **Access** 안에 => **Network access** 메뉴안에 있는, **Prompt for user name**와 **Prompt for password**가 활성화 되었는지 확인합니다.
3. 사용자 이름과 비밀번호를 **Manage Users** 메뉴에서 입력하던지, 기본 "admin" 계정과 지정된 비밀번호를 사용합니다. 주의: 이 비밀번호는, 불수 있는 텍스트로 클러스터 설정 파일 안에 저장됩니다, 그렇기에, 시스템의 비밀번호와 다른 비밀번호를 선택해 주십시요 (`/etc/cluster.conf` 파일의 권한은 root 만 읽을 수 있지만, 주의 하십시요).
4. 시스템의 이름을 공급원과 맞추려면, **Configuration** 메뉴로 가셔서, **Outlets** 메뉴, 그리고 마지막으로 **Name Outlets**을 선택하십시요 (예를 들어, *`clu1`* 는 outlet 1로, *`clu2`* 는 outlet 2 로 — 이것들이 클러스터의 구성원의 이름이라는 가정하에서 입니다).

전원 스위치의 번수를 지정하기 위해 **cluconfig**를 실행시:

- 스위치의 유형을 **BAYTECH**로 지정합니다.
- Baytech 스위치에 사용자명과 비밀번호를 지정합니다 (위의 세번째 단계를 참조하십시요).
- 플러그 혹은 포트 번호를 요구하면, 위의 네번째 단계에서 지정한 이름을 적어 주십시요.

다음은 Baytech 스위치를 설정할 때, 나오는 예제 화면 입니다, 위에서 사용한 *`clu1`*와 *`clu2`*를 클러스터 이름으로 사용하였습니다.

```
Outlet Operation Configuration Menu
 Enter request, CR to exit.

  1)...Outlet Status Display: enabled
  2)...Command Confirmation : enabled
  3)...Current Alarm Level (amps): 4.1
  4)...Name Outlets
  5)...Outlet Power-up Delay
  6)...Display Outlet Users
 Enter request>4

 Enter number of outlet to name, CR to exit.
  1)...clu1
  2)...clu2
  3)...Outlet 3
  4)...Outlet 4
  5)...Outlet 5
  6)...Outlet 6
  7)...Outlet 7
  8)...Outlet 8
```

A.1.4. Watchdog 전원 스위치 설정하기

클러스터의 데이타의 무결성을 위해 Watchdog timer (감시 타이머)를 사용하는 예가 2.1.3 절에 나와 있습니다. 그곳에 나온 것 처럼, Watchdog timer에는 2가지 종류가 있습니다: 하드웨어 기반과 소프트웨어 기반.

다음은 클러스터 하드웨어 설정을 통해 watchdog timer를 설정하기 위해 필요한 단계를 자세히 설명하고 있습니다.

무슨 유형의 watchdog timer가 사용되었든지, watchdog timer에 알맞는 특별 장치 파일을 만드셔야 합니다. 특별 장치 파일을 만드시려면 다음과 같이 하십시요:

```
# cd /dev
# ./MAKEDEV watchdog
```

cluconfig를 실행할때, 전원 스위치의 유형을 질문할 것입니다, 어떤 유형의 Watchdog timer이든 간에 SW_WATCHDOG를 지정해 주시면 됩니다.

A.1.4.1. 소프트웨어 Watchdog Timer 설정하기

어떤 특정 하드웨어가 필요하지 않기에 어떤 클러스터 시스템도 소프트웨어 watchdog 타이머를 데이타 무결성을 위해 사용할 수 있습니다. 만일 **cluconfig** 에서 전원 스위치의 유형을 **SW_WATCHDOG**로 지정하였다면, 클러스터 소프트웨어에서, 자동적으로 커널 모듈인, `softdog` 을 준비합니다.

만일 클러스터가 소프트웨어 watchdog timer를 사용하도록 설정이 되어있다면, 클러스터 quorum 데몬 (`cluquorumd`)이 일정 시간마다, 시간을 초기화 시킬 것입니다. 만일 `cluquorumd`이 시간을 초기화 시키는데 실패한다면, 실패한 클러스터 구성원은 자신을 재시작할 것입니다.

소프트웨어 watchdog timer를 사용할 때는, 소프트웨어 watchdog 프로그램이 제대로 시작하지 않으리라는 작은 위험 부담이 있습니다. 이런 경우에는, 다른 클러스터 구성원들이 멈춘 클러스터의 서비스들을 시작할 수 있으며, 이로 인해, 데이타에 문제가 생길 수 있습니다. 이런 문제가 생길 확률을 줄이려면, 소프트웨어 watchdog timer를 사용할때는, 관리자가 NMI watchdog timer를 함께 설정하는 것이 좋습니다.

A.1.4.2. NMI Watchdog timer를 활성화 시키기

만일 소프트웨어 watchdog timer를 데이타 무결성을 위해 사용한다면, Non-Maskable Interrupt (NMI) watchdog timer를 활성화 시키는 것이 더 확실한 무결성을 보장할 수 있습니다. NMI watchdog timer는 데이타 무결성의 확률을 더 높여줍니다. NMI watchdog timer는 인터럽트가 막혔을 때, 시스템을 재시작 시키는 소프트웨어 watchdog timer와는 좀 더 다른 방향을 취하고 있습니다. 이 NMI watchdog 는 소프트웨어 watchdog timer와 병행해서 사용이 가능합니다.

클러스터 Quorum 데몬 (`cluquorumd`)에 의해 재실행 되는, 소프트웨어 watchdog timer와는 다르게, NMI watchdog timer는 시스템 인터럽트를 계산합니다. 보통, 건강한 시스템은 일초에 몇백개의 장치와 timer들의 인터럽트를 받아야 합니다. 만일 5초 간격으로 인터럽트가 없는 경우, 시스템이 멈춘 것으로, NMI watchdog timer가 정지해, 시스템의 재시작을 초기화할 것입니다.

보다 나은 데이타 무결성은 클러스터 quorum 데몬의 상태 점검을 소프트웨어 watchdog timer에 NMI watchdog를 사용한 저레벨 시스템의 상태 점검을 통해 이루어 질 수 있습니다.

NMI watchdog timer이 제대로 작동하기 위해서는 주 시스템 보드에 APIC 칩이 포함 되어야 합니다. 최근에 나온 거의 모든 시스템들은 APIC 칩을 가지고 있습니다. 보통, 인텔 기반의 SMP 시스템과 인텔 기반의 SMP 시스템 보드에 프로세스 하나의 (2+ cpu 슬롯/소켓, 그러나 하나의 CPU) 보드들은 NMI watchdog를 지원합니다.

주의

인텔 기반의 시스템 보드외에도 NMI Watchdog를 지원하는 다른 서버들도 있을 수 있습니다. 불행하게도, 직접 사용하고 테스트 하는 방법 외에는 쉽게 지원을 하는지 않하는 지를 알 수 있는 방법이 없습니다.

NMI watchdog는 지원하는 시스템에서 커널에 *`nmi_watchdog=1`*과 같은 옵션을 전해 주는 방법으로 활성화 시킬 수 있습니다. 다음은 `/etc/grub.conf`의 예입니다:

```
#
# grub.conf
#
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title HA Test Kernel (2.4.9-10smp)
    root (hd0,0)
    # This is the kernel's command line.
    kernel /vmlinuz-2.4.9-10smp ro root=/dev/hda2 nmi_watchdog=1
```

```
# end of grub.conf
```

grub 대신 lilo를 사용하는 시스템에서는 `/etc/lilo.conf` 파일의 "append" 부분에 **`nmi_watchdog=1`** 라인을 추가하십시오. 예:

```
#
# lilo.conf
#
prompt
timeout=50
default=linux
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
lba32

image=/boot/vmlinuz-2.4.9-10smp
    label=linux
    read-only
    root=/dev/hda2
    append="nmi_watchdog=1"

# end of lilo.conf
```

서버가 NMI watchdog를 지원하는지 알기 위해서는, 먼저 "nmi_watchdog=1"를 커널에 위에서 얘기한바와 같이 정해주려 합니다. 만일 시스템이 시작하면, root로 접속하신 후에 다음과 같이 입력하십시요:

```
cat /proc/interrupts
```

출력이 다음과 비슷할 것입니다:

```
     CPU0
 0:  5623100    XT-PIC timer
 1:    13    XT-PIC keyboard
 2:    0    XT-PIC cascade
 7:    0    XT-PIC usb-ohci
 8:    1    XT-PIC rtc
 9:  794332    XT-PIC aic7xxx, aic7xxx
10:  569498    XT-PIC eth0
12:    24    XT-PIC PS/2 Mouse
14:    0    XT-PIC ide0
NMI:  5620998
LOC:  5623358
ERR:    0
MIS:    0
```

관련있는 부분은 왼쪽에 보이는 `NMI`란 줄 입니다. 만일 NMI 값 (중간 줄에)이 0이 아닐 경우, 서버는 NMI Watchdog를 지원합니다.

만일 이 방법이 맞지 않으면, 예를 들어, `NMI`의 값이 0이면, 커널에 ***`nmi_watchdog=1`*** 대신해서 ***`nmi_watchdog=2`***를 전해주려 한후, 위에서 얘기한 바를 따릅니다. 다시한번, /proc/interrupts를 시스템이 시작한 후에 확인하십시요. 만일 `NMI`이 0인 경우, 시스템이 NMI watchdog timer를 지원하지 않습니다.

A.1.4.3. 하드웨어 Watchdog Timer를 설정하기

커널상에서 여러 유형의 하드웨어 watchdog timer의 드라이버를 지원합니다. 그 중의 몇은 직접 시스템 보드에 있으며, 다른 것들은 PCI 카드와 같은 유형이 있습니다. 하드웨어 기반의 watchdog timer는 클러스터 시스템내에서, 시스템 프로세서와 무관하게 실행되며 시스템이 멈추었을때 완벽하게 사용될 수 있기에, 아주 좋은 데이타 무결성을 보장합니다.

저수준의 하드웨어 watchdog의 부품들의 공통성이 부족하기에, 전체적으로, 어느 시스템에 그런 부품이 있는지 없는지 확인하는 방법을 설명하기 힘듭니다. 많은 저레벨 하드웨어 watchdog 부품들은 자신을 표시하지 않습니다.

커널에서는 표 A-2에서 처럼 하드웨어 watchdog 변수들을 지원하고 있습니다:

카드/타이머	드라이버
Acquire SBC	acquirewdt
Advantech SBC	advantechwdt
Intel-810 based TCO WDT	i810-tco
Eurotech CPU-1220/1410 WDT	eurotech
IB700 WDT	ib700
60xx SBC WDT	sbc60xxwdt
W83877F WDT	w83877f
Netwinder W83977AF	wdt977
Industrial Computer WDT500	wdt
Industrial Computer WDT501	wdt
Industrial Computer WDT500PCI	wdt_pci
Industrial Computer WDT501PCI	wdt_pci

표 A-2. 하드웨어 watchdog timer

위에 나온 watchdog timer들 중 어떤 것이든 커널에 설정하려면, 알맞는 내용을 /etc/modules.conf 파일에 입력해야 합니다. 예를 들어, Intel-810 기반의 TC0 WDT를 사용하려면, 다음과 같은 줄이 /etc/modules.conf에 더해져야 합니다:

```
alias watchdog i810-tco
```

A.1.5. 다른 네트워크 전원 스위치들

클러스터 소프트웨어는 여러 종류의 전원 스위치 지원을 포함하고 있습니다. 이 전원 스위치의 모듈 지원은 Mission Critical의 개발자가 Linux-HA 프로젝트의 일부로 시작되었습니다. 시간과 자원의 부족으로 인해 완벽하게 모든 스위치의 유형을 검사하지는 못했습니다. 그렇기에, 전원 스위치의 STONITH 모듈은 사용되지 않습니다. 다른 전원 스위치 모듈의 예는 다음과 같습니다:

- APC 주 스위치: http://www.apc.com

주의

주 스위치는 많은 양의 브로드캐스트나 멀티-캐스트 패킷이 있는 네트워크상에서 문제가 있는 것이 발견 되었습니다. 만일 이런 일이 생길 경우, 전원 스위치를 다른 서브넷으로 옮겨 주십시요.

- APC 병렬 On/Off 스위치 (partAP9211): http://www.apc.com

주의

이 종류의 스위치는 클러스터가 상태를 알아낼 수 없도록 만들어 졌습니다. 그렇기에, 클러스터는 항상 이 스위치가 연결이 되었으며 동작 중이라 생각합니다.

A.1.6. None 유형의 전원 스위치 설정하기

클러스터에 전원 스위치의 기능없이 설정할 수 있습니다. 2.1.3 절에서 설명 한것처럼, 클러스터에 전원 스위치 없이 사용하는 것을 자동 오류 복구의 시나리오에서 데이타 무결성의 문제로 인해 권장하지 않습니다.

클러스터에 어떤 전원 스위치의 기능없이 설정하려면, **cluconfig**에서 전원 스위치의 유형을 물을때, 간단하게 **NONE**를 선택합니다.

주의

시스템이 멈추었을 때 데이타의 무결성의 문제 때문에 전원 스위치 유형을 **NONE**으로 놓고 사용하는 것을 절대 권장하지 않습니다. 만일 클러스터 설정에 하드웨어 전원 스위치를 포함하지 않는다면, 소프트웨어 watchdog 유형을 권장합니다.

A.2. SCSI 버스 설정 필요 조건

SCSI 버스는 몇가지 필요 조건을 갖추어야 적절하게 사용될 수 있습니다. 필요 조건을 맞추지 못할 때에는 클러스터 작동상, 프로그램 작동상 그리고 데이타의 사용에 문제를 초래할 수 있습니다.

다음이 **SCSI 버스 설정 필요 조건** 입니다:

- 버스는 양쪽에서 멈추어야 합니다. 자세한 내용은 A.3 절을 참조하시기 바랍니다.
- 버스는 버스 유형에 따라 정해진 최대 길이 제한을 넘지 말아야 합니다. 내부 연결의 선의 길이도 SCSI 버스의 길이에 포함 되어야 합니다. 자세한 내용은 A.4 절를 참조하십시요.
- 버스와 연결된 모든 장치(호스트 버스 아답터와 디스크들)는 유일한 번호를 가져야 합니다. 자세한 내용은 A.5 절를 참조하십시요.
- 모든 클러스터 시스템의 각 공유 SCSI 장치의 장치 이름은 같아야 합니다. 예를 들어, 한 클러스터 시스템에 공유 장치가 `/dev/sdc`라 명명 되었다면, 다른 클러스터 시스템도 `/dev/sdc` 이어야 합니다. 모든 장치의 이름이 동일하게 하는 하나의 방법은 모든 클러스터 시스템에 같은 하드웨어를 사용하는 것입니다.

시스템의 설정 프로그램을 사용하여, SCSI 번호를 설정하고, 호스트 버스 아답터의 멈춤을 활성화 시킬 수 있습니다. 시스템이 시작될때, 설정 프로그램을 시작하는 방법을 알리는 메세지를 볼 수 있습니다. 예를 들어, 프로그램이 사용자가 [Ctrl]-[A]를 입력하도록 알리며, 특정 작업을 하기 위해 질문을 따르도록 알릴 것입니다. 저장 장치의 닫힘과 RAID 콘트롤러 멈춤을 설정하려면, 제품 생산사의 문서를 참조 하십시요. 또한 A.3 절와 A.5 절를 참조하십시요.

http://www.scsita.org에서 좀 더 자세한 SCSI 버스의 요구 조건을 보실 수 있습니다.

A.3. SCSI 버스 멈춤

SCSI 버스는 두 멈춤 사이의 경로입니다. SCSI 버스에 작은 멈춤이 없는 버스 *stub*으로 연결된 장치 (호스트 버스 아답터, RAID 콘트롤러, 혹은 디스크)의 Stub은 0.1미터보다 짧아야 합니다.

버스는 두 멈춤이 있어야 하며, 버스의 양쪽끝에 위치해야 합니다. 추가 멈춤, 혹은 긴 stub은 버스가 제대로 작동치 않는 이유가 될 수 있습니다. SCSI 버스의 멈춤은 버스의 연결된 장치나 외부 멈춤, 만일 내부 (보드에 장착된) 멈춤이 비활성화 될수 있다면, 이 사용될 수 있습니다.

실험에 의하면, HBA 가 초당 80MB의 속도로 운영된다면, 외부 멈춤이 문제가 있을 수 있습니다.

만일 single-initiator SCSI 버스에서 장치의 제거하시려면, 다음을 따라 주십시요:

- 멈춤이 없는 SCSI 케이블은 현재 사용중인 호스트 버스 아답터나 저장 장치에 연결되면 안됩니다.
- 연결 핀은 SCSI 케이블에서 제거되었을때, 휘거나 전류가 통하는 물건에 닿아서는 안됩니다.
- 호스트 버스 아답터를 single-initiator 버스에서 제거 하려면, SCSI 케이블을 먼저 RAID 콘트롤러에서 분리하시고, 아답터에서 분리하십시요. 이것을 통해 RAID 콘트롤러에 잘못된 입력이 들어가는 것을 방지할 수 있습니다.
- SCSI 케이블이 분리되어 있는 동안, 접지가된 반-정전기 손목대를 사용하여 연결 핀을 전정기에서 보호하며, 케이블의 끝을 다른 물건과의 접촉에서 보호하십시요.
- 현재 SCSI 버스의 동작에 사용되는 장치를 분리하지 마십시요.

아답터의 내부 멈춤을 활성화 혹은 비활성화 시키기 위해서는 시스템 바이오스 프로그램을 사용하시기 바랍니다. 시스템이 시작할때, 프로그램을 어떻게 시작해야 하는지 메세지가 나올 것입니다. 예를 들어, 많은 프로그램은 사용자에게 Ctrl-A를 입력하라 할 것입니다. 그 후 멈춤 설정의 단계를 따르십시요. 이 곳에서 SCSI 번호를 설정할 수 있으며, 또한 필요하다면, SCSI 버스 재설정을 비활성화 시킬 수도 있습니다. 더 자세한 내용은 A.5 절을 참조하시기 바랍니다.

저장 장치와 RAID 콘트롤러의 멈춤을 설정하려면, 제조사의 문서를 참조하시기 바랍니다.

A.4. SCSI 버스 길이

SCSI 버스는 버스의 유형에 따라 길이 제한을 따라야 합니다. 이 제한을 따르지 않는 버스는 작동에 문제가 있을 수 있습니다. SCSI 버스의 길이는 한 쪽 끝의 멈춤에서 다른 멈춤까지를 측정하며, 시스템과 저장 장치안의 케이블도 포함됩니다.

만일 클러스터가 LVD 버스를 지원한다면, single-initiator LVD 버스의 길이 제한은 25미터 입니다. multi-initiator LVD 버스의 길이 제한은 12미터 입니다. SCSI의 기준에 따라, Single-initiator LVD 버스는 2개의 장치만 연결할 수 있는 버스이며, 각 장치는 멈춤에서 0.1미터이내에 있어야 합니다. 모든 다른 버스는 multi-initiator 버스로 정의됩니다.

LVD 버스에 single-ended 장치를 연결하지 마십시요, 그렇지 않으면, 버스가 single-ended 버스로 바뀌게 되며, Differential 버스보다 짧은 길이 제한을 갖게 됩니다.

A.5. SCSI 번호

SCSI 버스의 각 장치는 유일한 SCSI 번호를 가져야 합니다. 장치란 버스 아답터, RAID 콘트롤러 그리고 디스크들을 말합니다.

SCSI 버스에 장착할 수 있는 장치의 수는 버스의 데이타 경로에 따라 달라 집니다. 클러스터가 16bit 데이타 경로를 가진 Wide SCSI 버스를 지원한다면, 최고 16개의 장치를 지원할 수 있습니다. 그렇기에, 이 버스에는 16개의 번호가 주어질 수 있으며, 장치에 지정될 수 있습니다.

덧붙여, SCSI 번호에는 순서가 있습니다. 다음과 같은 순서를 SCSI 번호를 지정하는데 사용하십시요:

```
7 - 6 - 5 - 4 - 3 - 2 - 1 - 0 - 15 - 14 - 13 - 12 - 11 - 10 - 9 - 8
```

위의 순서에는 7이 가장 높으며, 8이 가장 낮게 되어있습니다. 호스트 버스 아답터에 지정된 기본 SCSI 번호는 7입니다. 이유는 아답터들에 높은 순서가 정해지기 때문입니다. RAID 관리 인터페이스를 통해 RAID안의 존재하는 시스템의 logical 단위들에게 번호를 지정할 수 있습니다.

아답터의 SCSI 번호를 수정하려면, 시스템 BIOS 프로그램을 사용하십시요. 시스템이 시작할때, 프로그램을 어떻게 시작하는지 메세지가 나올것입니다. 예를 들어, 사용자가 [Ctrl]-[A]를 입력하도록 나오며, 그 후 SCSI 번호를 설정하는 방법을 따르면 됩니다. 이 곳에서, 아답터의 내부 멈춤을 활성화 혹은 비활성화 시킬수 있으며, 필요하다면, SCSI 버스의 재설정을 비활성화 시킬 수도 있습니다. 더 자세한 내용은 A.3 절을 참조 하시기 바랍니다.

SCSI 버스에 순서를 정해 놓음으로써, 순서가 낮은 장치가 한정된 시간동안 멈추는 상황이 발생할 수 있습니다. 이로 인해 명령이 시간 제한에 인해 멈출 수 있으며, 만일 순서가 낮은 장치, 예를 들어 디스크, 가 순서 때문에, 호스트가 지정한 명령을 실행할 수 없을 경우, 몇몇의 경우, 호스트 버스 아답터에 낮은 순서의 번호를 지정함으로 문제를 풀 수 있습니다.

A.6. 호스트 버스 아답터의 기능과 설정 요구 사항

다음 표는 권장된 SCSI와 Fibre 채널 호스트 버스 아답터에 대해 설명하고 있습니다. 여기에는 아답터의 멈춤과 어떻게 아답터를 single initiator SCSI 버스에서 사용하며, Fibre 채널 내부 연결과 사용할 수 있는지 설명하고 있습니다.

이 표에 나와 있는 제품 장치들은 실험을 거쳤습니다. 그렇지만, 다른 장치들도 클러스터 상에서 제대로 작동 될 수 있습니다. 권장된 호스트 버스 아답터 외에 다른 아답터들도 사용 될 수 있습니다. 이 표에 있는 정보는 장치들에 기능이 있는지 또한 어떤 성격과 어떻게 그 것들을 클러스터 안에서 활성화 시킬 수 있는지에 대해 나와 있습니다.

호스트 버스 아답터	기능	Single-Initiator 설정
Adaptec 2940U2W	Ultra2, wide, LVD. HD68 외부 연결 채널 하나, 2개의 버스 segment. BIOS 프로그램을 사용해 내부 멈춤 설정. 전원이 꺼졌을때 내부 멈춤 비활성.	내부 멈춤을 자동으로 설정 (기본). 내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.
Qlogic QLA1080	Ultra2, wide, LVD VHDCI 와부 연결 채널 하나 BIOS 프로그램을 사용해 내부 멈춤 설정. 전원이 꺼졌을때, 내부 멈춤 비활성화, 점퍼를 통해 멈춤을 사용하도록 만들 수 있음.	내부 멈춤을 자동으로 설정 (기본). 내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.

호스트 버스 아답터	기능	Single-Initiator 설정
Tekram DC-390U2W	Ultra2, wide, LVD HD68 외부 연결 채널 하나, 두개의 세크먼트 만일 내부와 외부 케이블이 세크먼트 (segment)에 연결되어 있다면, 버스 세그먼트의 내부 멈춤은 비활성화 됩니다. 만일 세그먼트에 케이블이 하나만 연결되어 있다면, 내부 멈춤이 활성화됩니다. 전원이 꺼졌을때, 멈춤이 비활성화 될 것입니다.	내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.
Adaptec 29160	Ultra160 HD68 외부 연결 채널 하나, 두개의 segments BIOS 프로그램을 사용해 내부 멈춤을 설정. 전원이 꺼졌을때, 내부 멈춤 비활성화, 점퍼를 통해 멈춤을 사용하도록 만들 수 있음.	내부 멈춤을 자동으로 설정 (기본). 내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.
Adaptec 29160LP	Ultra160 VHDCI 외부 연결 채널 하나 BIOS 프로그램을 사용해 내부 멈춤을 설정. 전원이 꺼졌을때, 내부 멈춤 비활성화, 점퍼를 통해 멈춤을 사용하도록 만들 수 있음.	내부 멈춤을 자동으로 설정 (기본). 내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.
Adaptec 39160 Qlogic QLA12160	Ultra160 두개의 VHDCI 외부 연결 두개의 채널 BIOS 프로그램을 사용해 내부 멈춤을 설정. 원이 꺼졌을때, 내부 멈춤 비활성화, 점퍼를 통해 멈춤을 사용하도록 만들 수 있음.	내부 멈춤을 자동으로 설정 (기본). 내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.
LSI Logic SYM22915	Ultra160 두개의 VHDCI 외부 연결 두개의 채널 BIOS 프로그램을 사용해 내부 멈춤을 설정. 내부 멈춤은 자동으로 설정에 따라, 전원이 꺼졌을 때도 활성화 또는 비활성화 될수 있습니다. 점퍼를 사용하여, 자동 멈춤을 비활성화 할수 있습니다.	내부 멈춤을 자동으로 설정 (기본). 내부 SCSI 연결을 (클러스터 아닌) 저장 장치에 사용.
Intel L440GX+ 마더보드에 장착된 Adaptec AIC-7896 (VA Linux 2200 시리즈에 사용됨)	하나의 Ultra2, wide, LVD port, 하나의 Ultra, wide port. 내부 멈춤이 활성화로 고정 되어 있습니다, 그렇기에 아답터가 다른 끝에 있어야 합니다.	멈춤이 활성으로 고정되어 있기 때문에 아답터를 single-initiator 버스에 사용하기 위해서 따로 설정할 필요가 없습니다.

표 A-3. 호스트 버스 아답터의 기능과 설정 요구 사항

호스트 버스 아답터	기능	Single-Initiator 설정	Multi-Initiator 설정
QLA2200 (minimum driver: QLA2x00 V2.23	Fibre 채널 arbitrated loop 과 fabric 하나의 채널	아답터에서 다중 포트 저장 장치로 point-to-point 연결이 가능합니다. 아답터를 이중 콘트롤러 RAID array 혹은 다중 RAID array로 연결 하려면 Hub이 필요합니다.	FC hub이나 스위치와 사용될 수 있습니다.

표 A-4. QLA2200 기능과 설정 요구 사항

A.7. 자동 오류 복구 시간 설정하기

이 곳에서는 cluqourumd 데몬에 관련된 변수의 사용 방법을 설명하고 있습니다. 이 변수들은 건강한 클러스터 구성원이 멈춘 클러스터에게, 클러스터가 멈추었다고 단정을 내리기 전에 주어지는 시간을 결정합니다. 이 시간이 끝나면, 건강한 클러스터 구성원이 멈춘 클러스터 멤버에 전원 사이클할 것입니다 (전원 스위치에 따라 다름), 그리고 멈춘 구성원에서 작동중이던 서비스를 계속할 것입니다.

자동 오류 복구의 시간을 정하는데 관련된 몇가지의 변수가 있는데 다음과 같습니다:

이름	기본 (초.)	설명
pingTime	2	cluquorumd 데몬이 얼마나 자주 디스크의 상태 정보를 업데이트 하며, 다른 클러스터 구성원의 상태를 읽을지 정합니다.
sameTimeNetdown	7	cluhbd 하트비트 데몬이 다른 클러스터 구성원과 통신되지 않을때 클러스터 구성원이 멈추었다고 결정할때까지 기다리는 시간.
sameTimeNetup	12	cluhbd heartbeat 데몬이 다른 클러스터 구성원과 통신이 가능하지만, 클러스터 구성원에 문제가 있다고 결정을 내릴때까지 기다리는 시간. 이 시간은 sameTimeNetdown 변수 시간보다 커야 합니다.

표 A-5. cluquorumd안의 자동 오류 복구 시간 변수

예를 들어, 클러스터 구성원이 하드웨어에 문제가 있고 작동을 멈추었다고 합시다. 이런 경우 cluquorumd와 cluhbd이 클러스터 구성원에 문제가 있다고 결정을 내릴 것입니다. (*pingInterval* * *sameTimeNetdown*) (모두 다해 14초) 시간이 지난후 자동 오류 복구가 시작될 것입니다.

이 자동 오류 복구 시간을 설정하기 위해서는 cludb 프로그램이 사용됩니다. 예를 들어, 자동 오규 복구 시간을 10초로 낮추기 위해서는 다음과 같이 *sameTimeNetdown* 의 값을 5초로 낮추어야 합니다:

```
cludb -p cluquorumd%sameTimeNetdown 5
```


주의

이런 변수들을 사용할때는 주의를 기울여야 합니다. 만일 지정한 자동 오류 복구의 시간이 너무 짧을 경우, spike가

벌어졌을때, 잘못으로 구성윈이 오류로 들어갔다고 결론질 수 있습니다.

또한 주의 할 것은 만일 전원 스위치의 유형이 "watchgod"일 경우, watchdog의 expiration 시간이 자동 오류 복구 시간보다 짧아야 합니다. 보통 watchdog 시간이 자동 오류 복구 시간의 2/3 으로 지정하실 것을 권장합니다.

부록 B.

추가 소프트웨어 정보

다음 자료는 클러스터 소프트웨어 설정을 관리하는데 도움이 될 것입니다.

B.1. 클러스터 통신 메커니즘

한 클러스터에는 오류 발생시 데이터의 무결성과 올바른 클러스터의 작업을 보증하기 위하여 다음과 같은 여러 가지 내부 클러스터 통신 메커니즘을 사용합니다:

- 한 시스템이 클러스터의 구성원이 될 시기 조정
- 클러스터 시스템의 현 상태 파악
- 클러스터에 오류 발생시 클러스터 행동 조정

클러스터 통신 방법은 다음과 같습니다:

- Quorum 디스크 파티션

  주기적으로, 각 클러스터 시스템은 시간 도장 (timestpamp)과 시스템 상태 (**UP** 혹은 **DOWN**)를 일차 quorum 파티션과 공유 저장 장치에 위치한 원 파티션인 백업 quoram 파티션이 기록합니다. 각 클러스터 시스템은 다른 클러스터에 의해 작성된 시스템 상태 정보와 시간 도장 정보를 읽은 후 자신이 업데이트되었는지 확인합니다. 만일 이 파티션에 문제가 있을 경우, 클러스터 시스템에서 백업 quorum 파티션에서 정보를 읽어들인 후 기본 파티션을 고칩니다. 데이터의 일관성은 체크섬 (checksum)으로 확인되며, 만일 파티션 사이에 일관성에 문제가 있을 경우, 자동으로 고쳐집니다.

  만일 클러스트 시스템이 재부팅된 후 양 quorum 파티션에 정보를 작성할 수 없다면, 시스템은 클러스터에 연결할 수 없습니다. 덧붙여서, 현재 클러스터 시스템이 양 파티션에 작성할 수 없을 경우, 이 시스템은 자동으로 종료하여 클러스터에서 스스로를 제거합니다.

- 원격 전원 스위치 모니터링

  주기적으로, 각 클러스터 시스템은 원격 전원 스위치 연결을 검사할 것입니다. 클러스터 시스템은 이 정보를 다른 클러스터 시스템의 상태를 결정하는데 사용합니다. 하지만, 전원 스위치와 완전한 통신 두절로 오류 복구로 들어가지 않습니다.

- 이더넷과 병렬 Heartbeat

  클러스터 시스템들은 일대일로 이더넷이나 병렬적으로 연결되어있습니다. 주기적으로, 각 클러스터 시스템은 heartbeat (`ping`)을 통해 정보를 보냅니다. 클러스터는 이 정보를 사용해 시스템들의 상태를 파악하며, 클러스터의 문제를 고칠수 있게 도와 줍니다. Heartbeat 통신 방법의 완전한 실패는 오류 복구로 들어가지 않습니다.

만일 클러스터 시스템이, 다른 클러스터 시스템의 quorum 시간 도장이 최신 것이 아니라고 여겨지면, heartbeat 상황을 검사할 것입니다. 만일 그 시스템과의 Heartbeat이 여전히 작동 중이라면, 클러스터 시스템은 아무런 조치도 취하지 않습니다. 그러나 클러스터 시스템이 일정 시간이 지난 후에도 시간 도장을 업데이트하지 않고, heartbeat `ping`에 답하지 않는다면, 클러스터 시스템이 다운된 것으로 간주합니다.

주의하실 점은 클러스터 시스템은 다른 내부 통신 방법이 다 실패하더라도, 최소한 한 시스템이 quorum 디스크 파티션에 쓰기가 가능하다면 클러스터는 계속적으로 운영될 것입니다.

B.2. 클러스터 대몬

클러스터 대몬은 다음과 같습니다:

- Quorum 데몬

  각 클러스터 시스템에, `cluquorumd` quorum 데몬이 주기적으로 시간 도장과 시스템 상태를 기본과 백업 quorum 디스크 파티션의 지정된 장소에 적습니다. 이 데몬은 다른 클러스터 시스템들의 시간 도장과 시스템 상태 자료를 기본 quorum 파티션, 혹은 만일 기본 파티션에 문제가 있다면, 백업 파티션에서 읽어 들입니다.

- Heartbeat 데몬

  각 클러스터 시스템 상에서 **cluhbd** heartbeat 데몬은 양 클러스터 시스템이 연결되어 있는 point-to-point 이더넷과 시리얼 연결 상에 ping 신호를 보냅니다.

- 전원 데몬

  각 클러스터 시스템에서 **clupowerd** 전원 데몬은 원격 전원 연결을 모니터링할 것입니다. 만일 있다면, **clupowerd**이 두개가 실행 되고 있는 것을 보실 것입니다. 하나는 주 프로세스로 주 메시지들에 대답을 할 것입니다. (e.g. 상태, 전원 사이클); 다른 프로세스는 주기적인 전원 스위치의 상태만 보고하게 됩니다.

- 서비스 관리 데몬

  각 클러스터 시스템의, **clusvcmgrd** 서비스 관리 데몬은 클러스터 구성원의 변화에 반응을 하게 되어있습니다. 가끔씩, 하나 이상의 `clusvcmgrd`가 실행되고 있는 것을 볼 수 있습니다. 이것은 `clusvcmgrd`가 서비스의 *`start`*, *`stop`*, 또는 *`monitoring`* 변수를 위해 다른 프로세스를 시작하기 때문입니다.

- 시스템 감시 데몬

  각 클러스터 시스템 상에서 `clumibd` 데몬과 `xmproxyd` 데몬은 클러스터의 모니터링 요청을 담당하고 있습니다. **Red Hat 클러스터 관리자** GUI가 이 서비스의 주 사용원 입니다.

B.3. 오류 복구 시나리오

클러스터에 문제가 발생시, 클러스터가 어떤 단계를 거쳐서 복구를 하는지를 아는것은 문제의 정확한 해결에 크게 도움이 됩니다. 주의하실 것은 클러스터의 행동은 전원 스위치가 설정에 사용되었는지 아닌지에 따라 달라진다는 것입니다. 전원 스위치는 모든 클러스터 실패 환경에서도 데이터의 무결성을 완벽하게 지킬 수 있게 해줍니다.

다음은 다양한 실패와 오류 시나리오에 시스템이 어떻게 반응할 지를 보여줍니다.

B.3.1. 시스템 멈춤

전원 스위치를 사용하는 클러스터 설정에서 만일 시스템이 멈출 경우, 클러스터는 다음과 같이 작업합니다:

1. 제대로 작동 중인 클러스터 시스템은 멈춘 시스템이 quorum 파티션에 시간 도장을 업데이트하지 않으며 heartbeat 채널 상으로 통신하지 않고 있음을 알아냅니다.
2. 제대로 작동 중인 클러스터 시스템은 멈춘 시스템을 전원 사이클 (power-cycle) 시킬 것입니다. 그렇지 않고 만일 감시 타이머가 사용된 경우에는 문제가 생긴 시스템이 스스로 재부팅됩니다.
3. 제대로 작동 중인 시스템은 멈춘 시스템에서 실행 중이던 서비스를 다시 시작할 것입니다.
4. 만일 전에 멈추었던시스템이 재시동한 수, 클러스터에 연결할 수 있다면, (즉, 시스템이 양쪽 quorum 파티션에 쓸수 있다면), 서비스들이 구성원 시스템들에 각 서비스의 방침에 따라 재분배 될 것입니다.

   If the previously *hung* system reboots, and can join the cluster (that is, the system can write to both quorum partitions), services are re-balanced across the member systems, according to each service's placement policy.

전원 스위치를 사용하지 않는 클러스터 설정에서는, 만일 시스템이 멈춘다면, 다음과 같이 행동합니다:

1. 현재 동작 중인 클러스터 시스템이 멈춘 클러스터 시스템이 quorum 파티션에 자신의 시간도장을 업데이트 하지 않거나, heartbeat 채널을 통해 통신하지 않는 것을 발견합니다.
2. 만일 감시 타이머가 사용된다면, 오류가 있는 시스템이 스스로 재시작할 것입니다.

3. 작동 중인 클러스터 시스템은 멈춘 시스템을 **DOWN**으로 quorum 파티션에 보고하며, 멈춘 시스템의 서비스들을 재시작 합니다.
4. 만일 멈추었던 시스템이 재시작되면, 시스템 상태에 **DOWN**로 되어있는 것을 보고, 시스템을 재시작할 것입니다.

   만일 시스템이 계속 멈추어 있으면, 수동으로 멈춘 시스템을 전원-사이클 시켜야, 클러스터로써의 임무를 계속할 수 있습니다.
5. 만일 전에 멈추었던 시스템이 재시작하면, 클러스터에 다시 연결할 수 있으며, 서비스들은 구성원에 서비스의 배치 방침에 따라 배치될 것입니다.

B.3.2. 시스템 패닉

시스템 패닉은 소프트웨어에서 발견한 오류에 의해 조정됩니다. 패닉은 시스템을 정지시킴으로써, 시스템을 안정된 상태로 바꾸려고 하는 것입니다. 만일 클러스터 시스템이 패닉하게 되면 다음과 같은 일이 벌어집니다:

1. 작동중인 클러스터 시스템이, 패닉 중인 클러스터 시스템이 시간도장을 quorum 파티션에 업데이트 하지 않고, heartbeat 채널의 통신이 없음으로 문제를 발견합니다.
2. 패닉 중인 클러스터 시스템은 시스템을 정지한 후 재시작합니다.
3. 만일 전원 스위치가 사용되면, 작동 중인 클러스터 시스템이 패닉 중인 클러스터의 전원-사이클을 시작합니다.
4. 작동중인 클러스터 시스템이 패닉중인 시스템에서 실행되고 있던 서비스들을 재시작 할것입니다.
5. 패닉 후 시스템이 재시작하면, (양 quorum 파티션에 작성할 수 있다면) 클러스터에 연결할 수 있으며, 서비스들은 구성원 시스템들에 방침에 따라 재배치될 것입니다.

B.3.3. Quorum 파티션을 사용할 수 없을때

Quorum 파티션을 사용할 수 없는 경우는 SCSI (혹은 Fibre 채널에) 어댑터에 문제가 발생하거나, 혹은 SCSI 케이블이 공유 저장 디스크에서 분리되었을 때 일어납니다. 만일 이 중 한가지 상황이라도 발생한다면, SCSI 버스는 정지된 상태로 있으며, 클러스터는 다음과 같이 행동합니다:

1. Quorum 파티션을 사용할 수 없는 클러스터 시스템은 자신의 시간 도장을 quorum 파티션에 업데이트 할수 없는 것으로 보고, 재시작하려고 할 것입니다.
2. 만일 클러스터 설정에 전원 스위치도 있다면, 작동 중인 클러스터 시스템이 전원-사이클을 통해 시스템을 재시작하려 할 것입니다.
3. 작동 중인 클러스터 시스템은 quorum 파티션을 사용할 수 없는 시스템에서 실행중이던 서비스를 재실행할 것입니다.
4. 만일 클러스터 시스템이 재시작하면, (또한 모든 quorum 파티션에 쓸 수 있으면) 클러스터에 다시 연결할 수 있으며, 서비스들은 자신들의 방침에 따라 재배치될 것입니다.

B.3.4. 모든 네트워크 연결 실패

모든 heartbeat 네트워크 연결에 실패하면 모든 네트워크에 연결 실패하게 됩니다. 다음과 같은 이유에서 네트워크 연결 실패가 발생합니다:

- 모든 heartbeat 네트워크 케이블이 시스템에서 분리되었습니다.
- heartbeat 연결에 사용된 모든 벙럴 연결과 네트워크 인터페이스가 실패했습니다.

만일 모든 네트워크 연결 실패가 일어나면, 두 시스템이 문제를 인식하고, SCSI 디스크 연결이 아직도 살아있다는 것을 알아냅니다. 따라서 서비스가 시스템 상에서 계속 실행되며, 방해되지 않을 것입니다.

만일 모든 네트워크 연결 실패가 일어나면, 문제를 진단하고, 다음 중 한가지 방법을 선택하십시오:

- 만일 문제가 한 클러스터 시스템만 문제가 된다면, 이 시스템의 서비스를 재배치 하십시요. 그리고, 문제를 고친 후에, 다시 시스템들을 재배치 하십시요.
- 수동으로 한 시스템의 서비스를 정지하십시요. 이 때, 서비스들은 자동적으로 오류 복구로 들어가지 않습니다. 그 대신, 서비스를 다른 시스템에서 수동으로 시작하십시요. 문제가 고쳐진 후에, 서비스들을 시스템들 중에 재배치 하실 수 있습니다.
- 만일 다음과 같은 상황이 발생한다면, 클러스터 시스템을 정지하시기 바랍니다:
  1. 정지시킨 클러스터 시스템의 서비스들이 정지되었습니다.
  2. 남은 클러스터 시스템들이 시스템이 정지중이라고 얘기합니다.
  3. 정지시켰던 시스템에서 실행 중이던 서비스가 다른 클러스터 시스템에서 시작되었습니다.
  4. 만일 시스템이 재시작하면, (두 quorum 파티션에 쓸수 있으면) 클러스터에 연결할 수 있으며, 서비스들은 각 방침에 따라 구성원 시스템들에 재배치됩니다.

B.3.5. 원격 전원 스위치 연결 실패

만일 원격 전원 스위치 연결에 대한 질의에 실패했지만, 양 시스템에 계속적으로 전원이 켜져있다면, 클러스터 행동에 바뀌는 것이 없습니다. 전원 데몬은 문제가 있다는 메시지 (예, 케이블이 분리되었다는 메시지)를 계속적으로 보낼 것입니다.

만일 클러스터 시스템이 문제 있는 리모트 전원 스위치를 사용하려고 하면, 현재 문제가 있는 시스템에서 실행되고 있는 서비스는 정지 할 것입니다. 그러나, 데이타의 무결성을 위해서, 다른 시스템으로 오류 복구 되지는 않을 것입니다. 하지만, 하드웨어 문제가 풀릴때까지 정지되어 있을 것입니다.

B.3.6. Quorum 데몬 문제

만일 quorum 데몬이 한 클러스터 시스템에서 문제가 나면, 시스템은 더 이상, quorum 파티션을 감시할 수 없습니다. 만일 클러스터에서 전원 스위치가 사용되지 않았다면, 이러한 오류는 한 클러스터 이상의 시스템에서 서비스가 실행되지 않거나, 데이터의 오류를 초래할 수도 있습니다.

만일 quorum 데몬에 문제가 생긴 경우, 전원 스위치가 사용되었다면, 다음과 같은 일이 벌어질 것입니다:

1. 작동중인 클러스터 시스템에서 quorum 데몬에 문제가 생긴 시스템에서 heartbeat 채널에서는 통신이 가능하지만 시간 도장을 업데이트하지 않는 것을 알게될 것입니다.
2. 한 동안의 시간이 지난후, 작동중인 클러스터 시스템에서 quorum 데몬에 문제가 생긴 시스템에 전원 사이클 시킬것입니다. 혹은 watchdog timer가 사용될 경우, 문제가 생긴 시스템이 자동으로 재시작 할 것입니다.
3. 작동 중인 클러스터 시스템은 quorum 데몬에 문제가 있는 시스템에서 실행중이던 서비스들을 재실행할 것입니다.
4. 만일 클러스터 시스템이 재부팅한 후, (모든 quorum 파티션에 쓸수 있다면) 클러스터에 연결할 수 있고, 서비스들은 각자의 방침에 따라, 구성원 시스템에 재배치될 것입니다.

만일 quorum 데몬에 문제가 생기고 전원 스위치나 감시 타이머가 클러스터에 사용되지 않았다면, 다음과 같은 작업을 수행하십시오:

1. 작동중인 클러스터 시스템에서 quorum 데몬에 문제가 생긴 시스템에서 heartbeat 채널에서는 통신이 가능하지만 시간도장을 업데이트 하지 않는것을 알게 될 것입니다.

2. 작동 중인 클러스터 시스템은 quorum 데몬에 문제가 있는 시스템에서 실행 중인 서비스들을 재시작할 것입니다. 하지만 다른 때와는 다르게, 양쪽의 시스템에서 서비스를 동시에 시작할 수도 있으며, 그것으로 인해 데이타에 문제가 생길 수도 있습니다.

B.3.7. Heartbeat 데몬 문제

만일 클러스터 시스템의 heartbeat 데몬에 문제가 생긴 경우, 서비스의 오류 복구 시간은 오래 걸릴 것입니다. 그 이유는 quorum 데몬이 다른 클러스터 시스템의 상태를 알 수 없기 때문입니다. 이 문제의 하나로 클러스터 시스템은 서비스 오류 복구를 시작하지 않을 것입니다.

B.3.8. 전원 데몬 문제

만일 한 클러스터의 전원 데몬에 문제가 생기고 다른 시스템에 큰 문제가 생길 경우, (예를 들어, 시스템 패닉), 클러스터 시스템은 문제가 생긴 시스템에 전원-사이클을 시작할 수 없습니다. 그 대신, 클러스터 시스템은 자신의 서비스들을 실행 할 것이며, 문제가 있는 시스템은 오류 복구가 되지 않을 수 있습니다. 클러스터 행동은 원격 파워 스위치 언결 문제와 같습니다.

B.3.9. 서비스 관리 데몬 문제

만일 서비스 관리 데몬에 문제가 생길시에는, 서비스 관리 데몬을 재시작하거나, 시스템을 재시작할 때까지 서비스를 시작하거나 정지할 수 없습니다. 서비스 관리 데몬을 가장 쉽게 재시작 하는 방법은 클러스터 소프트웨어를 정지시키고 재시작하는 것입니다. 예를 들어, 서비스를 정지하려면, 다음과 같은 명령어를 사용합니다:

```
/sbin/service cluster stop
```

클러스터 소프트웨어를 재시작하시려면 다음 명령을 입력하시면 됩니다:

```
/sbin/service cluster start
```

B.3.10. 모니터링 데몬 문제

만일 클러스터 모니터링 데몬 (`clumibd`)에 문제가 생기면, GUI를 통해 클러스터를 감시할 수 없습니다. 기억 할 것은, 클러스터 구성원이 아닌 시스템에서 원격으로 클러스터의 GUI를 통해 시스템을 감시하시려면, **cluconfig**에 지정하시기 바랍니다.

B.4. 클러스터 데이터베이스 영역

클러스터 데이터베이스의 복사본은 `/etc/opt/cluster/cluster.conf` 파일에 저장되어 있습니다. 이 파일에는 클러스터 구성원과 서비스에 대한 자세한 내용이 들어 있습니다. 수동으로 설정 파일을 편집하지 마십시요.. 대신 클러스터 도구를 사용해서 클러스터 설정을 편집하시기 바랍니다.

cluconfig을 실행하실 때, *`[members]`* 영역에 입력된 사이트에 관련된 데이터베이스에 입력됩니다. 다음은 각 클러스터 구성원 영역이며 대한 설명입니다:

```
start member0
start chan0
 device = serial_port
 type = serial
end chan0
```

Heartbeat 채널에 사용될 null 모델 케이블이 연결되어 있는 tty 포트를 적어 주십시요. 예를 들어 **serial_port**는 **/dev/ttyS1** 입니다.

```
start chan1
 name = interface_name
 type = net
end chan1
```

이더넷 heartbeat채널을 위한 네트워크 인터페이스를 지정해 주십시요. **interface_name**는 인터페이스가 지정된 호스트 이름입니다. (예를 들어, **storage0**).

```
start chan2
 device = interface_name
 type = net
end chan2
```

두번째 이더넷 heartbeat채널을 위한 네트워크 인터페이스를 지정해 주십시요. **interface_name**는 인터페이스가 지정된 호스트 이름입니다. (예, **cstorage0**). 이 필드에는 1:1 heartbeat 채널을 지정할 수 있습니다.

```
id = id
name = system_name
```

클러스터 시스템의 지정 번호 (0 혹은 1)과, hostname을 사용했을 때, 받을 이름을 지정해 주십시요. (예를 들어, **storage0**).

```
powerSerialPort = serial_port
```

전원이 연결된 빙렬 포트를 위한 장치 특벌 파일을 지정해 주십시요. (예를 들어, **/dev/ttyS0**).

```
powerSwitchType = power_switch
```

전원 스위치의 유형을 지정해 주십시요. **RPS10**, **APC**, 혹은 **None**.

```
quorumPartitionPrimary = raw_disk
quorumPartitionShadow = raw_disk

end member0
```

일차 quorum 파티션과 백업 Quorum 파티션을 위한 원 장치를 지정해 주십시요. (예를 들어, **/dev/raw/raw1** 와 **/dev/raw/raw2**).

클러스터 서비스를 추가하실 때, 서비스-관련 정보는 데이터베이스 안의 *[services]* 라는 영역에 지정하시기 바랍니다. 다음은 각 클러스터 서비스 영역에 알맞는 자료입니다:

```
start service0
name = service_name
disabled = yes_or_no
userScript = path_name
```

서비스 이름을 지정하시고, 서비스가 추가된 후에 비활성화 여부와 사용할 시작/정지 스크립트의 완전 경로를 지정해 주십시요.

```
preferredNode = member_name
relocateOnPreferredNodeBoot = yes_or_no
```

서비스의 우선 구성원의 이름을 지정하고 시스템이 재시작 후 클러스터에 연결했을 때, 서비스를 재배치할 지를 지정해 주십시요.

```
start network0
ipAddress = aaa.bbb.ccc.ddd
netmask = aaa.bbb.ccc.ddd
broadcast = aaa.bbb.ccc.ddd
end network0
```

서비스에서 사용할 IP 주소와 만약 있다면, 넷마스크와 브로드캐스트 주소를 지정해 주십시요. 한 서비스에 여러 IP 주소가 지정될 수 있습니다.

```
start device0
 name = device_file
```

서비스에서 사용되는 특별 장치 파일을 지정해 주십시요. (예를 들어, **/dev/sda1**). 한 서비스를 위해 여럿의 장치가 있을 수 있습니다.

```
 start mount
 name = mount_point
 fstype = file_system_type
 options = mount_options
 forceUnmount = yes_or_no
```

만일 있다면, 장치를 사용한 장착점과, 파일 시스템 유형, 그리고 강제적인 탈착의 활성화 여부를 지정해 주십시요.

```
 owner = user_name
 group = group_name
 mode = access_mode
end device0
end service0
```

장치의 주인과 그룹 그리고 이용되는 모드를 지정해 주십시요.

B.5. Red Hat 클러스터 관리자와 Piranha 사용하기

부하분산 능력에 덧붙여, 전자 상거래 사이트의 완벽한 데이터 무결성과 고성능 프로그램을 위해서 클러스터는 Piranha와 함께 사용될 수 있습니다.

그림 B-1은 **Red Hat 클러스터 관리자** 와 Piranha 가 어떻게 함께 사용될 수 있는지를 보여주고 있습니다. 이 그림이 보여주는 것과 같이 클러스테는 3계층을 가지게 됩니다. 여기서 가장 윗층은 Piranha 부하분산 시스템이 웹의 요청을 담당합니다. 두번째 층은 몇개의 웹서버들로써, 들어온 요청을 제공하고 있습니다. 세번째 단계는 클러스터들로써, 데이터를 웹서버에 제공하고 있습니다.

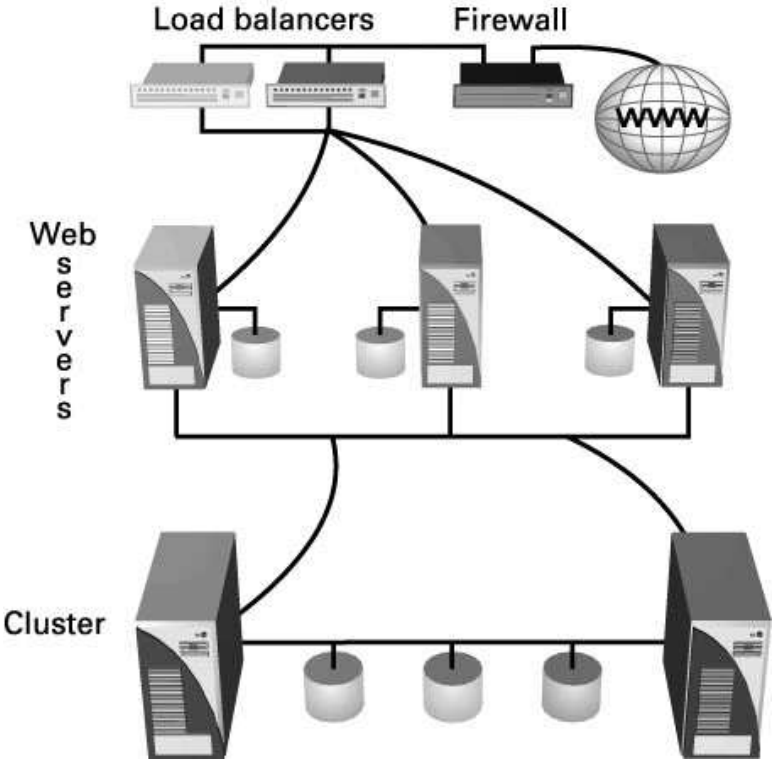

그림 B-1. LVS 환경에서의 클러스터

Piranha 설정에서, 클라이언트 시스템은 WWW을 통해서 정보를 요청합니다. 보안 이유로 인해 이러한 요청은 리눅스 시스템으로 만들어지거나 전형적인 방화벽을 거쳐 웹사이트로 옵니다. 보다 조심스럽기 위해, 방화벽 장치를 오류 복구 설정할 수도 있습니다. 방화벽 뒤에는 Piranha, active-standby 모드로 설정된 부하분산 시스템이 있습니다. 작동중인 부하 분산 시스템은 요청을 웹 서버들로 보냅니다.

각 웹 서버는 개별적으로 클라이언트에게서 온 요청을 처리하고 클라이언트에게로 보낼 수 있습니다. Piranha는 관리자가 활동중인 웹서버에 웹서버들을 더함으로써, 웹사이트의 최대 한도를 늘릴수 있게 해줍니다. 더불어, 만일 웹 서버에 문제가 생기면, 그냥 제거할 수도 있습니다.

이 Piranha 설정은 작은 내용에 자주 바뀌지 않는 정적인 웹 내용을 가지고 있는 웹서버들에 알맞습니다, 예를 들어, 회사 로고들, 그래서 쉽게 복제될수 있게 말입니다. 그렇나, 이 설정은 계속적으로 바뀌는 동적인 제품 내역이라든지, 구매 내용을 가진 웹서버에는 좋지 않습니다. 동적인 내용에는 제품 내역이라든지, 구매 내용, 혹은 고객 데이타 베이스와 같이 모든 웹서버에 동일 해야 하며, 최신의 정보로 업데이트 되어야 하는 정보들이 있습니다.

동적인 웹 내용을 Piranha 설정으로 지원하려면, 전의 그림에서 본 것 같이 웹서버 뒤에 클러스터를 더합니다. 이렇게 Piranha와 **Red Hat 클러스터 관리자**를 더하게 되면, 높은 데이타 무결성, 오류 복구가능한 전자 상거래 사이트가 나오는 것입니다. 클러스터는 high-availability 데이타 베이스를 실행하며, 혹은 웹 서버에서 이용 가능한 네트워크 데이터베이스를 실행할 수 있습니다.

예를 들어, 그림은, 웹 사이트를 통해 들어오는 온라인 구매를 담당하는 전자 상거래 사이트의 보여주고 있습니다. 클라이언트가 URL을 통해 정보를 요청하면, 방화벽을 지나 작동중인 Piranha 부하분산 시스템으로 가고, 그 곳에서 요청을 셋 중에 하나인 웹 서버로 전달하고, **Red Hat 클러스터 관리자** 시스템이 동적 데이타를 웹서버에 지원하며, 클라이언트 시스템까지 전달되는 것입니다.

색인