

Linux, iSCSI 환경에서 Oracle RAC 10g Release 2 클러스터 설치하기 낙서장

2007/07/05 21:01

<http://blog.naver.com/s5b8s4/140039720644>

Linux, iSCSI 환경에서 Oracle RAC 10g Release 2 클러스터 설치하기

저자 - Jeffrey Hunter 

(2,500달러 이하의) 저렴한비용으로 Oracle RAC 10g Release 2 클러스터 개발환경을 셋업하고 설정하는 방법을 배워 보십시오.

개발, 테스트 용도로만 사용할 수 있습니다. 운영 목적의 시스템에는 적용될 수 없습니다!

업데이트일 - 2007년 6월

목차

1. 서론
2. Oracle RAC 10g 개요
3. 공유 스토리지 개요
4. iSCSI 테크놀로지
5. 하드웨어 및 비용
3. Linux 운영체제의 설치
7. 네트워크 설정
3. Openfiler 설치
3. Openfiler를 이용한 iSCSI 볼륨 설정
3. Oracle RAC 노드의 iSCSI 볼륨 설정
1. "oracle" 사용자 및 디렉토리 생성
2. 오라클 환경을 위한 Linux 서버 설정
3. hangcheck-timer 커널 모듈의 설정
4. 원격 액세스를 위한 RAC 노드 설정
5. 양쪽 Oracle RAC 노드를 위한 스타트업 커맨드
3. Oracle Cluster File System(OCFS2)의 설치 및 설정
7. Automatic Storage Management (ASMLib 2.0)의 설치 및 설정
3. Oracle 10gRAC 소프트웨어 다운로드
3. Oracle10g Release 2 설치를 위한 사전 준비 작업
3. Oracle 10gClusterware 소프트웨어 설치
1. Oracle 10g Database 소프트웨어 설치
2. Oracle 10g Companion CD 소프트웨어 설치
3. TNS 리스너 프로세스 생성
4. 오라클 클러스터 데이터베이스의 생성
5. TNS 네트워킹 파일의 검증
3. Create / Alter Tablespace
7. RAC 클러스터 및 데이터베이스 설정의 검증
3. 클러스터의 시작 / 종료
3. Transparent Application Failover - (TAF)
3. 트러블슈팅
1. 결론
2. 감사의 말씀

아티클 관련 다운로드:

- * Oracle Enterprise Linux Release 4 Update 5 — (Available for x86 and x86_64)
- * Oracle Database 10g Release 2 EE, Clusterware, Companion CD - (10.2.0.1.0)
- * Openfiler 2.2 (respin 2) — (openfiler-2.2-x86-disc1.iso -OR- openfiler-2.2-x86_64-disc1.iso)
- * ASMLib 2.0 Library - (2.0.2-1) — oracleasmllib-2.0.2-1.i386.rpm
- * Support files

1. 서론

Oracle Real Application Clusters (RAC) 10g 테크놀로지를 이해하는 가장 쉽고 빠른 방법은 Oracle RAC 10g 클러스터를 직접 설치해 보는 것입니다. Oracle RAC 10g가 제공하는 기능(폴트 톨러런스, 보안, 로드 밸런싱, 확장성 등)을 이해하는데 이보다 효과적인 방법은 없습니다.

하지만 운영 시스템에 RAC를 구성하는데 드는 비용이 지나치게 높다는 문제가 걸림돌이 됩니다. 소규모

모 2-노드 클러스터를 구성하는 비용은 적게는 1만 불(미화 기준)에서 많게는 2만 불에 이를 수도 있습니다. 또 RAC 환경의 기반이 되는 SAN(Storage Area Network)을 구축하려면 최소 1만 불이 요구됩니다.

이 가이드는 많은 비용을 들이지 않고 Oracle RAC 10g에 친숙해지고자 하는 사용자들을 위해, 상용 소프트웨어 컴포넌트와 무료 소프트웨어를 이용하여 (2,200 - 2,500 달러 정도의) 저렴한 비용으로 Oracle RAC 10g Release 2 시스템을 구축하는 방법을 소개합니다. 시스템은 Linux(CentOS 4.4 또는 Red Hat Enterprise Linux 4 Update 4), Oracle10g Release 2, OCFS2, ASMLib 2.0을 기반으로 (각각 하나의 프로세서를 장착한) 2 노드 클러스터로 구성됩니다. Oracle RAC의 모든 공유 디스크 스토리지는 iSCSI 를 기반으로 하며, 이를 위해 *네트워크 스토리지 서버*(Openfiler Release 2.1)가 사용됩니다. (물론 VMWare Virtual Machine에 가상 클러스터 가상 클러스터를 구현하는 방안도 고려해 볼 수 있을 것입니다. 하지만 교육 효과가 같을 수는 없습니다!)

본 문서의 다음 버전에서는 CentOS, Red Hat Linux 대신 오라클의 Enterprise Linux 가 사용될 것임을 참고하시기 바랍니다. Enterprise Linux는 OCFS2, ASMLib 소프트웨어 패키지를 포함하고 있으며, 기존 Linux OS와 동등한 (또는 보다 개선된) 안정성을 제공할 것입니다. Enterprise Linux에 대한 자세한 정보는 <http://www.oracle.com/linux> 를 참고하시기 바랍니다. 이 구성이 지금까지 Enterprise Linux 환경에서 테스트된 적은 없지만, 최소한의 변경 작업만으로 실행이 가능할 것이라 예상할 수 있습니다.

rPath Linux를 기반으로 하는, Openfiler 는 무료 브라우저 기반 네트워크 스토리지 관리 유틸리티로, 파일 기반 NAS(Network Attached Storage)와 블록 기반 SAN(Storage Area Network)을 단일 프레임워크를 통해 지원합니다. Openfiler는 CIFS, NFS, HTTP/DAV, FTP 등을 지원합니다. 하지만 여기에서는 iSCSI 지원 기능만을 이용하여 Oracle 10g RAC에서 요구하는 SAN 기반 공유 스토리지 환경을 저렴하게 구성하기로 합니다. 네트워크 스토리지 서버(본 문서에서는 Openfiler 서버라 불리기도 합니다)에는 USB 2.0 인터페이스를 통해 외장형 500 GB 하드 드라이브가 연결됩니다. 본 문서에서는 Openfiler 서버를 이용하여 이 디스크를 iSCSI 기반 스토리지로 설정한 후, Oracle Clusterware와 Oracle ASM 볼륨이 요구하는 공유 파일을 저장하기 위해 사용합니다.

참고: 본 문서는 교육을 목적으로 작성되었으며, 기본적인 개념의 이해가 용이하도록 셋업 과정을 단순화하였습니다. 예를 들어 디스크 미러링은 하나의 물리적 디스크 상에 구현되었습니다. (실제 환경이라면 최소한 2 개의 물리적 디스크를 사용해야 할 것입니다.)

여기서 설명된 내용이 저가형 Oracle RAC 10g 시스템을 구축하는 유일한 방법은 아닙니다. 필자는 SCSI를 기반으로 공유 스토리지 컴포넌트를 구현하는 다른 솔루션을 구현해 보기도 했습니다. 경우에 따라서는 SCSI가 여기서 설명된 구성보다 더 많은 비용을 요구할 수도 있습니다. 저가형 SCSI 구성의 예가 아래와 같습니다:

SCSI 컨트롤러: SCSI 컨트롤러 2 조 (컨트롤러 당 가격: \$20 (*Adaptec AHA-2940UW*) - \$220 (*Adaptec 39320A-R*))

SCSI 인클로저: \$70 - (*Inclose 1 Bay 3.5" U320 SCSI Case*)

SCSI 하드 드라이브: \$140 - (*36GB 15K 68p U320 SCSI Hard Drive*)

SCSI 케이블: SCSI 케이블 2 조 (개당 \$20) - (*3ft External HD68 to HD68 U320 Cable*)

일부 마더보드의 경우 빌트인 SCSI 컨트롤러를 기본 탑재하고 있음을 참고하십시오.

기존의 Oracle9i, Oracle 10g Release 1 가이드에서는 로우 파티션(raw partition)을 이용하여 공유 스토리지에 파일을 저장했습니다. 하지만 여기에서는 Oracle Cluster File System Release 2(OCFS2)와 Oracle Automatic Storage Management(ASM) 기능을 활용하기로 합니다. 두 대의 Oracle RAC 노드는 아래와 같이 구성됩니다:

오라클 데이터베이스 파일				
RAC 노드 네임	인스턴스 네임	데이터베이스 네임	\$ORACLE_BASE	DB 파일을 위한 파일 시스템/볼륨 관리자
linux1	orcl1	orcl	/u01/app/oracle	ASM
linux2	orcl2	orcl	/u01/app/oracle	ASM
Oracle Clusterware 공유 파일				
파일 타입	파일 네임	iSCSI 볼륨 네임	마운트 포인트	파일 시스템
Oracle Cluster Registry	/u02/oradata/orcl/OCRFile	crs	/u02/oradata/orcl	OCFS2
CRS Voting Disk	/u02/oradata/orcl/CSSFile	crs	/u02/oradata/orcl	OCFS2

Oracle Database 10g Release 2 (10.2)부터 Cluster Ready Services(CRS)의 명칭이 Oracle Clusterware로 변경되었음을 참고하시기 바랍니다.

Oracle Database 10g Release 2 (10.2) 이후 버전에서는 Oracle Clusterware를 (버전과 무관한) 별도의 Oracle Clusterware 홈 디렉토리에 설치해야 합니다. Optimal Flexible Architecture(OFA) 룰에도

변경 사항이 있습니다. Oracle Clusterware를 (버전별로 지정된) Oracle home 마운트 포인트 (예: /u01/app/oracle/product/10.2.0/...)에 설치해서는 안됩니다. 이후 버전의 Oracle Clusterware가 동일한 경로 상의 Oracle Clusterware 설치 파일을 덮어쓰기 때문입니다. 또 Oracle Clusterware 10g Release 2(10.2)가 기존의 Oracle Cluster Ready Services 설치본을 감지하는 경우, 동일한 경로 상의 기존 설치 파일을 덮어 쓰게 됩니다.

Oracle Clusterware 소프트웨어는 RAC 클러스터를 구성하는 각 노드의 /u01/app/oracle/product/crs 경로에 설치됩니다. Oracle Clusterware 소프트웨어는 두 종류의 파일-Oracle Cluster Registry (OCR) 파일과 "Voting Disk" 파일-이 클러스터의 모든 노드에 의해 공유될 것을 요구합니다. 이 두 파일은 Oracle's Cluster File System Release 2(OCFS2)를 사용한 공유 스토리지에 설치됩니다. 이 파일을 로우 디바이스(raw device)를 설치하는 것도 가능하지만, 오라클은 이러한 구성을 권장하지 않으며 이렇게 구성된 두 가지 Clusterware 파일을 ASM이 사용할 수 없다는 문제가 따릅니다.

Oracle 10g Release 2 데이터베이스 소프트웨어는 RAC 클러스터를 구성하는 각 노드의 별도 Oracle Home 경로(/u01/app/oracle/product/10.2.0/db_1)에 설치됩니다. 모든 물리적 데이터베이스 파일 (데이터, 온라인 리두 로그, 컨트롤 파일, 아카이브 리두 로그)은 공유 드라이브의 다른 파티션에 설치되며 ASM(Automatic Storage Management)에 의해 관리됩니다. (물론 오라클 데이터베이스 파일을 그냥 OCFS2에 저장하는 것도 가능합니다. 하지만 ASM을 직접 구성해 보으로써 보다 많은 것을 배우실 수 있을 것입니다!)

참고: 본 문서는 아래에 기술된 환경을 그대로 사용하는 것을 가정하여 작성되었습니다. 본 문서에서 언급된 구성요소를 다른 구성요소로 대체하는 것은 일체 허용되지 않습니다. (다만 서버, 네트워킹 장비, 외장형 하드 드라이브 등의 벤더 하드웨어는 다른 제품으로 교체될 수 있습니다.) 구입한 하드웨어가 Red Hat Linux 4에 의해 지원되는지 확인하시기 바랍니다.

RHEL 4와 FireWire 기반의 Oracle RAC 10g Release 2 설치 가이드가 필요하신 경우에는, 이곳을 누르십시오(한글).

RHEL3 기반 Oracle RAC 10g Release 1 설치 가이드가 필요하신 경우, 이곳을 누르십시오(한글).

Oracle 9/RAC 버전의 가이드를 참고하시려면, 이곳을 누르십시오(영문).

2. Oracle RAC 10g 개요

Oracle 9에서 처음 소개된 Oracle RAC는 Oracle Parallel Server(OPS)의 다음 세대 솔루션입니다. RAC는 여러 개의 인스턴스가 동일한 데이터베이스(스토리지)에 동시 액세스할 수 있는 환경을 제공합니다. 모든 노드가 동일한 데이터에 액세스하고 특정 인스턴스에 장애가 발생해도 데이터베이스의 가용성이 희생되지 않으므로, 폴트 톨러런스, 로드 밸런싱, 수평적 확장을 통한 가용성 및 성능 개선이 가능합니다.

Oracle 10g RAC의 중심에는 공유 디스크 서브시스템이 있습니다. 클러스터의 모든 노드는 데이터, 리두 로그 파일, 컨트롤 파일, 매개변수 파일에 동시 접근할 수 있어야 합니다. 또 모든 노드가 데이터베이스에 접근하는 것을 허용하려면 데이터 디스크를 글로벌하게 공유해야 합니다. 마지막으로, 각각의 노드는 리두 로그 파일과 컨트롤 파일을 개별적으로 저장하고 관리하지만, 다른 노드에 시스템 장애가 발생한 경우 복구를 위해 이 파일에 접근할 수 있어야 합니다.

Oracle RAC와 OPS의 가장 큰 차이점의 하나로 Cache Fusion 테크놀로지를 들 수 있습니다. OPS에서 노드 간의 데이터 요청을 처리하려면, 요청 노드가 데이터를 읽기 전에 먼저 데이터를 디스크에 저장해야 합니다. Cache Fusion을 사용하는 RAC에서는 데이터가 정교한 락(lock) 알고리즘을 기반으로 고속 네트워크를 통해 다른 노드에 직접 전달됩니다.

모든 클러스터링 솔루션이 공유 스토리지를 사용하는 것은 아닙니다. 일부 벤더는 "federated cluster"라 불리는 방법을 사용합니다. 이 환경에서는 데이터가 여러 대의 서버에 분산되어 저장된 형태로 공유됩니다. 이와 달리, Oracle 10g RAC에서는 모든 노드가 동일한 저장 공간을 공유합니다. Oracle 10g RAC의 데이터 파일, 리두 로그 파일, 컨트롤 파일, 아카이브 로그 파일은 로우 디바이스(raw device), NAS, SAN, ASM, 또는 클러스터 파일 시스템(clustered file system) 기반의 공유 스토리지에 저장됩니다. 오라클의 솔루션은 모든 노드의 프로세싱 파워의 활용도를 극대화하는 동시에 페일오버 작업의 안정성을 보장한다는 이점을 제공합니다.

Dell, IBM, HP 등의 벤더는 운영 환경을 위해 사전 구성된 Oracle 10g RAC 솔루션을 제공하고 있습니다. 하지만 본 문서에서는 Linux 서버와 iSCSI를 이용하여 저렴한 가격의 테스트/개발용 Oracle 10g RAC 환경을 구축하는 방법을 설명하는데 초점을 맞추고 있습니다.

Oracle RAC에 대한 자세한 정보는 Oracle RAC Product Center를 참고하시기 바랍니다.

3. 공유 스토리지 개요

파이버 채널(Fiber Channel)은 공유 스토리지 환경에서 가장 널리 사용되는 기술입니다. 파이버 채널은 시스템과 스토리지 디바이스를 포인트-투-포인트(FC-P2P), 루프(FC-AL), 또는 스위치 토폴로지(FC-SW) 형태로 연결하기 위한 고속 시리얼 전송 인터페이스입니다. 파이버 채널은 SCSI 프로토콜과 IP 프로토콜을 지원합니다. 파이버 채널 구성은 최대 127 개의 노드와 각 방향별로 초당 2.12 기가비트의 전송률을 지원합니다. 실효 전송률은 4.25 Gbps입니다.

하지만 파이버 채널은 가격이 매우 비싸다는 문제가 있습니다. 스위치 하드웨어에만 1,000불 가량이 필요하며, 하이-엔드 드라이브의 가격은 300불(36GB 드라이브 기준)에 이릅니다 서버에 장착할 파이버 채널 카드를 포함하는 전형적인 파이버 채널 구성(서버 어댑터 포함)에는 약 10,000불의 비용이 필요합니다(물론 여기에는 클러스터를 구성하는 서버 가격이 포함되어 있지 않습니다).

파이버 채널보다 저렴한 대안으로 SCSI가 있습니다. SCSI 테크놀로지는 공유 스토리지 환경에서 적절한 수준의 성능을 제공하지만, Linux의 GPL 기반 가격 체계에 익숙한 관리자와 개발자에게는 이것도 비싸게 느껴질 수 있습니다. SCSI를 이용하여 2-노드 클러스터를 구성하는 데에는 대략 2,000~5,000불이 듭니다.

그 밖에 대중적인 솔루션으로 NAS 기반의 Sun NFS (Network File System)가 있습니다. Sun NFS 역시 공유 스토리지 환경에 응용될 수 있지만, Network Appliance 장비(또는 호환 장비)가 필요하다는 제약 사항이 있습니다 또 서버가 NFS를 통한 direct I/O를 지원해야 하며, TCP를 전송 프로토콜을 사용해야 하며, 읽기/쓰기 블록 사이즈는 32K로 설정되어야 하는 등의 요구사항이 존재합니다.

본 문서에서 사용되는 공유 스토리지는 Openfiler가 설치된 네트워크 스토리지 서버와 iSCSI 테크놀로지를 기반으로 구현됩니다. 이 솔루션은 테스트 및 교육 환경에서 파이버 채널을 대체할 수 있는 저렴한 대안을 제공합니다. 하지만 로우-엔드 하드웨어가 사용되고 있는 점을 감안할 때 운영 환경에 적용되어서는 안될 것입니다.

4. iSCSI 테크놀로지

오랜 세월 동안 네트워크 기반 스토리지 솔루션을 구현하기 위한 유일한 대안은 Fibre Channel Storage Area Network (FC SAN) 밖에 존재하지 않았습니다. *Fiber Distributed Data Interface (FDDI)*, 라는 초기의 ANSI 프로토콜을 기반으로 하는 Fibre Channel은 *storage network*상에서 SCSI 커맨드를 전달하기 위한 방안으로서 개발되었습니다.

FC SAN은 뛰어난 성능과 개선된 디스크 사용률, 가용성, 확장성을 제공합니다. 또 서버 클러스터링을 지원한다는 매우 중요한 장점을 제공합니다! 하지만 아직까지도 FC SAN에는 세 가지 중요한 단점이 존재합니다. 첫 번째는 가격입니다. FC SAN의 구현에 드는 비용이 최근 몇 년 동안 감소 추세에 있었던 것은 사실이지만, 제한된 IT 예산을 가진 소규모 기업의 입장에서는 여전히 부담스러운 가격입니다. 두 번째로 하드웨어 컴포넌트의 호환성 문제가 있습니다. 표준이 발표된 이후로 각 벤더들은 서로 다른 방식으로 Fiber Channel 표준을 해석하였고, 이로 인해 다양한 인터커넥트 문제가 발생하고 있습니다. 단 동일한 벤더로부터 Fiber Channel 컴포넌트를 구입할 때에는 이것이 문제가 되지 않습니다. 세 번째로 Fiber Channel은 이더넷이 아니라는 점을 지적해야 할 것입니다! 따라서 Fiber Channel은 데이터 센터 네트워크와는 구분된 별도의 네트워크 테크놀로지와 스킬을 필요로 합니다.

Gigabit Ethernet이 대중화되고 가격 절감의 요구가 증대되면서, 최근에는 Fiber Channel 대신 iSCSI 기반 스토리지 시스템을 구현하는 사례가 늘고 있습니다. 오늘날, iSCSI SAN은 FC SAN의 주요 경쟁 기술로 자리잡고 있습니다.

2003년 2월 11일 IETF(Internet Engineering Task Force)에 의해 인준된 iSCSI(Internet Small Computer System Interface)는 IP 기반 스토리지 디바이스, 호스트, 클라이언트 간의 연결 설정 및 관리를 위한 IP 기반 스토리지 네트워킹 표준입니다. iSCSI는 SCSI-3 프레임워크 내에 정의된 데이터 전송 프로토콜로 스토리지 네트워크 상에서 *block-level* 데이터 전송을 수행한다는 점에서 Fiber Channel과 유사합니다. 블록-레벨 커뮤니케이션이란 호스트/클라이언트 간에 데이터가 "블록(block)"이라는 단위로 전송됨을 의미합니다. (대부분의 NAS 시스템에서 사용되는 파일 레벨 커뮤니케이션과 달리) 데이터베이스 서버가 정상적으로 지원되기 위해서는 블록-레벨 커뮤니케이션이 필수적입니다. FC SAN과 마찬가지로, iSCSI SAN은 스토리지를 위한 별도의 물리적 네트워크로 구성됩니다. 하지만 컴포넌트들은 일반적인 IP 네트워크(LAN)과 동일한 형태로 구성될 수 있습니다.

iSCSI의 미래가 밝다는 사실은 분명하지만, 아직까지는 성능적인 문제로 인해 비판을 받기도 합니다. iSCSI는 우리가 이미 친숙한 IP 네트워크를 전송 매커니즘으로 사용한다는 가장 중요한 특징을 가집니다. 하지만 TCP/IP 프로토콜은 매우 복잡할 뿐 아니라 CPU 자원을 많이 소모합니다. iSCSI에서, 데이터 처리 작업(TCP, SCSI 레벨)의 대부분은 소프트웨어에 의해 처리되며 (하드웨어에 전적으로 의존하는) Fiber Channel에 비해 훨씬 느린 성능을 보입니다. 또 SCSI 커맨드를 iSCSI 트랜잭션으로 매핑하는 과정에서 과도한 오버헤드가 발생합니다. 이 때문에 SCSI 소프트웨어의 오버헤드를 줄이기 위한 대안으로서 TCP/IP, iSCSI 프로세싱을 담당하는 하드웨어 어댑터가 사용되기도 합니다. 이 어댑터를 *iSCSI Host Bus Adaptor (HBA)* 또는 *TCP Offload Engine (TOE)* 카드라 부르기도 합니다. 또 이와 별도로 10 Gigabit Ethernet도 현실화된 테크놀로지로 각광받고 있습니다.

새로운 기술이 모두 그러하듯, iSCSI에도 여러 가지 관련 기술 용어와 약어들이 존재합니다. 본 문서에서는 iSCSI 이니시에이터와 iSCSI 타겟의 차이점만을 이해하는 것이 중요합니다.

iSCSI 이니시에이터(Initiator) . iSCSI 이니시에이터는 서버(iSCSI 타겟)에 연결하고 서버가 제공하는 서비스를 요청하는 클라이언트 디바이스를 의미합니다. iSCSI 이니시에이터 소프트웨어는 각 Oracle RAC 노드 (linux1, linux2)상에 설치되어 있어야 합니다.

iSCSI 이니시에이터는 소프트웨어 또는 하드웨어로 구현될 수 있습니다. 소프트웨어 iSCSI 이니시에이터는 대부분의 운영 체제 플랫폼을 기반으로 구현되어 있습니다. 본 문서에서는 Linux-iSCSI 프로젝트의 일환으로 제공되는 *iscsi-initiator-utils* RPM에 포함된 Linux-iSCSI 소프트웨어 드라이버를

사용하게 될 것입니다. iSCSI 소프트웨어 이니시에이터는 일반적으로 Gigabit Ethernet 카드와 같은 표준 네트워크 인터페이스 카드(NIC)와 함께 사용됩니다. 하드웨어 이니시에이터인 iSCSI HBA(TCP Offload Engine(TOE) 카드)는 SCSI ASIC을 보드에 내장한 특수한 형태의 Ethernet 카드로 볼 수 있으며, 시스템 CPU를 대신하여 TCP, SCSI 커맨드를 처리하는 역할을 담당합니다. iSCSI HBA는 Adaptec, Alacritech, Intel, QLogic 등의 벤더를 통해 구현되고 있습니다.

iSCSI 타겟 (Target). iSCSI 타겟은 iSCSI 네트워크의 "서버" 컴포넌트입니다. iSCSI 타겟은 일반적으로 이니시에이터의 요청에 응답하는 스토리지 디바이스를 의미합니다. 본 문서에서는 openfiler1 노드가 iSCSI 타겟으로 사용될 것입니다.

지금까지 iSCSI에 대해 장황한 설명을 했습니다. 그렇다면 iSCSI의 등장이 Fiber Channel의 종말을 의미하는 것일까요? 그럴 가능성은 별로 없습니다. Fiber Channel은 오랜 세월 동안 뛰어난 성능, 유연성, 안정성을 입증해 온 기술입니다. 고성능 스토리지, 매우 복잡한 연결 환경, 미션 크리티컬 수준의 안정성을 요구하는 고객들은 앞으로도 Fiber Channel 기술을 계속적으로 선택하게 될 것입니다.

이 섹션을 마무리하기 전에, 다양한 디스크 인터페이스와 네트워크 테크놀로지의 속도를 비교한 자료를 소개하는 것이 적절해 보입니다. 각각의 인터페이스별로 최대 전송 속도를 초 당 킬로비트(kb), 킬로바이트(KB), 메가비트(Mb), 메가바이트(MB) 단위로 표시하였습니다. 일반적으로 사용되는 용례는 회색으로 표시되었습니다.

디스크 인터페이스 / 네트워크	속도				
	Kb	KB	Mb	MB	Gb
Serial	115	14.375	0.115	0.014	
Parallel (표준)	920	115	0.92	0.115	
10Base-T Ethernet			10	1.25	
IEEE 802.11b wireless Wi-Fi (2.4 GHz band)			11	1.375	
USB 1.1			12	1.5	
Parallel (ECP/EPP)			24	3	
SCSI-1			40	5	
IEEE 802.11g wireless WLAN (2.4 GHz band)			54	6.75	
SCSI-2 (Fast SCSI / Fast Narrow SCSI)			80	10	
100Base-T Ethernet (Fast Ethernet)			100	12.5	
ATA/100 (parallel)			100	12.5	
IDE			133.6	16.7	
Fast Wide SCSI (Wide SCSI)			160	20	
Ultra SCSI (SCSI-3 / Fast-20 / Ultra Narrow)			160	20	
Ultra IDE			264	33	
Wide Ultra SCSI (Fast Wide 20)			320	40	
Ultra2 SCSI			320	40	
FireWire 400 - (IEEE1394a)			400	50	
USB 2.0			480	60	
Wide Ultra2 SCSI			640	80	
Ultra3 SCSI			640	80	
FireWire 800 - (IEEE1394b)			800	100	
Gigabit Ethernet			1000	125	1
Serial ATA I - (SATA I)			1200	150	1.2
Wide Ultra3 SCSI			1280	160	1.28
Ultra160 SCSI			1280	160	1.28
Serial ATA II - (SATA II)			2400	300	2.4
Ultra320 SCSI			2560	320	2.56
FC-AL Fibre Channel			3200	400	3.2
Serial ATA III - (SATA III)			4800	600	4.8
10G Ethernet (IEEE 802.3ae)			10000	1250	10

5. 하드웨어 및 비용

Oracle10g RAC 환경을 구현하기 위한 하드웨어로, 세 대의 Linux 서버(두 대의 Oracle RAC 노드와 한 대의 Network Storage Server), 그리고 인터넷 또는 일반 컴퓨터 매장에서 구입 가능한 컴포넌트들이 사용되었습니다.

Oracle RAC Node 1 - (linux1)	
<p>Dimension 2400 Series</p> <ul style="list-style-type: none"> ● Intel(R) Pentium(R) 4 프로세서 at 2.80GHz ● 1GB DDR SDRAM (at 333MHz) ● 40GB 7200 RPM 내장형 하드 드라이브 ● Integrated Intel 3D AGP 그래픽 카드 ● 통합 10/100 이더넷 ● CDROM (최대 48 배속) ● 3.5" 플로피 ● 모니터 (기존 장비 활용) ● USB 마우스, 키보드 	US\$620
<p>1 - 이더넷 LAN 카드 linux2와 Openfiler 네트워크 스토리지를 연결하기 위한 RAC 인터커넥트에 사용. Oracle RAC 노드로 사용되는 두 대의 Linux 서버에는 각각 2 조의 NIC 어댑터가 장착됩니다. Dell Dimension 서버에 내장된 통합 100/100 이더넷 어댑터는 퍼블릭 네트워크에 연결하는데 사용됩니다. 두 번째 NIC 어댑터는 사설 네트워크(RAC 인터커넥트와 Openfiler 네트워크 스토리지를 위해 사용됩니다. 이때 사설 네트워크의 네트워크 스위치가 지원하는 최대 데이터 전송 속도와 호환 가능한 NIC 어댑터를 사용해야 합니다.</p> <p>10/100 Ethernet Linksys 10/100 Mbps - (LNE100TX)</p> <p>Gigabit Ethernet NETGEAR 10/100/1000Mbps PCI Network Adapter - (GA311)</p>	US\$20
Oracle RAC Node 2 - (linux2)	
<p>Dimension 2400 Series</p> <ul style="list-style-type: none"> ● Intel(R) Pentium(R) 4 프로세서 2.80GHz ● 1GB DDR SDRAM (at 333MHz) ● 40GB 7200 RPM 내장형 하드 드라이브 ● Integrated Intel 3D AGP 그래픽 카드 ● 통합 10/100 이더넷 ● CD-ROM (최대 48배속) ● 3.5" 플로피 ● 모니터 (기존 장비 활용) ● USB 마우스, 키보드 	US\$620
<p>1 - 이더넷 LAN 카드 linux1과 Openfiler 네트워크 스토리지를 연결하기 위한 RAC 인터커넥트에 사용.</p> <p>Oracle RAC 노드로 사용되는 두 대의 Linux 서버에는 각각 2 조의 NIC 어댑터가 장착됩니다. Dell Dimension 서버에 내장된 통합 100/100 이더넷 어댑터는 퍼블릭 네트워크에 연결하는데 사용됩니다. 두 번째 NIC 어댑터는 사설 네트워크(RAC 인터커넥트와 Openfiler 네트워크 스토리지를 위해 사용됩니다. 이때 사설 네트워크의 네트워크 스위치가 지원하는 최대 데이터 전송 속도와 호환 가능한 NIC 어댑터를 사용해야 합니다.</p> <p>10/100 Ethernet Linksys 10/100 Mbps - (LNE100TX)</p> <p>Gigabit Ethernet NETGEAR 10/100/1000Mbps PCI Network Adapter - (GA311)</p>	US\$20
Network Storage Server - (openfiler1)	
<p>Clone / Pentium 4</p> <ul style="list-style-type: none"> ● Intel(R) Pentium(R) 4 프로세서 1.80GHz ● 1GB DDR SDRAM (at 333MHz) ● 40GB 7200 RPM 내장형 하드 드라이브 ● NVIDIA GeForce FX 5200 / AGP 그래픽 카드 ● 통합 10/100 이더넷 	

<ul style="list-style-type: none"> ● 4x 통합 USB 2.0 포트 ● CD-ROM (최대 48배속) ● 3.5" 플로피 ● 모니터 (기존 장비 활용) ● USB 마우스, 키보드 	US\$500
<p>1 - 이더넷 LAN 카드 사설 네트워크의 네트워크 스토리지를 위해 사용.</p> <p>Network Storage Server(Openfiler 서버)는 2 조의 NIC 어댑터를 포함하고 있습니다. Clone / Pentium 4 머신에 내장된 통합 100/100 이더넷 어댑터는 퍼블릭 네트워크에 연결하는데 사용 됩니다. 두 번째 NIC 어댑터는 사설 네트워크(Openfiler 네트워크 스토리지)를 위해 사용됩니다. 이때 사설 네트워크의 네트워크 스위치가 지원하는 최대 데이터 전송 속도와 호환 가능한 NIC 어댑터를 사용해야 합니다.</p> <p>10/100 Ethernet Linksys 10/100 Mbps - (LNE100TX)</p> <p>Gigabit Ethernet NETGEAR 10/100/1000Mbps PCI Network Adapter - (GA311)</p>	US\$20
기타 컴포넌트	
<p>스토리지 디바이스 - 외장형 하드 드라이브 데이터베이스 스토리지를 위해서는 USB 2.0 인터페이스를 통해 Openfiler 서버에 연결된 외장형 Maxtor OneTouch III (500 GB) 드라이브를 사용하였습니다. 본 문서에서는 Openfiler 서버를 이용하여 이 디스크를 iSCSI 기반 스토리지로 설정한 후, Oracle10g RAC 구성에서 Oracle Clusterware와 Oracle ASM 볼륨의 공유 파일을 저장하기 위해 사용합니다 네트워크 스토리지 서버에서 인식하는데 문제가 없다면, 또 여유 공간이 충분하다면 어떤 종류의 하드 디스크(내장형 또는 외장형)를 사용하더라도 무방합니다.</p> <p>Maxtor OneTouch III - 500GB FireWire 800/FireWire 400/USB 2.0 Drive - (F01W500) 이번 테스트를 위해 필자가 구입한 디스크의 제원이 다음과 같습니다:</p> <p> 벤더: Maxtor 모델: OneTouch III Mfg. Part No. / KIT No.: F01W500 용량: 500 GB 캐시 버퍼: 16 MB 회전 속도(rpm): 7200 RPM 인터페이스 전송률: FireWire 800 - 800 Mbits/sec FireWire 400 - 400 Mbits/sec USB 2.0 - 480 Mbits/sec</p> <p>"Combo" 인터페이스: IEEE 1394b - FireWire 800 IEEE 1394a - FireWire 400 USB 2.0 / USB 1.1</p>	US\$320
<p>1 - 이더넷 스위치 linux1-priv와 linux2-priv의 인터커넥트를 위해 사용됩니다. 또 이 스위치를 통해 Openfiler의 네트워크 스토리지 트래픽이 전달됩니다. 비용 절감을 위해, 필자는 사설 네트워크에 10/100 MB 이더넷 스위치와 10/100 Mb 이더넷 카드를 사용하였습니다. 하지만 실제 환경에서는 Gigabit Ethernet 스위치와 1 GB 이더넷 카드가 강력하게 권장됩니다.</p> <p>10/100 Ethernet Linksys EtherFast 10/100 5-port Ethernet Switch - (EZXS55W)</p> <p>Gigabit Ethernet D-Link 8-port 10/100/1000 Desktop Switch - (DGS-2208)</p>	US\$25
<p>6 - 네트워크 케이블</p> <p>Category 5e 패치 케이블 - (linux1을 퍼블릭 네트워크에 연결) US\$5 Category 5e 패치 케이블 - (linux2를 퍼블릭 네트워크에 연결) US\$5 Category 5e 패치 케이블 - (openfiler1을 퍼블릭 네트워크에 연결) US\$5 Category 5e 패치 케이블 - (linux1을 인터커넥트 이더넷 스위치에 연결) US\$5 Category 5e 패치 케이블 - (linux2을 인터커넥트 이더넷 스위치에 연결) US\$5 Category 5e 패치 케이블 - (openfiler1을 인터커넥트 이더넷 스위치에 연결) US\$5</p>	
합계	US\$2,175

이제 설치 작업을 시작할 차례입니다. 지금까지는 본 예제에서 사용할 하드웨어에 대해 설명했습니다.

이제 환경을 개념적으로 이해해 봅시다. (아래 이미지를 클릭하면 큰 그림을 보실 수 있습니다.)

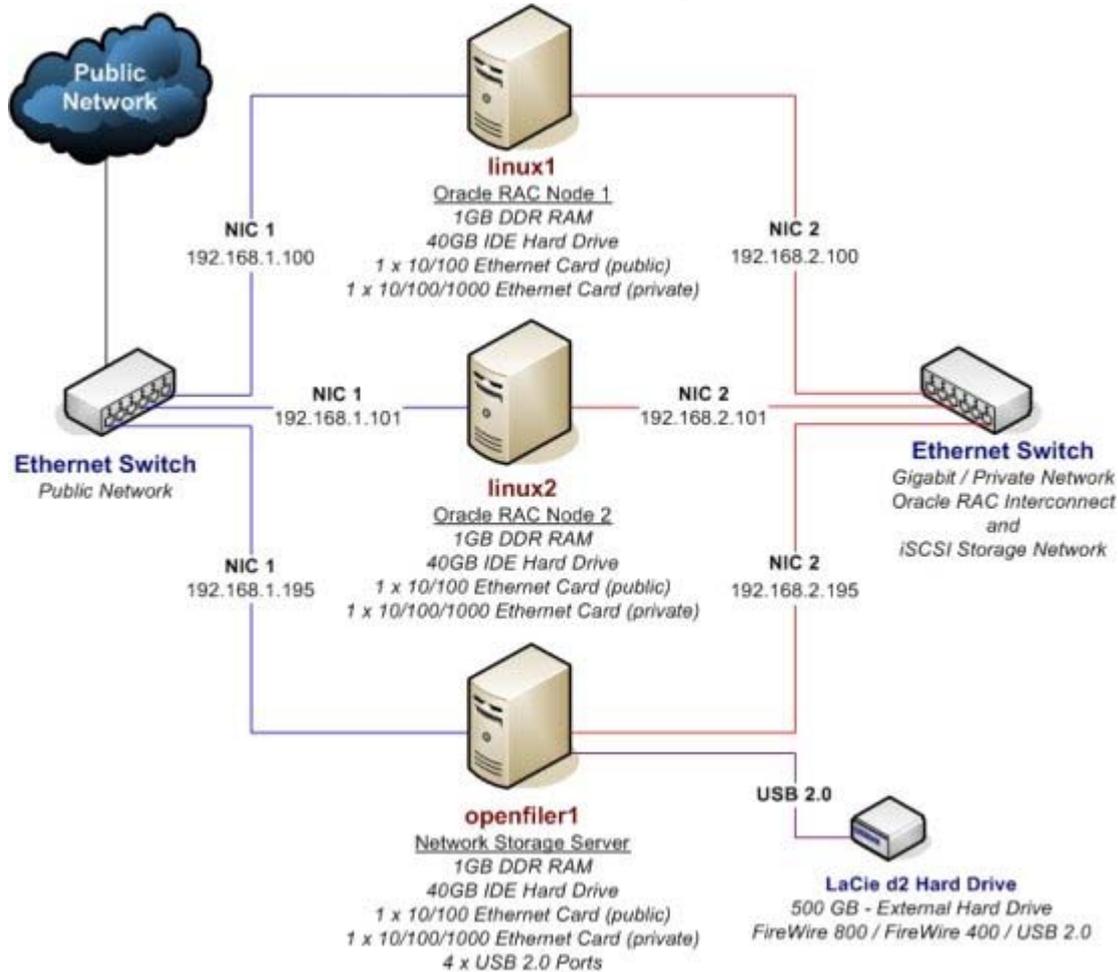


그림 1 아키텍처

설치 작업을 본격적으로 시작하기 전에, 본 문서에서 설명되는 작업의 대부분이 두 대의 Oracle RAC 노드에서 동시 수행되어야 함을 명심하시기 바랍니다. 각 섹션의 시작 부분에서 작업이 양쪽 Oracle RAC 노드에 모두 실행되어야 하는지 또는 네트워크 스토리지 서버(openfiler1)에서 실행되어야 하는지 설명하도록 하겠습니다.

6. Linux 운영체제의 설치

아래 작업을 클러스터의 모든 Oracle RAC 노드에서 실행합니다!

이 섹션에서는 Linux 운영체제를 설치하는 과정을 단계별로 설명합니다. 본 문서에서 설명된 내용은 Oracle's Enterprise Linux Release 4 Update 5. 운영 체제를 기준으로 하고 있습니다.

좀 더 자세한 설치 과정은 Red Hat Linux가 제공하는 manuals에서 참고할 수 있습니다. 하지만, 아래에 제시된 설명을 기반으로 시스템을 구성해야 함을 잊지 마시기 바랍니다.

두 노드에 Linux 운영체제를 설치하기 전에 먼저 2 조의 NIC 인터페이스 카드를 설치해야 합니다.

Enterprise Linux Release 4 Update 5의 ISO 이미지를 아래 링크에서 다운로드합니다:

Oracle E-Delivery Web site for Enterprise Linux

V10378-01_1of4.zip (572 MB)

V10378-01_2of4.zip (619 MB)

V10378-01_3of4.zip (621 MB)

V10378-01_4of4.zip (269 MB)

Enterprise Linux software를 다운로드 후 각 파일을 압축을 풀어주십시오. ISO images를 CD로 굽습니다.

Enterprise-R4-U5-i386-disc1.iso

Enterprise-R4-U5-i386-disc2.iso

Enterprise-R4-U5-i386-disc3.iso

Enterprise-R4-U5-i386-disc4.iso

MS Windows 머신에서 ISO 파일을 다운로드한 경우, 이미지를 CD로 굽기 위한 다양한 옵션을 활용할 수 있습니다. 본 문서에서는 ISO 이미지를 CD를 굽기 위한 소프트웨어와 사용 방법에 대해 여러분이 잘 알고 있는 것으로 가정하겠습니다. 방법을 잘 모르거나, 이미지를 CD로 굽는데 필요한 소프트웨어

가 없다면 아래 두 가지 소프트웨어 패키지 중 하나를 사용할 것을 추천 드립니다:

UltraISO

Magic ISO Maker

Cent OS 이미지(ISO 파일)를 downloading하여 CD로 구운 다음, CentOS Disk #1을 첫 번째 서버 (linux1)에 넣고 파워를 켭니다. 그런 다음 아래 설명된 방법 대로 설치 스크린 상에서 답변을 입력합니다. 첫 번째 노드의 Linux 설치 작업을 끝낸 다음에는 두 번째 노드에서 같은 작업을 반복하되, 노드명을 linux1에서 linux2로 변경하고 다른 IP 주소를 적용해야 함을 명심해야 합니다.

Boot Screen

첫 번째 설치 화면(CentOS Enterprise Linux 부트 스크린)의 프롬프트에서 [Enter] 키를 눌러 설치 프로세스를 시작합니다

Media Test

CD 미디어를 테스트할 것인지 묻는 질문에 대해 [Skip]을 선택하고 [Enter] 키를 누릅니다. (미디어에 에러가 있다면 CD 버닝 소프트웨어를 통해 이미 경고 메시지가 표시 되었을 것입니다.) 몇 초 동안 비디오 카드, 모니터, 마우스 등의 인식 작업이 수행되고, 인스톨러는 GUI 모드로 전환됩니다.

Welcome to CentOS Enterprise Linux

[Next]를 클릭하여 다음 단계로 넘어갑니다.

Language / Keyboard Selection

언어, 키보드 설정을 위한 화면입니다. 적절한 구성을 선택합니다.

Installation Type

[Custom] 옵션을 선택하고 [Next]를 클릭합니다.

Disk Partitioning Setup

[Automatically partition]을 선택하고 [Next]를 클릭합니다.

서버에 이전에 설치한 Linux 운영체제가 존재하는 경우, 오래된 파티션을 “제거(remove)” 또는 “보존(keep)”할 것인지 묻는 스크린이 표시됩니다. [Remove all partitions on this system] 옵션을 선택합니다. 또 설치 대상에 [hda] 드라이브가 선택되었는지 확인합니다. [Review (and modify if needed) the partitions created] 옵션도 함께 체크하도록 합니다. [Next]를 클릭하여 다음 단계로 진행합니다. 정말로 모든 파티션을 삭제할 것인지 묻는 대화 창이 표시됩니다. [Yes]를 클릭합니다.

Partitioning

이 단계에서 인스톨러가 자동으로 선택한 디스크 파티션을 조회하고, 필요한 경우 수정할 수 있습니다. 대부분의 경우 인스톨러는 /boot를 위해 100MB, swap 공간을 위해 메모리 용량의 두 배를 할당하고, 나머지를 root(/) 파티션에 할당합니다. 스왑 공간의 크기는 최소 1GB로 설정하는 것이 좋습니다. 여기에서는 디폴트 설정을 그대로 사용하기로 합니다. (1GB RAM이 설치되어 있으므로 스왑 공간으로는 2GB가 할당됩니다.)

RHEL 4에서부터, 인스톨러는 LVM(Logical Volume Manager)를 사용하여 디스크 설정 작업을 수행합니다. 예를 들어, 첫 번째 하드 드라이브(예의 경우 /dev/hda)를 두 개의 파티션으로 나눌 때, 먼저 boot 파티션(/dev/hda1)을 할당하고 디스크의 나머지 영역을 VolGroup00(/dev/hda2)이라는 이름의 LVM에 할당합니다. LVM Volume Group(VolGroup 00)은 다시 두 개의 LVM 파티션(root 파일 시스템과 swap)으로 파티셔닝 됩니다. 여기에서는 스왑 공간이 최소 1GB를 넘는 지의 여부만 확인하면 충분합니다. 필자의 시스템에는 1GB RAM이 설치되어 있으므로, 인스톨러는 2GB의 스왑 공간을 생성하였습니다. 따라서 디폴트 디스크 레이아웃을 그대로 적용해도 아무런 문제가 없을 것입니다.

Boot Loader Configuration

인스톨러는 디폴트로 GRUB 부트 로더를 사용합니다. 디폴트 값을 선택하여 GRUB 부트 로더를 사용하도록 하고 [Next]를 클릭합니다.

Network Configuration

앞에서 설명한 것처럼, 운영체제를 설치하기 전에 각 서버에 NIC 카드가 설치되어 있어야 합니다. 이 화면에서는 두 조의 NIC 카드가 모두 정상적으로 인식되고 있어야 합니다

먼저, 각 NIC 카드에 대해 [Active on boot] 항목이 체크되어 있는지 확인합니다. (인스톨러가 eth1을 활성화하지 않았을 수도 있습니다.)

두 번째로, eth0과 eth1에 대해 [Edit] 작업을 수행합니다. eth0과 eth1에 서로 다른 IP 주소를 할당하고, 가능하다면 eth1(인터넷)과 eth0(퍼블릭 네트워크)에 서로 다른 서브넷을 할당합니다.

eth0:

- [Configure using DHCP] 옵션의 체크를 해제합니다.
- [Activate on boot]를 체크된 상태로 둡니다.
- IP 주소: 192.168.1.100
- 넷마스크: 255.255.255.0

eth1:

- [Configure using DHCP] 옵션의 체크를 해제합니다.
- [Activate on boot]를 체크된 상태로 둡니다.

- IP 주소: 192.168.2.100
- 넷마스크: 255.255.255.0

hostname은 수동으로 설정합니다. 첫 번째 노드는 "linux1"으로, 두 번째 노드는 "linux2"로 설정합니다. 게이트웨이와 DNS 서버를 설정하고 다음 단계로 넘어갑니다.

Firewall

[No firewall]이 선택되어 있는지 확인하고 [Next]를 클릭합니다. 방화벽을 설정하지 않았다는 경고 메시지가 뜰 수도 있습니다. 경고 메시지가 떴다면, [Proceed]를 클릭하여 다음 단계로 진행합니다.

Additional Language Support/Time Zone

추가 언어 지원 및 시간대 설정을 위한 화면이 표시됩니다. 대부분의 경우, 디폴트 설정을 그대로 사용하는 것이 권장됩니다.

Set Root Password

root 패스워드를 선택하고 [Next]를 클릭합니다.

Package Group Selection

스크린 하단으로 스크롤한 뒤 "Miscellaneous" 항목에서 [Everything]을 선택합니다. [Next]를 클릭하여 다음 단계로 진행합니다.

오라클을 설치하기 전에 모든 Linux 패키지를 미리 설치해 두어야 하는 것은 아니라는 점을 참고하시기 바랍니다. 여기에서는 예제를 최대한 단순화하기 위해 모든 패키지를 한꺼번에 설치하는 방법을 사용했습니다. 오라클 소프트웨어를 성공적으로 설치하기 위해 필요한 패키지에 대한 상세한 정보는 Section 19("Oracle10g Release 2 설치 전 준비 작업")에서 확인하실 수 있습니다.

일부 RHEL4 배포판의 경우, 디폴트로 "Package Group Selection" 스크린이 표시됩니다. 이 화면에서는 "Install default software packages" 또는 "Customize software packages to be installed" 중 하나를 선택해야 합니다. "Customize software packages to be installed" 옵션을 선택하고 [Next]를 클릭하여 다음 단계로 진행하면 "Package Group Selection" 스크린이 표시됩니다. 여기에서 스크린 하단으로 스크롤한 후 "Miscellaneous" 항목에서 [Everything]을 선택합니다. [Next]를 클릭하여 다음 단계로 진행합니다.

About to Install

설치 전 최종 확인을 위한 스크린이 표시됩니다. [Continue]를 클릭하여 설치 작업을 시작합니다. 설치 과정에서 Disk #2, Disk #3, Disk #4를 바꿔 넣으라는 프롬프트가 표시될 것입니다.

CentOS 4.2의 경우, Disk #2, Disk #3, Disk #4, Disk #1, 그리고 다시 Disk #4의 순서로 디스크를 교환해 주어야 합니다.

Graphical Interface (X) Configuration

(CentOS 4.2를 제외한) 대부분의 RHEL4 배포판의 경우, 설치가 완료된 후 인스톨러에 의해 비디오 하드웨어의 인식이 시도됩니다. 인스톨러가 X Windows 서버 사용에 필요한 비디오 하드웨어(그래픽 카드와 모니터)를 올바르게 인식했는지 확인합니다. 그리고 다음 단계에서는 X Window 설정 작업이 수행됩니다.

Congratulations

이것으로 이제 첫 번째 노드(linux1)의 CentOS Enterprise Linux 운영체제 설치 작업이 성공적으로 완료되었습니다. 인스톨러가 CD-ROM 드라이브에서 CD를 자동으로 꺼냅니다. CD를 꺼낸 후 [Reboot]를 클릭하여 시스템을 리부트 합니다.

Linux 시스템이 처음 부팅되면, 새로운 Welcome 스크린이 표시됩니다. 마법사를 이용하여 날짜와 시간을 설정하고, 사용자 계정을 추가하고, 사운드 카드를 테스트하고, CD를 이용한 추가 설치 작업을 수행할 수 있습니다 여기에서는 시간/날짜 확인 이외의 다른 작업에는 신경 쓰지 않아도 됩니다. (단 CentOS 4.x의 경우에는 모니터/디스플레이 설정도 확인해 주어야 합니다.) 모든 작업이 성공적으로 완료되었다면 로그인 스크린이 표시되는 것을 확인하실 수 있을 것입니다.

두 번째 노드에서 동일한 설치 작업을 반복

첫 번째 노드에서 Linux 운영체제의 설치를 완료한 후, 두 번째 노드(linux2)에서 같은 작업을 반복합니다. 이때 서버명과 네트워크 설정이 달라지는 점에 주의하시기 바랍니다. linux2의 설정 예가 아래와 같습니다:

먼저, 각 NIC 카드에 대해 [Active on boot] 항목이 체크되어 있는지 확인합니다. (인스톨러가 eth1을 활성화하지 않았을 수도 있습니다.)

두 번째로, eth0과 eth1에 대해 [Edit] 작업을 수행합니다:

eth0:

- [Configure using DHCP] 옵션의 체크를 해제합니다.
- [Activate on boot]를 체크된 상태로 둡니다.
- IP 주소: 192.168.1.101
- 넷마스크: 255.255.255.0

eth1:

- [Configure using DHCP] 옵션의 체크를 해제합니다.
- [Activate on boot]를 체크된 상태로 둡니다.
- IP 주소: 192.168.2.101

- 넷마스크: 255.255.255.0

hostname은 수동으로 설정합니다. 필자는 두 번째 노드의 이름으로 "linux2"를 사용하였습니다. 마지막으로 게이트웨이와 DNS 서버를 설정하고 다음 단계로 넘어갑니다.

7. 네트워크 설정

아래 작업을 클러스터의 모든 Oracle RAC 노드에서 실행합니다!

참고: 이 섹션에서는 RAC 환경에 필수적인 설정 작업을 설명하고 있습니다. Linux 설치 과정에서 기본적인 네트워크 설정을 완료한 상태라 해도, 이 단계를 생략해서는 안됩니다.

네트워크 설정 개요

Linux OS의 설치 과정에서 각 노드의 IP 주소와 호스트네임 설정을 이미 완료한 상태입니다. 이제 /etc/hosts 파일을 설정하고 인터커넥트를 위한 네트워크 설정 작업을 수행할 차례입니다.

Oracle RAC의 각 노드는 퍼블릭 네트워크를 위한 정적 IP 주소와 노드 간의 직접 연결(인터커넥트)을 위한 정적 IP 주소를 갖습니다. (절대로 퍼블릭 IP 주소 또는 인터커넥트를 위해 DHCP를 사용해서는 안됩니다!) 프라이빗 인터커넥트(private interconnect)의 사설 IP 주소는 오라클 데이터베이스가 Cluster Manager, Cache Fusion 데이터, 네트워크 스토리지 서버(Openfiler) 데이터를 전송하는 용도로만 사용됩니다. 인터커넥트에 퍼블릭 네트워크를 사용하는 것도 가능하지만, 데이터베이스 성능이 저하될 수 있으므로 권장되지 않습니다. (Cache Fusion과 Cluster Manager 트래픽을 위한 대역폭이 제약될 수 있습니다.) 운영 환경에서는 인터커넥트를 위해 최소한 기가바이트 급의 네트워크를 이용한 전용 네트워크를 구성해야 합니다. 네트워크 스토리지 서버(Openfiler)도 별도의 기가바이트 네트워크에 구성해야 합니다.

Public/Private 네트워크의 설정

퍼블릭 네트워크와 인터커넥트를 위한 사설 네트워크의 설정 방법이 아래와 같습니다. RHEL 4의 "Network Configuration" 프로그램을 사용하면 네트워크 설정을 쉽게 완료할 수 있습니다. Network Configuration 프로그램을 시작하려면 root 계정으로 로그인한 뒤 커맨드 라인에서 다음과 같이 실행합니다:

```
# su -
# /usr/bin/system-config-network &
```

Network Configuration 프로그램을 통해 NIC 디바이스와/etc/hosts파일의 설정을 변경합니다. 두 가지 작업 모두 Network Configuration GUI를 통해 수행할 수 있습니다. 두 노드의/etc/hosts 설정은 동일합니다.

아래와 같이 설정 작업을 수행합니다:

Oracle RAC Node 1 - (linux1)				
디바이스	IP 주소	서브넷	게이트웨이	용도
eth0	192.168.1.100	255.255.255.0	192.168.1.1	linux1 을 퍼블릭 네트워크에 연결
eth1	192.168.2.100	255.255.255.0		linux1 (interconnect)과 linux2 (linux2-priv)의 직접연결
/etc/hosts				
127.0.0.1 localhost loopback				
# Public Network - (eth0)				
192.168.1.100 linux1				
192.168.1.101 linux2				
# Private Interconnect - (eth1)				
192.168.2.100 linux1-priv				
192.168.2.101 linux2-priv				
# Public Virtual IP (VIP) addresses for - (eth0)				
192.168.1.200 linux1-vip				
192.168.1.201 linux2-vip				
Oracle RAC Node 2 - (linux2)				
디바이스	IP 주소	서브넷	게이트웨이	용도

eth0	192.168.1.101	255.255.255.0	192.168.1.1	linux2를 퍼블릭 네트워크에 연결
eth1	192.168.2.101	255.255.255.0		linux2 (interconnect)과 linux1 (linux1-priv)의 직접연결

```

/etc/hosts
127.0.0.1 localhost loopback
# Public Network - (eth0)
192.168.1.100 linux1
192.168.1.101 linux2
# Private Interconnect - (eth1)
192.168.2.100 linux1-priv
192.168.2.101 linux2-priv
# Public Virtual IP (VIP) addresses for - (eth0)
192.168.1.200 linux1-vip
192.168.1.201 linux2-vip
    
```

가상 IP 주소는 두 Oracle RAC 노드의 /etc/hosts 파일에만 정의된다는 점에 주의하시기 바랍니다. 퍼블릭 가상 IP 주소는 Oracle Universal Installer 실행 과정에서 Virtual Internet Protocol Configuration Assistant (VIPCA)에 의해 자동으로 설정됩니다. 모든 가상 IP 주소는 `svctl start nodeapps -n <node_name>` 커맨드가 실행될 때 활성화됩니다. 바로 이 주소가 클라이언트 `tnsnames.ora` 파일에 Host Name/IP Address로 설정 됩니다(자세한 내용은 뒤에서 설명합니다). 첫 번째 Oracle RAC 노드(linux1)를 위한 설정 화면이 아래와 같습니다. 양쪽 노드에 네트워크 설정을 해 주는 것을 잊지 마시지 바랍니다.

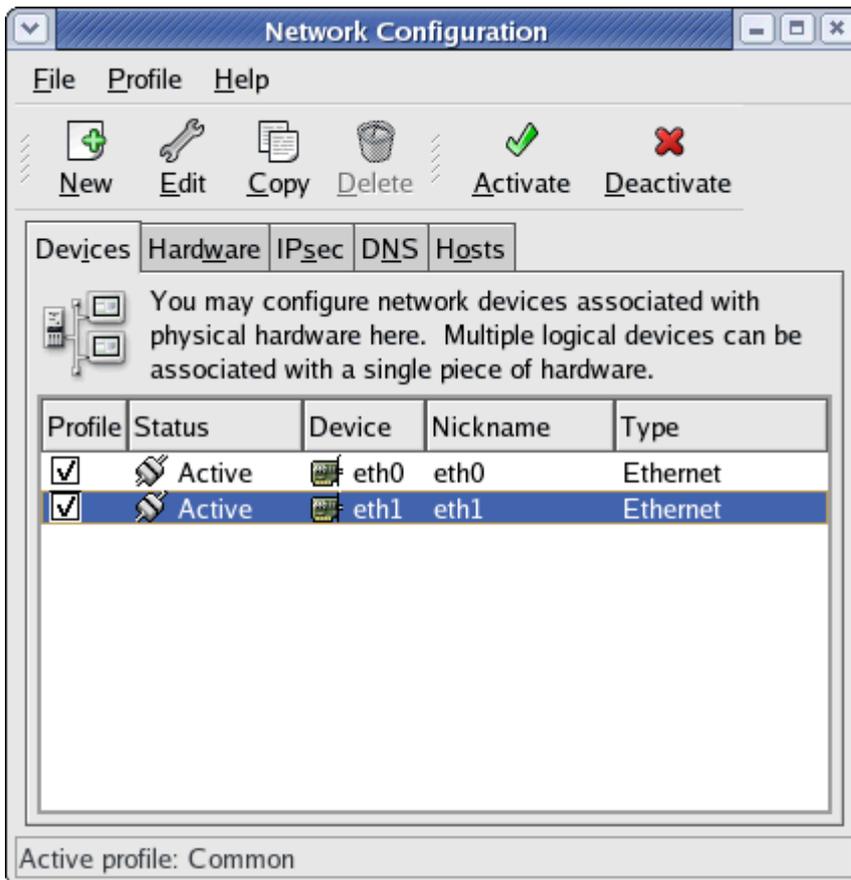


그림 2 Network Configuration 스크린, Node 1 (linux1)

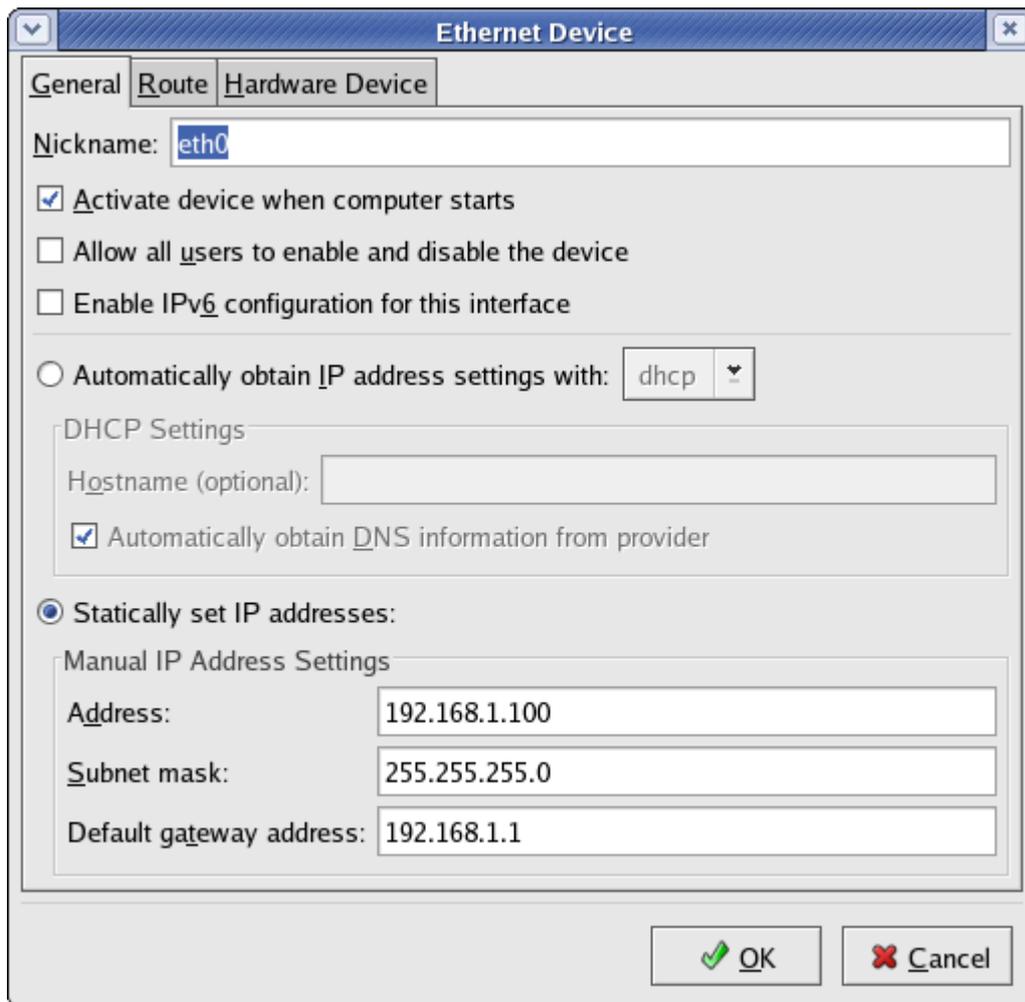


그림 3 Ethernet Device 스크린, eth0 (linux1)

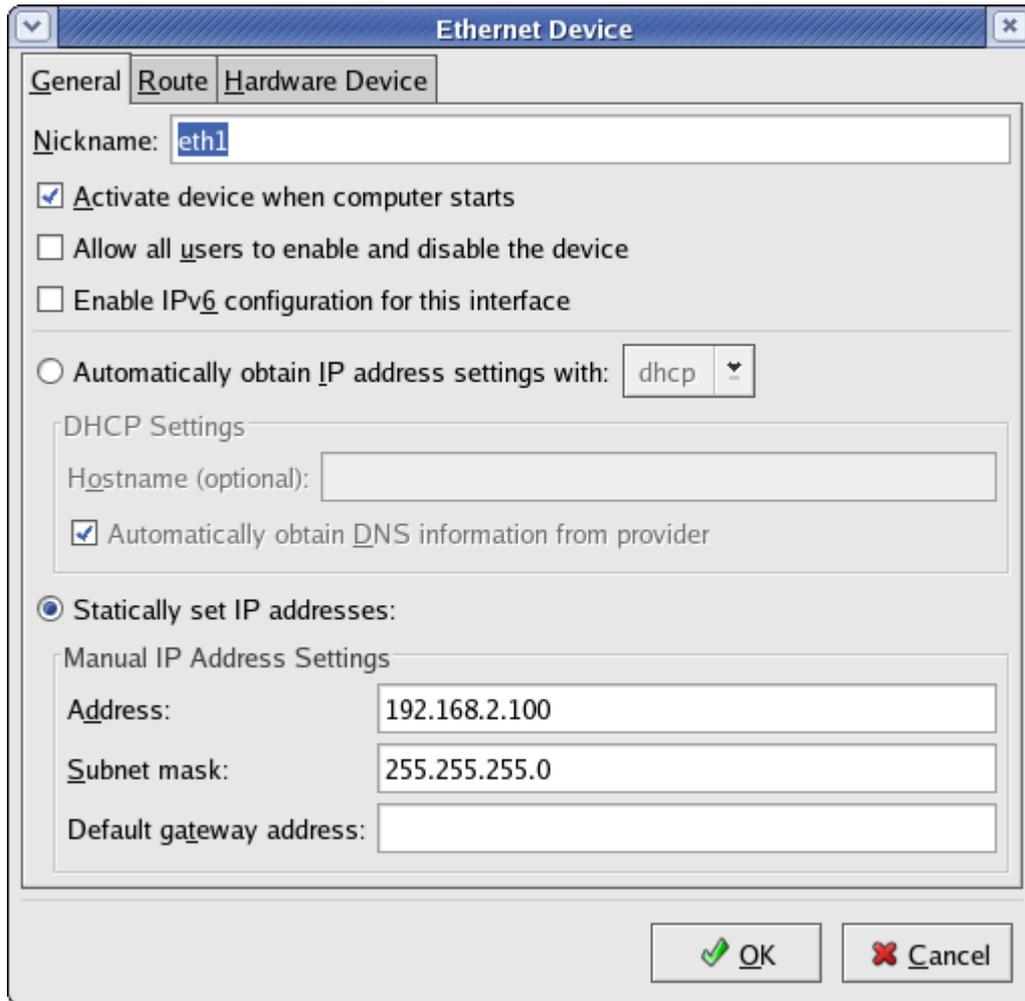


그림 4 Ethernet Device 스크린, eth1 (linux1)

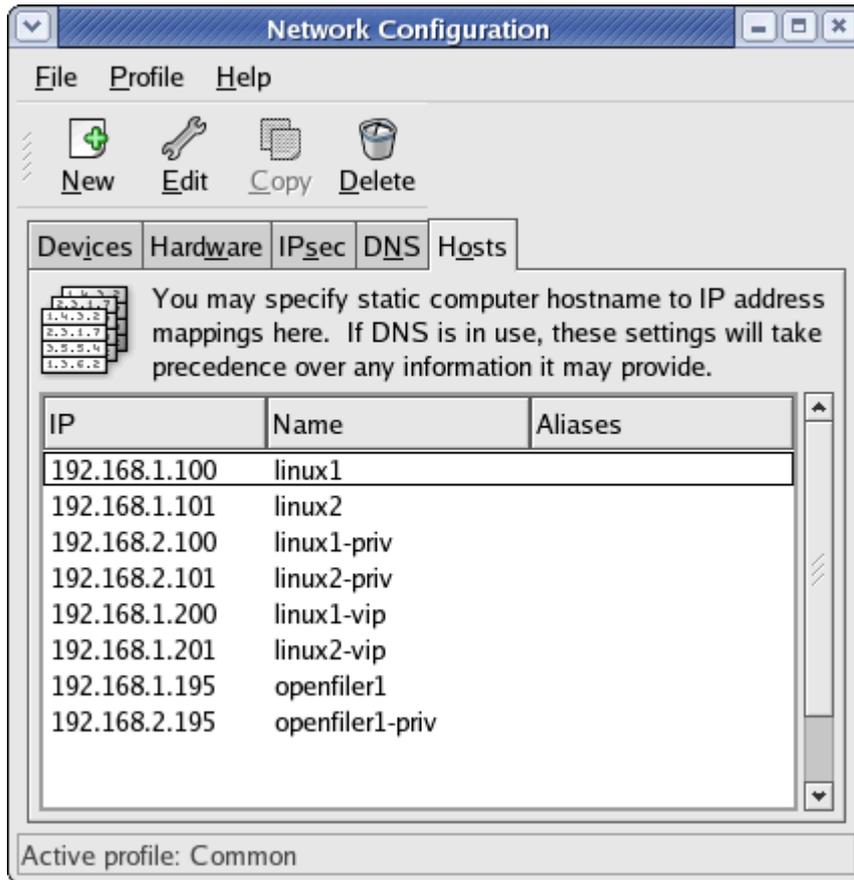


그림 5: Network Configuration 스크린, /etc/hosts (linux1)

네트워크 설정 작업을 완료한 뒤 ifconfig 커맨드를 이용하여 설정 내역을 확인할 수 있습니다. linux1에서 확인한 설정 정보가 아래와 같습니다. :

```
$ /sbin/ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:0D:56:FC:39:EC
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::20d:56ff:fe80:39ec/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:835 errors:0 dropped:0 overruns:0 frame:0
TX packets:1983 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:705714 (689.1 KiB) TX bytes:176892 (172.7 KiB)
Interrupt:3
eth1 Link encap:Ethernet HWaddr 00:0C:41:E8:05:37
inet addr:192.168.2.100 Bcast:192.168.2.255 Mask:255.255.255.0
inet6 addr: fe80::20c:41ff:fee8:537/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 b) TX bytes:546 (546.0 b)
Interrupt:11 Base address:0xe400
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:5110 errors:0 dropped:0 overruns:0 frame:0
TX packets:5110 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:8276758 (7.8 MiB) TX bytes:8276758 (7.8 MiB)
```

```

sit0 Link encap:IPv6-in-IPv4
NOARP MTU:1480 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:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

```

가상 IP에 대해

10g에서 가상 IP(VIP, Virtual IP)가 필요한 이유는 무엇일까요?

바로 애플리케이션의 가용성 보장을 위해서입니다. 노드에 장애가 발생하면, 노드에 할당된 VIP는 자동으로 다른 노드에 페일오버 됩니다. 이 과정에서 두 가지 작업이 진행됩니다.

1. 새로운 노드가 arp 시그널을 브로드캐스트하고 IP 주소에 연관된 MAC 주소를 공개합니다. 서버가 직접 연결된 상황에서는, 이전의 주소로의 연결하는 과정에서 장애가 발생하게 됩니다
2. 가상 IP를 주소로 하는 패킷이 새로운 노드로 전달되고, 새로운 노드는 클라이언트에 RST 패킷을 회신합니다. 따라서 클라이언트는 즉각적으로 에러를 수신하게 됩니다.

다시 말해, 클라이언트가 다운된 노드에 대해 SQL 구문을 실행한 경우, TCP/IP 타임아웃이 발생할 때까지 오랜 시간 대기하지 않고 클라이언트가 TCP 리셋 신호를 즉각적으로 수신하게 됩니다. SQL 구문을 실행한 경우에는 ORA-3113 에러가 발생하며, 연결 과정에서 tnsnames.ora에 정의된 다른 주소가 사용됩니다.

Transparent Application Failover(TAF)를 사용하는 경우에는 상황이 조금 달라집니다. TAF를 이용하면 ORA-3113에러를 완벽하게 방지하는 것이 가능합니다! TAF에 대해서는 Section 29("Transparent Application Failover - (TAF)")에서 자세히 설명합니다.

VIP를 사용하지 않은 경우, 다운된 노드에 연결된 클라이언트는 10분의 TCP 타임아웃 기간이 완료된 후에야 에러를 확인할 수 있습니다. 따라서 VIP를 이용하지 않고는 효과적인 HA 솔루션을 구현하는 것이 불가능합니다(출처 - Metalink Note 220970.1).

RAC Node Name이 Loopback Address로 등록되지 않았음을 확인

노드 네임 (linux1 or linux2)이 /etc/hosts 파일에 루프백 주소로 등록되지 않았음을 확인합니다. 루프백 주소 항목에 아래와 같이 서버 네임이 등록되어 있는 경우:

```
127.0.0.1 linux1 localhost.localdomain localhost
```

아래와 같이 제거되어야 합니다:

```
127.0.0.1 localhost.localdomain localhost
```

RAC node name이 루프백 주소에 포함되어 있으면, RAC 설치 과정에서 다음과 같은 에러가 발생합니다:

```

ORA-00603: ORACLE server session terminated by fatal error
또는
ORA-29702: error occurred in Cluster Group Service operation

```

네트워크 설정의 조정

Oracle 9.2.0.1 및 이후 버전에서는, Linux의 IPC(inter-process communication, Cache Fusion, Cluster Manager의 RAC 클러스터 인스턴스 간 버퍼 전송 등)를 위한 디폴트 프로토콜로 UDP를 사용합니다.

Oracle strongly suggests to adjust the default and maximum send buffer size (SO_SNDBUF socket option) to 256KB, and the default and maximum receive buffer size (SO_RCVBUF socket option) to 256KB.

Receive buffer는 TCP/UDP에서 수신된 데이터를 (애플리케이션이 read 작업을 수행하기 전까지) 임시로 저장하는데 사용됩니다. TCP/UDP 프로토콜은 receive buffer의 오버플로우를 방지하기 위해 버퍼 사이즈 윈도우를 초과하는 전송을 허용하지 않습니다. 따라서 receive buffer의 용량을 초과하는 데이터그램은 폐기 처리 됩니다.

Default/maximum window size는 /proc파일 시스템에서 변경 가능하며 리부팅하지 않고도 적용이 가능합니다.

```

# su - root
# sysctl -w net.core.rmem_default=262144
net.core.rmem_default = 262144

```

```
# sysctl -w net.core.wmem_default=262144
net.core.wmem_default = 262144
# sysctl -w net.core.rmem_max=262144
net.core.rmem_max = 262144
# sysctl -w net.core.wmem_max=262144
net.core.wmem_max = 262144
```

위는 OS가 실행 중인 상태에서 변경 내용을 적용한 예입니다. 영구적으로 변경사항을 적용하려면 모든 RAC 클러스터 노드의 /etc/sysctl.conf 파일에 다음 라인을 추가해야 합니다:

```
# Default setting in bytes of the socket receive buffer
net.core.rmem_default=262144
# Default setting in bytes of the socket send buffer
net.core.wmem_default=262144
# Maximum socket receive buffer size which may be set by using
# the SO_RCVBUF socket option
net.core.rmem_max=262144
# Maximum socket send buffer size which may be set by using
# the SO_SNDBUF socket option
net.core.wmem_max=262144
```

UDP ICMP 필터링 설정의 확인 및 비활성화

Linux 운영체제 설치 과정에서 방화벽 옵션을 선택하기 않는 것으로 설명하였습니다. 디폴트 상태에서 방화벽 옵션은 인스톨러에 의해 자동 설정됩니다. 이것 때문에 필자가 고생한 일이 한 두 번이 아니었습니다. 여기에서는 방화벽 옵션이 설정되어 있지 않으며 ICMP 필터링이 비활성화 되어 있음을 다시 한 번 확인해 보기로 합니다.

UDP ICMP가 방화벽에 의해 거부되는 경우, Oracle Clusterware가 실행 후 몇 분 안에 크래시 되는 문제를 겪게 됩니다. Oracle Clusterware 프로세스가 실패하면 <machine_name>_evmocr.log 파일에 아래와 같은 내용이 기록됩니다:

```
08/29/2005 22:17:19
oac_init:2: Could not connect to server, clsc retcode = 9
08/29/2005 22:17:19
a_init:12!: Client init unsuccessful : [32]
ibctx:1:ERROR: INVALID FORMAT
proprinit:problem reading the bootblock or superbloc 22
```

이러한 문제가 발생하는 원인은 대부분 ICMP (iptables) Rejection Rule이 설정되어 있기 때문입니다. 이 룰을 비활성화하거나 방화벽을 아예 꺼버리면 문제의 재발을 방지할 수 있습니다. 이와 같이 하면 Oracle Clusterware 소프트웨어가 문제 없이 정상 실행될 것입니다. root 사용자 계정을 이용하여 아래 명령을 실행합니다:

1. 방화벽 옵션이 비활성화 상태인지 확인합니다. 아래와 같이 방화벽 옵션이 사용되지 않고 있다면, 뒷 부분에 설명되는 작업을 실행할 필요가 없습니다.

```
# /etc/rc.d/init.d/iptables status
Firewall is stopped.
```

2. 방화벽 옵션이 실행 중이라면 먼저 UDP ICMP rejection을 수동으로 비활성화해야 합니다:

```
# /etc/rc.d/init.d/iptables stop
Flushing firewall rules: [ OK ]
Setting chains to policy ACCEPT: filter [ OK ]
Unloading iptables modules: [ OK ]
```

3. 그런 다음, 아래와 같이 실행하여 서버가 리부트된 후에도 UDP ICMP rejection이 실행되지 않도록 설정합니다:

```
# chkconfig iptables off
```

8. Openfiler 설치

아래 설치작업을 네트워크 스토리지 서버(openfiler1)에 대해 실행합니다!

양쪽 Oracle RAC 노드에 네트워크 설정 작업을 완료했다면, 다음 단계에서는 네트워크 스토리지 서버(openfiler1)에 Openfiler 소프트웨어를 설치해야 합니다. 문서의 뒷부분에서 네트워크 스토리지 서버를 전체 Oracle 10g RAC 공유 스토리지 환경을 위한 iSCSI 스토리지 디바이스로 설정하게 될 것입니다.

rPath Linux 를 기반으로 하는 Openfiler는 단일 프레임워크를 통해 파일 기반 NAS(Network Attached Storage)와 블록 기반 SAN(Storage Area Network)을 동시에 지원하는 무료 브라우저 기반 네트워크 스토리지 관리 유틸리티입니다. Openfiler의 전체 소프트웨어 스택은 Apache, Samba, LVM2, ext3, Linux NFS, iSCSI Enterprise Target 등의 오픈 소스 애플리케이션과 인터페이스하고 있습니다.

Openfiler는 이러한 다양한 테크놀로지를 간결한 형태의 솔루션에 통합하고 강력한 웹 기반 관리 인터페이스를 통해 관리할 수 있게 합니다.

Openfiler는 CIFS, NFS, HTTP/DAV, FTP 등을 지원합니다. 하지만 여기에서는 iSCSI 지원 기능만을 이용하여 Oracle 10g RAC에서 요구하는 SAN 기반 공유 스토리지 환경을 저렴하게 구성하기로 합니다. Openfiler 서버에는 USB 2.0 인터페이스를 통해 500GB 외장형 하드 드라이브가 연결됩니다. 본 문서에서는 Openfiler 서버를 이용하여 이 디스크를 iSCSI 기반 스토리지로 설정한 후, Oracle Clusterware와 Oracle ASM 볼륨이 요구하는 공유 파일을 저장하기 위해 사용합니다.

Openfiler에 대한 자세한 설명은 Openfiler 웹사이트 <http://www.openfiler.com/>에서 확인하실 수 있습니다.

Openfiler 다운로드

아래 링크를 통해 Openfiler 2.1 x86(Final)을 다운로드할 수 있습니다. Openfiler를 다운로드한 다음에는 ISO 이미지를 CD로 구워야 합니다.

Openfiler

openfiler-2.1-x86-disc1.iso (302 MB)

MS Windows 머신에서 ISO 파일을 다운로드한 경우, 이미지를 CD로 굽기 위한 다양한 옵션을 활용할 수 있습니다. 본 문서에서는 ISO 이미지를 CD를 굽기 위한 소프트웨어와 사용 방법에 대해 여러분이 잘 알고 있는 것으로 가정하겠습니다. 방법을 잘 모르거나, 이미지를 CD로 굽는데 필요한 소프트웨어가 없다면 아래 두 가지 소프트웨어 패키지 중 하나를 사용할 것을 추천 드립니다:

UltraISO

Magic ISO Maker

Openfiler 설치

이 섹션에서는 Openfiler 소프트웨어를 설치하는 과정을 단계별로 설명합니다. 여기에서는 디폴트 설정을 그대로 사용하여 Openfiler를 설치하고 있습니다. 수작업을 통한 변경 작업이 필요한 부분은 로컬 네트워크 설정이 유일합니다.

설치가 완료되면, 서버를 리부트한 뒤 모든 컴포넌트, 서비스, 드라이버가 정상적으로 실행, 인식되었는지 확인합니다. 리부트가 완료된 후 외장형 Maxtor 하드 드라이브는 Openfiler 서버에 의해 /dev/sda 디바이스로 인식되어야 합니다.

좀 더 자세한 설치 방법은 <http://www.openfiler.com/docs/>에서 확인하실 수 있습니다. 하지만, Oracle 10g RAC 환경에서는 아래에 제시된 설명을 기반으로 시스템을 구성해야 함을 잊지 마시기 바랍니다.

Openfiler 소프트웨어를 네트워크 스토리지 서버에 설치하기 전에, NIC 인터페이스 카드와 외장형 하드 드라이브를 설치/연결하고 전원을 켜 두어야 합니다.

Openfiler ISO 이미지를 CD로 구운 뒤, CD를 네트워크 스토리지 서버(openfiler1)에 넣고 전원을 켭니다. 그런 다음 아래 설명된 방법 대로 설치 스크린 상에서 답변을 입력합니다.

Boot Screen

Openfiler 부트 스크린이 표시됩니다. boot: 프롬프트에서 [Enter] 키를 눌러 설치 프로세스를 시작합니다.

Media Test

CD 미디어를 테스트할 것인지 묻는 질문에 대해 [Skip]을 선택하고 [Enter] 키를 누릅니다. (미디어에 에러가 있다면 CD 버닝 소프트웨어를 통해 이미 경고 메시지가 표시 되었을 것입니다.) 몇 초 동안 비디오 카드, 모니터, 마우스 등의 인식 작업이 수행되고, 인스톨러는 GUI 모드로 전환됩니다.

Welcome to Openfiler NAS/SAN Appliance

[Next]를 눌러 다음 단계로 진행합니다.

Keyboard Configuration

키보드 설정을 위한 프롬프트가 표시됩니다. 환경에 따라 적절한 설정을 선택합니다.

Disk Partitioning Setup

다음 스크린에서는 "Automatic Partitioning" 또는 "Manual Partitioning with Disk Druid" 중 하나를 선택하여 디스크 파티셔닝을 실행해야 합니다. 어떤 옵션을 선택해도 무방하지만, Openfiler 공식 제품문서에서는 Manual Partitioning을 사용할 것을 권고하고 있습니다. 필자가 설치를 위해 사용된 내장 하

드 드라이브의 용량이 작은 데다가 Openfiler 소프트웨어의 저장 공간으로만 사용되기 때문에 (40GB 용량의 내장 하드 드라이브는 iSCSI 스토리지로 활용되지 않습니다), 필자는 "Automatic Partitioning"을 선택하였습니다.

[Automatically partition]을 선택하고 [Next]를 클릭합니다.

서버에 이전에 설치한 Linux 운영체제가 존재하는 경우, 오래된 파티션을 "제거(remove)" 또는 "보존(keep)"할 것인지 묻는 스크린이 표시됩니다. [Remove all partitions on this system] 옵션을 선택합니다. 또 설치 대상에 [hda] 드라이브만이 선택되어 있음을 확인합니다. [Review (and modify if needed) the partitions created] 옵션도 함께 체크하도록 합니다. [Next]를 클릭하여 다음 단계로 진행합니다.

정말로 모든 파티션을 삭제할 것인지 묻는 대화 창이 표시됩니다. [Yes]를 클릭하여 경고를 승인합니다.

Partitioning

이 단계에서 인스톨러가 /dev/hda를 위해 자동으로 선택한 디스크 파티션을 조회하고, 필요한 경우 수정할 수 있습니다. 대부분의 경우 인스톨러는 /boot를 위해 100MB, swap 공간을 위해 메모리 용량의 두 배를 할당하고, 나머지를 root(/) 파티션에 할당합니다. swap 공간의 크기는 최소 1GB로 설정하는 것이 좋습니다. 여기에서는 디폴트 설정을 그대로 사용하기로 합니다. (1GB RAM이 설치되어 있으므로 swap 공간으로는 2GB가 할당됩니다.)

Network Configuration

Openfiler를 설치하기 전에 네트워크 스토리지 서버에 NIC 카드를 설치해 둔 상태이어야 합니다. 이 화면에서는 두 조의 NIC 카드가 모두 정상적으로 인식되고 있어야 합니다.

먼저, 각 NIC 카드에 대해 [Active on boot] 항목이 체크되어 있는지 확인합니다. (인스톨러가 eth1을 활성화하지 않았을 수도 있습니다.)

두 번째로, eth0과 eth1에 대해 [Edit] 작업을 수행합니다. eth0과 eth1에 서로 다른 IP 주소를 할당하더라도 무방합니다. 하지만 eth1(스토리지 네트워크)가 linux1, linux2의 eth1과 동일한 서브넷을 사용하도록 설정해 주는 것이 중요합니다:

eth0:

- [Configure using DHCP] 옵션의 체크를 해제합니다.
- [Activate on boot]를 체크된 상태로 둡니다.
- IP 주소: 192.168.1.195
- 넷마스크: 255.255.255.0

eth1:

- [Configure using DHCP] 옵션의 체크를 해제합니다.
- [Activate on boot]를 체크된 상태로 둡니다.
- IP 주소: 192.168.2.195
- 넷마스크: 255.255.255.0

hostname은 수동으로 설정합니다. 필자의 경우 호스트네임으로 "openfiler1"을 사용하였습니다. 마지막으로 게이트웨이와 DNS 서버를 설정하고 다음 단계로 넘어갑니다.

Time Zone Selection

표준시간대 설정을 위한 화면입니다. 환경에 따라 적절한 설정을 선택합니다

Set Root Password

root 패스워드를 선택하고 [Next]를 클릭합니다.

About to Install

설치 전 최종 확인을 위한 스크린이 표시됩니다. [Next]를 클릭하여 설치 작업을 시작합니다.

Congratulations

이것으로 네트워크 스토리지 서버에 Openfiler를 설치하는 작업을 성공적으로 마쳤습니다. 인스톨러가 CD-ROM 드라이브에서 CD를 자동으로 꺼냅니다. CD를 꺼낸 후 [Reboot]를 클릭하여 시스템을 리부트 합니다.

별다른 문제가 없다면, 리부트 후에 텍스트 로그인 스크린과 (Openfiler 서버 관리를 위한) URL이 표시 될 것입니다.

9. Openfiler를 이용한 iSCSI 볼륨 설정

아래설정작업을네트워크스토리지서버(openfiler1)에대해실행합니다!

Openfiler Storage Control Center는 Openfiler의 관리를 위한 브라우저 기반 톨로 446 포트를 이용한 https 연결을 사용합니다. 접속 방법의 예가 다음과 같습니다:

<https://openfiler1:446/>

Openfiler Storage Control Center 홈 페이지에서 관리자로 로그인합니다. Openfiler를 위한 디폴트 관리자 로그인 계정/암호가 아래와 같습니다:

Username: openfiler

Password: password

첫 번째 페이지에는 [Accounts] / [Authentication] 스크린이 표시됩니다. 본 문서에서는 사용자 계정/그룹 설정 작업이 불필요하므로 자세하게 설명하지 않습니다.

Openfiler를 iSCSI 스토리지 서버로 활용하기 위해서는 iSCSI 서비스 설정, 네트워크 액세스 설정, 물리적 스토리지 생성의 세 가지 작업을 수행해 주어야 합니다.

서비스

서비스를 컨트롤하기 위해 Openfiler Storage Control Center의 [Services] / [Enable/Disable] 화면으로 이동합니다:

Accounts	Volumes	Quota	Shares	Services	Status	General
Enable/Disable		SMB Settings		LDAP Settings		UPS Settings

Enable/Disable services		
Service Name	Status	Modification
SMB/CIFS	Disabled	Enable
NFSv3	Enabled	Disable
HTTP / WebDAV	Disabled	Enable
FTP	Disabled	Enable
iSCSI target	Disabled	Enable
Rsync	Disabled	Enable
UPS	Disabled	Enable
LDAP	Disabled	Enable

그림 6 iSCSI Openfiler 서비스의 활성화

iSCSI 서비스를 활성화하려면, 'iSCSI target' 서비스 네임 아래의 'Enable'을 클릭합니다. 그런 다음, 'iSCSI target'의 상태가 'Enabled'로 변경되었는지 확인합니다.

ietd 프로세스는 Linux 환경에 iSCSI 스토리지 시스템을 구현하기 위한 iSCSI Enterprise Target 소프트웨어의 사용자 레벨 환경을 구현합니다. iSCSI 타겟이 활성화된 상태에서 SSH로 Openfiler 서버에 접속하여 `iscsi-target` 서비스가 실행 중임을 확인할 수 있습니다:

```
[root@openfiler1 ~]# service iscsi-target status
ietd (pid 3784) is running...
```

네트워크 액세스의 제한

다음으로 Oracle RAC 노드(linux1, linux2)가 스토리지 (사실) 네트워크를 통해 iSCSI 볼륨에 접근 가능하도록 Openfiler의 네트워크 액세스를 설정해 주어야 합니다. (iSCSI 볼륨은 다음 단계에서 생성합니다!)

이 작업은 Openfiler Storage Control Center의 [General] / [Local Networks] 스크린에서 수행할 수 있습니다. 관리자는 Local Networks 스크린을 통해 Openfiler 어플라이언스가 익스포트(export)한 리소스에 접근할 수 있는 네트워크 또는 호스트를 설정할 수 있습니다. 본 문서에서는 전체 192.168.2.0 네트워크에 대해 Openfiler 리소스의 접근을 허용하는 대신, 두 개의 Oracle RAC 노드에 대해서만 접근을 허용하는 방법을 사용하기로 합니다.

각각의 Oracle RAC 노드를 입력하는 과정에서 사용되는 'Name' 필드는 참조를 위해서만 사용되는 논리적 이름(logical name)임을 참고하시기 바랍니다. 필자는 노드를 입력할 때에, 해당 IP 주소에 대해 정의된 노드 네임을 사용하였습니다. 다음으로, 'Network/Host' 필드에 실제 노드를 입력할 때에는, (실사 /etc/hosts 파일 또는 DNS에 호스트 네임이 등록되어 있다 할지라도) 반드시 IP 주소를 사용하도록 합니다. 마지막으로 Class C 네트워크에 실제 호스트를 입력할 때에는 서브넷 마스크를 "255.255.255.255"로 설정합니다.

여기서, 클러스터의 각 RAC 노드에 대해 *private* 네트워크(eth1)의 IP 주소를 입력해야 함을 잊지 마시기 바랍니다.

두 개의 Oracle RAC 노드를 추가한 결과가 아래 그림과 같습니다:

Accounts	Volumes	Quota	Shares	Services	Status	General		
Local Networks	HA Cluster Resources	Clock	UPS	Shutdown	Notification	Update	Backup/Restore	Security Shares

Local networks configuration

Delete	Name	Network/Host	Netmask	Type
<input type="checkbox"/>	linux1-priv	192.168.2.100	255.255.255.255	Share
<input type="checkbox"/>	linux2-priv	192.168.2.101	255.255.255.255	Share
New	<input type="text"/>	<input type="text"/>	0.0.0.0 <input type="button" value="v"/>	Share <input type="button" value="v"/>

그림 7 Oracle RAC 노드의 Openfiler 호스트 액세스를 위한 설정

물리적 스토리지

이번 섹션에서는, 클러스터의 Oracle RAC 노드들이 공유 스토리지로 사용할 5 개의 iSCSI 볼륨을 생성하겠습니다. 이를 위해 Openfiler 서버에 연결된 외장형 USB 하드 드라이브에 여러 가지 작업을 실행해 주어야 합니다.

Openfiler 서버는 클라이언트에 서비스를 제공하기 위한 스토리지 디바이스로 내장형 IDE/SATA/SCSI 디스크, 외장형 USB 또는 FireWire 드라이브 등 모든 종류의 드라이브를 지원합니다. OS 레벨에서 디바이스를 감지하고 나면, Openfiler Storage Control Center를 이용하여 스토리지의 셋업 및 관리 작업을 수행할 수 있습니다.

본 문서에서는 500 GB의 외장형 USB 하드 드라이브를 사용하고 있습니다. Openfiler 서버에서는 이 드라이브가 /dev/sda(외장형 Maxtor OneTouch III 드라이브)로 인식됩니다. 이를 호가인하고 iSCSI 볼륨 생성 작업을 시작하기 위해, Openfiler Storage Control Center의 [Volumes] / [Physical Storage Mgmt.] 스크린으로 이동합니다:

Accounts	Volumes	Quota	Shares	Services	Status	General
List of Existing Volumes	Create New Volume	Volume Group Mgmt.		Software RAID Mgmt.	Physical Storage Mgmt.	

Physical Storage Management

Edit Disk	Type	Description	Size	Label type	Partitions
/dev/hda	IDE	SAMSUNG SP4002H	37.31 GB	msdos	3 (view)
/dev/sda	SCSI	HDS72505 0KLAT80	465.76 GB	msdos	0 (view)

그림 8 Openfiler의 물리적 스토리지

물리적 디스크의 파티셔닝

제일 먼저 /dev/sda 외장형 USB 하드 드라이브에 하나의 'primary partition'을 생성합니다. /dev/sda 링크를 클릭하면 파티션의 'Edit' 또는 'Create' 옵션 중 하나를 선택할 수 있습니다. 전체 디스크에 대해 하나의 primary partition을 생성할 것이므로, 대부분의 옵션을 디폴트 상태로 유지하되 'Partition Type'만은 'Extended partition'에서 'Physical volume'으로 수정해 줍니다. 필자가 /dev/sda에 primary partition을 생성하기 위해 사용한 설정값이 아래와 같습니다:

Mode: Primary

Partition Type: Physical volume

Starting Cylinder: 1

Ending Cylinder: 60801

이제 파티션의 사이즈가 465.76 GB로 표시됩니다. 이제 Create 버튼을 클릭합니다. 외장형 하드 드라이브에 새로운 파티션(/dev/sda1)이 생성 되었습니다.

Accounts	Volumes	Quota	Shares	Services	Status	C...
List of Existing Volumes	Create New Volume	Volume Group Mgmt.	Software RAID Mgmt.	Physical Mgr		

Edit partitions in /dev/sda (60801 cylinders with "msdos" label)

Device	Type	Number	Start cyl	End cyl	Blocks	Size	Type
/dev/sda1	Linux Physical Volume (0x8e)	1	1	60801	488384001	455.76 GB	Primary

[Back to the list of physical storage devices](#)

Create a partition in /dev/sda
Can't create any more partitions.

그림 9 물리적 볼륨의 파티셔닝

볼륨 그룹 관리

다음으로 *Volume Group*. 을 생성합니다. 여기서는 새로 생성된 primary partition을 포함하는 rac1이라는 이름의 단일 볼륨 그룹을 생성합니다.

Openfiler Storage Control Center에서 [Volumes] / [Volume Group Mgmt.] 스크린으로 이동합니다. 이미 볼륨 그룹이 존재하는 경우 이 스크린을 통해 표시되지만, 현재는 아무 것도 표시되지 않을 것입니다. Volume Group Management 스크린을 이용하여 새로운 볼륨 그룹의 이름(rac1)을 입력하고 /dev/sda1의 체크박스를 선택하여 파티션을 선택한 다음, 'Add volume group' 버튼을 클릭합니다. 이제 새로 생성된 볼륨 그룹(rac1)이 목록에 표시됩니다.

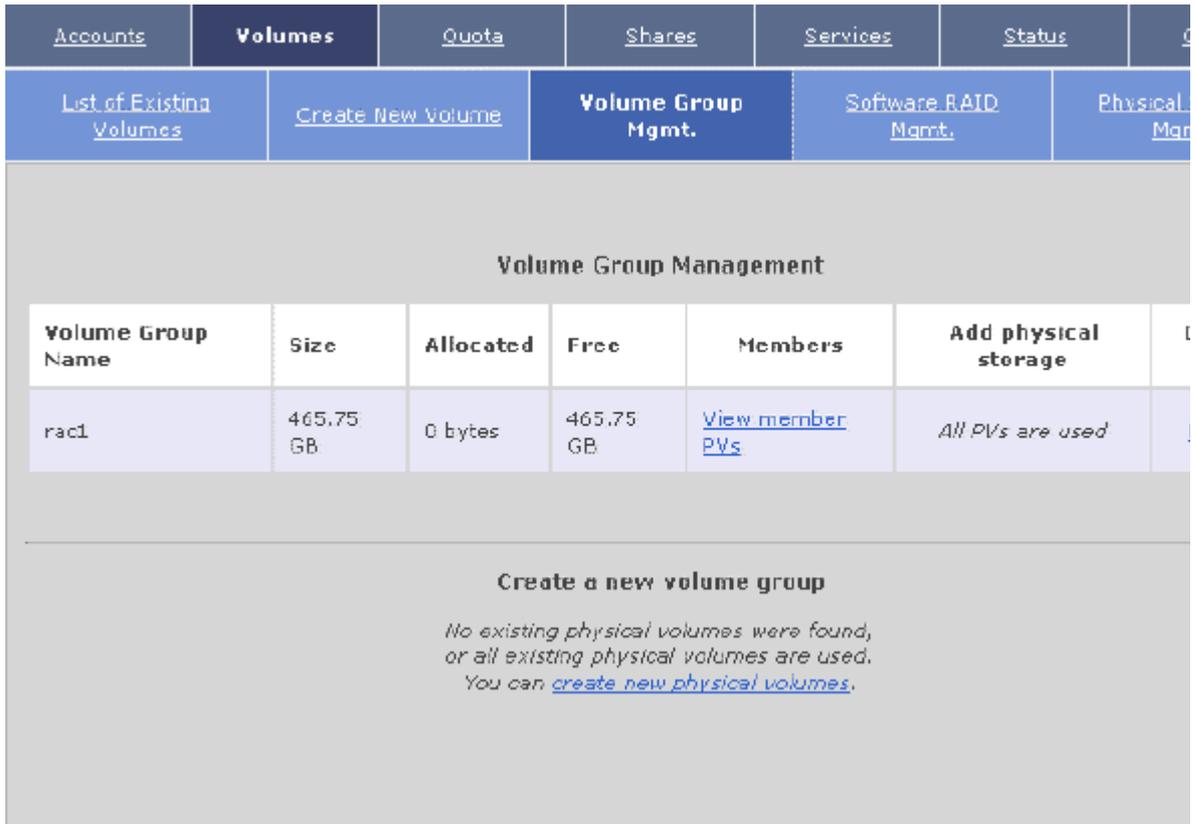


그림 10 새로운 볼륨 그룹의 생성

논리적 볼륨

이제 새로 생성된 볼륨 그룹(rac1) 내에 5 개의 논리적 볼륨(logical volume)을 생성해야 합니다.

Openfiler Storage Control Center에서 [Volumes] / [Create New Volume] 스크린으로 이동합니다. 화면에 새로운 볼륨 그룹(rac1)에 대한 블록 스토리지 통계가 표시되는 것을 확인할 수 있습니다. 또 스크린 하단에는 선택된 볼륨 그룹에 대해 새로운 볼륨을 생성하는 옵션이 있습니다. 이 스크린을 이용하여 아래와 같은 5 개의 논리적(iSCSI) 볼륨을 생성합니다. 각각의 논리적 볼륨을 생성한 뒤에는 "List of Existing Volumes" 스크린이 표시됩니다. 이 화면에서 다시 "Create New Volume" 탭으로 돌아가 5 개의 논리적 볼륨을 차례로 생성해 주어야 합니다.

iSCSI / Logical Volumes			
Volume Name	Volume Description	Required Space (MB)	Filesystem Type
crs	Oracle Clusterware	2,048	iSCSI
asm1	Oracle ASM Volume 1	118,720	iSCSI
asm2	Oracle ASM Volume 2	118,720	iSCSI
asm3	Oracle ASM Volume 3	118,720	iSCSI
asm4	Oracle ASM Volume 4	118,720	iSCSI

이제 네트워크의 iSCSI 클라이언트(linux1, linux2)에 의해 접근될 수 있는 5 개의 iSCSI 디스크를 생성하였습니다. 마지막으로 표시되는 "List of Existing Volumes" 스크린이 아래와 같습니다:

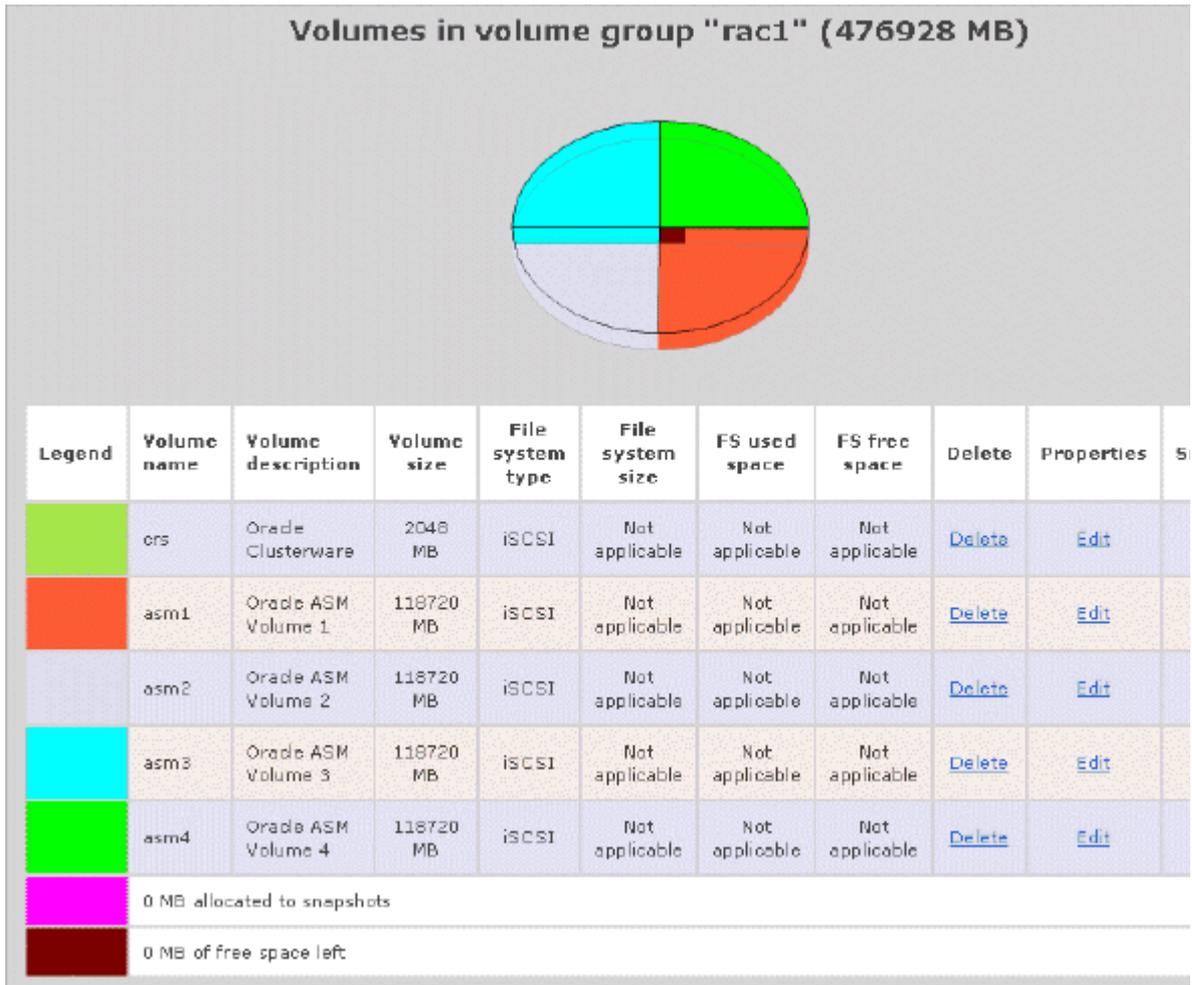


그림 11 새로운 논리적 (iSCSI) 볼륨

새로운 논리적 볼륨에 액세스 권한을 할당

이제 새로 생성된 iSCSI 볼륨에 iSCSI 클라이언트가 접근하기 위한 적절한 권한을 할당해 주어야 합니다. 앞부분에서 두 대의 호스트(Oracle RACE 노드)가 Openfiler의 리소스에 접근하기 위한 액세스 권한을 설정해 준 것을 기억하실 것입니다. 이번에는 두 대의 Oracle RAC 노드가 새로 생성된 5 개의 iSCSI 볼륨에 접근하기 위한 권한을 할당해 줍니다.

Openfiler Storage Control Center에서 [Volumes] / [List of Existing Volumes] 스크린으로 이동합니다. 앞에서 이미 확인한 스크린이 표시될 것입니다. 각각의 논리적 볼륨에 대해 (Properties 컬럼 밑에 위치한) 'Edit' 링크를 클릭합니다. 그러면 해당 볼륨의 'Edit properties' 스크린이 표시됩니다. 스크린의 하단으로 스크롤 한 후, 두 대의 호스트에 대한 설정을 'Deny'에서 'Allow'로 변경한 뒤 'Update' 버튼을 클릭합니다:

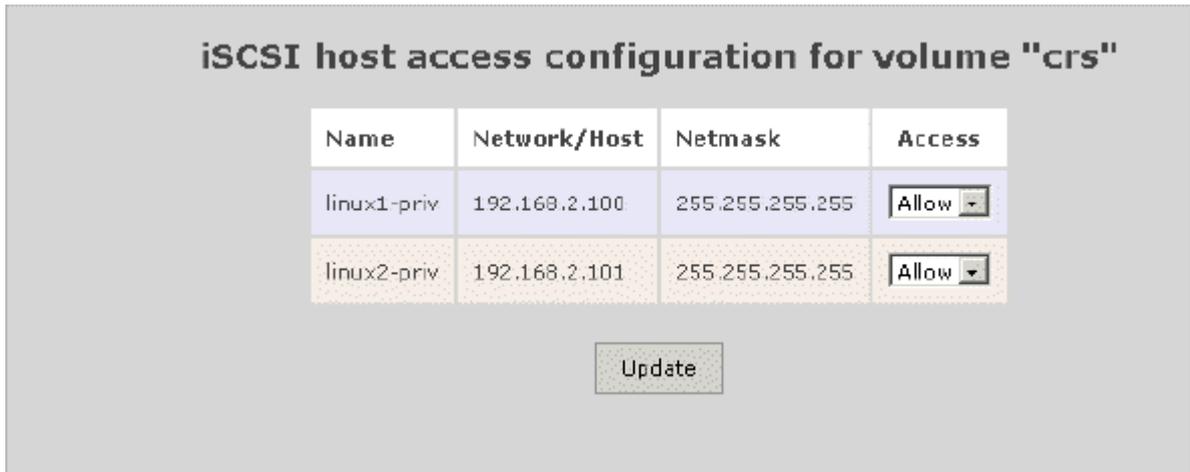


그림 12 논리적 (iSCSI) 볼륨에 호스트 액세스를 허용
이 작업을 5개의 논리적 볼륨에 대해 모두 수행해 줍니다.

iSCSI 타겟의 존재를 클라이언트에 알리기

새로운 논리적 볼륨이 추가될 때마다, Openfiler 서버에 관련 서비스를 재시작해야 합니다. 본 문서의 시나리오에서는 iSCSI 논리적 볼륨을 생성했으므로, iSCSI 타겟(iscsi-target) 서비스를 재시작해야 합니다. 이렇게 함으로써 새로운 iSCSI 타겟의 존재를 네트워크 상의 (액세스 권한을 가진) 모든 클라이언트에게 알릴 수 있습니다. iSCSI 타겟 서비스를 재시작하려면 Openfiler Storage Control Center의 [Services] / [Enable/Disable] 화면으로 이동합니다: iSCSI 타겟 서비스가 이미 활성화되어 있을 것입니다. 이 서비스를 비활성화한 다음 다시 활성화합니다. (그림 6 참고)
Openfiler 서버에서 SSH 세션을 이용해서 같은 작업을 수행할 수도 있습니다:

```
[root@openfiler1 ~]# service iscsi-target restart
Stopping iSCSI target service: [ OK ]
Starting iSCSI target service: [ OK ]
```

10. Oracle RAC 노드에 iSCSI 볼륨 설정하기

클러스터의 모든 Oracle RAC 노드에 대해 iSCSI 이니시에이터(initiator)를 설정해 주어야 합니다! 단, 파티션 생성 작업은 RAC 클러스터의 노드 중 하나에서만 실행합니다.

iSCSI 드라이버가 지원되는 모든 시스템(Linux, Unix, MS Windows, Apple Mac 등)이 iSCSI 클라이언트가 될 수 있습니다. 본 문서의 시나리오에서는 두 대의 Red Hat 4 Linux 서버(linux1 and linux2)가 클라이언트가 됩니다.

이제 두 대의 Oracle RAC 노드에 iSCSI 이니시에이터를 설정할 차례입니다. 이를 위해 양쪽 Oracle RAC 노드의 /etc/iscsi.conf 파일에 네트워크 스토리지 서버(openfiler1)의 이름을 등록하고, 앞에서 생성한 iSCSI 볼륨이 발견될 수 있게 해야 합니다. 그런 다음 노드 중 하나(앞에서 iSCSI 볼륨의 파티셔닝을 수행했던 linux1노드)에서 Openfiler에서 발견한 iSCSI 타겟 이름을 로컬 SCSI 디바이스 네임으로 매핑하는 복잡한 작업을 수행합니다. 이 작업은 Oracle Cluster File System(OCFS2)과 Automatic Storage Management(ASM)를 이용하여 iSCSI 볼륨을 수행하는 경우에 필요합니다. 로컬 SCSI 디바이스 네임과, 이 디바이스 네임으로 매핑되는 iSCSI 타겟을 알고 있어야만, 어떤 볼륨(디바이스)이 OCFS2를 위해 사용되는지, 또 어떤 볼륨이 ASM에서 사용되는지를 알 수 있습니다. Oracle RAC 노드 중 하나가 리부팅될 때마다, 매핑이 달라질 수 있음을 참고하시기 바랍니다. 예를 들어, iSCSI 타겟 네임 "iqn.2006-01.com.openfiler:rac1.crs"가 볼륨 설정 직후 linux1의 /dev/sdd으로 매핑되었다고 가정해 봅시다. 하지만 노드를 리부팅하고 난 뒤에는 iqn.2006-01.com.openfiler:rac1.crs가 /dev/sde로 매핑이 변경될 수 있습니다. 이것이 문제가 되지는 않습니다. 모든 디스크가 OCFS2 또는 ASM에 의해 레이블(label) 처리 되기 때문입니다(뒷부분에서 자세히 설명합니다). 서비스가 볼륨 마운트를 시도하는 경우, 로컬 SCSI 디바이스 네임 대신 레이블을 이용하여 볼륨을 마운트합니다.

iSCSI (initiator) 서비스

각각의 Oracle RAC 노드에서 iSCSI (initiator) 서비스가 실행 중인지를 확인해 봅시다. 이 서비스가 운영 체제의 일부로 포함되어 있지 않다면 iscsi-initiator-utils RPM (i.e. iscsi-initiator-utils-4.0.3.0-

4.i386.rpm) 을 다운로드하여 각 Oracle RAC 노드에 설치해야 합니다.
패키지가 설치되었는지 확인하기 위해, 양쪽 Oracle RAC 노드에서 아래와 같이 실행합니다:

```
# rpm -qa | grep iscsi
iscsi-initiator-utils-4.0.3.0-4
```

패키지가 설치되어 있지 않다면 RHEL4 Update 4 배포본의 3 번 디스크에 있는 iscsi-initiator-utils RPM 패키지를 이용하거나 인터넷 RPM 리소스 사이트에서 downloaded 할 수 있습니다.
패키지가 설치되어 있지 않다면 아래 명령을 실행하여 iscsi-initiator-utils RPM 패키지를 설치합니다:

```
# rpm -Uvh iscsi-initiator-utils-4.0.3.0-4.i386.rpm
warning: iscsi-initiator-utils-4.0.3.0-4.i386.rpm:
  V3 DSA signature: NOKEY, key ID 443e1821
Preparing...                ##### [100%]
   1:iscsi-initiator-utils  ##### [100%]
```

iscsi-initiator-utils RPM의 설치 여부를 확인했다면, 이제 Oracle RAC 노드(iSCSI 클라이언트) 의 /etc/iscsi.conf 파일에 네트워크 스토리지 서버(iSCSI 서버)를 등록해 주는 작업만 수행해 주면 됩니다. /etc/iscsi.conf 파일을 열고 Openfiler 네트워크 스토리지 서버의 호스트네임을 정의하는 DiscoveryAddress 항목을 추가합니다. 본 문서 시나리오에서의 실행 방법이 다음과 같습니다:

```
...
DiscoveryAddress=openfiler1-priv
...
```

양쪽 Oracle RAC 노드의 /etc/iscsi.conf 파일을 수정한 뒤, 두 노드의 iscsi initiator 서비스를 시작(또는 재시작)합니다:

```
# service iscsi restart
Searching for iscsi-based multipath maps
Found 0 maps
Stopping iscsid: iscsid not running

Checking iscsi config: [ OK ]
Loading iscsi driver: [ OK ]
Starting iscsid: [ OK ]
```

또 양쪽 Oracle RAC 노드에서 리부팅 이후에도 iSCSI 서비스가 실행되도록 설정해 주어야 합니다. 이를 위해 아래와 같이chkconfig Linux 커맨드를 실행합니다:

```
# chkconfig --level 345 iscsi on
```

iSCSI 타겟의 발견

iSCSI 이니시에이터 서비스를 설정하고 양쪽 Oracle RAC 노드에서 실행한 뒤 linux1 노드(파티셔닝 및 볼륨 레이블링을 수행할 노드)에서만 아래의 작업을 수행합니다.
(Openfiler 서버의 iscsi-target 서비스를 시작/재시작함으로써) Openfiler 서버에서 사용 가능한 iSCSI 타겟이 공개되면, 또는 iSCSI initiator 서비스가 클라이언트에서 시작/재시작되면, 클라이언트는 새로운 iSCSI 디스크가 사용 가능함을 인식하게 됩니다. 이제 클라이언트의 /var/log/messages 파일에서 아래와 같은 내용을 확인할 수 있습니다:

```
...
Oct 10 19:55:39 linux1 iscsi: iscsid startup succeeded
Oct 10 19:55:39 linux1 iscsid[3073]: Connected to Discovery Address 192.168.2.195
Oct 10 19:55:39 linux1 kernel: iscsi-sfnet:host1: Session established
Oct 10 19:55:39 linux1 kernel: iscsi-sfnet:host0: Session established
Oct 10 19:55:39 linux1 kernel: scsi1 : SFNet iSCSI driver
Oct 10 19:55:39 linux1 kernel: scsi0 : SFNet iSCSI driver
Oct 10 19:55:39 linux1 kernel: iscsi-sfnet:host2: Session established
```

```

Oct 10 19:55:39 linux1 kernel: Vendor: Openfile Model: Virtual disk Rev: 0
Oct 10 19:55:39 linux1 kernel: Type: Direct-Access ANSI SCSI revision
Oct 10 19:55:39 linux1 kernel: iscsi-sfnet:host3: Session established
Oct 10 19:55:39 linux1 kernel: Vendor: Openfile Model: Virtual disk Rev: 0
Oct 10 19:55:39 linux1 kernel: Type: Direct-Access ANSI SCSI revision
Oct 10 19:55:39 linux1 kernel: scsi3 : SFNet iSCSI driver
Oct 10 19:55:39 linux1 kernel: SCSI device sda: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:39 linux1 kernel: Vendor: Openfile Model: Virtual disk Rev: 0
Oct 10 19:55:39 linux1 kernel: Type: Direct-Access ANSI SCSI revision
Oct 10 19:55:39 linux1 kernel: iscsi-sfnet:host4: Session established
Oct 10 19:55:39 linux1 kernel: scsi4 : SFNet iSCSI driver
Oct 10 19:55:39 linux1 kernel: scsi2 : SFNet iSCSI driver
Oct 10 19:55:39 linux1 kernel: Vendor: Openfile Model: Virtual disk Rev: 0
Oct 10 19:55:39 linux1 kernel: Type: Direct-Access ANSI SCSI revision
Oct 10 19:55:39 linux1 kernel: SCSI device sda: drive cache: write through
Oct 10 19:55:39 linux1 kernel: Vendor: Openfile Model: Virtual disk Rev: 0
Oct 10 19:55:39 linux1 kernel: Type: Direct-Access ANSI SCSI revision
Oct 10 19:55:39 linux1 kernel: SCSI device sda: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:39 linux1 kernel: SCSI device sda: drive cache: write through
Oct 10 19:55:39 linux1 kernel: sda: unknown partition table
Oct 10 19:55:39 linux1 kernel: Attached scsi disk sda at scsi1, channel 0, id 0, lun 0
Oct 10 19:55:39 linux1 kernel: SCSI device sdb: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:39 linux1 kernel: SCSI device sdb: drive cache: write through
Oct 10 19:55:39 linux1 kernel: SCSI device sdb: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:39 linux1 scsi.agent[3222]: disk at /devices/platform/host0/target0:0:0/0:0:0:0
Oct 10 19:55:39 linux1 scsi.agent[3237]: disk at /devices/platform/host3/target3:0:0/3:0:0:0
Oct 10 19:55:39 linux1 scsi.agent[3208]: disk at /devices/platform/host1/target1:0:0/1:0:0:0
Oct 10 19:55:39 linux1 kernel: SCSI device sdb: drive cache: write through
Oct 10 19:55:39 linux1 scsi.agent[3283]: disk at /devices/platform/host4/target4:0:0/4:0:0:0
Oct 10 19:55:39 linux1 scsi.agent[3311]: disk at /devices/platform/host2/target2:0:0/2:0:0:0
Oct 10 19:55:40 linux1 kernel: sdb: unknown partition table
Oct 10 19:55:40 linux1 kernel: Attached scsi disk sdb at scsi0, channel 0, id 0, lun 0
Oct 10 19:55:40 linux1 kernel: SCSI device sdc: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:40 linux1 kernel: SCSI device sdc: drive cache: write through
Oct 10 19:55:40 linux1 kernel: SCSI device sdc: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:40 linux1 kernel: SCSI device sdc: drive cache: write through
Oct 10 19:55:40 linux1 kernel: sdc: unknown partition table
Oct 10 19:55:40 linux1 kernel: Attached scsi disk sdc at scsi3, channel 0, id 0, lun 0
Oct 10 19:55:40 linux1 kernel: SCSI device sdd: 4194304 512-byte hdwr sectors (2147 MB)
Oct 10 19:55:40 linux1 kernel: SCSI device sdd: drive cache: write through
Oct 10 19:55:40 linux1 kernel: SCSI device sdd: 4194304 512-byte hdwr sectors (2147 MB)
Oct 10 19:55:40 linux1 kernel: SCSI device sdd: drive cache: write through
Oct 10 19:55:40 linux1 kernel: sdd: unknown partition table
Oct 10 19:55:40 linux1 kernel: Attached scsi disk sdd at scsi4, channel 0, id 0, lun 0
Oct 10 19:55:40 linux1 kernel: SCSI device sde: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:40 linux1 kernel: SCSI device sde: drive cache: write through
Oct 10 19:55:40 linux1 kernel: SCSI device sde: 243138560 512-byte hdwr sectors (124487 MB)
Oct 10 19:55:40 linux1 kernel: SCSI device sde: drive cache: write through
Oct 10 19:55:40 linux1 kernel: sde: unknown partition table
Oct 10 19:55:40 linux1 kernel: Attached scsi disk sde at scsi2, channel 0, id 0, lun 0
Oct 10 19:56:09 linux1 portmap: portmap startup succeeded
...

```

위 로그 정보는 클라이언트 (linux1) 가 iSCSI 스토리지 서버 (openfiler1-priv at 192.168.2.195)와 iSCSI 세션을 설정할 수 있게 되었음을 의미합니다..

또 로컬 SCSI 디바이스 네임이 iSCSI 타겟의 호스트 ID와 LUN에 어떻게 매핑되는지 확인할 수 있습니다:

```

Attached scsi disk sda at scsi1, channel 0, id 0, lun 0
Attached scsi disk sdb at scsi0, channel 0, id 0, lun 0

```

```
Attached scsi disk sdc at scsi3, channel 0, id 0, lun 0
Attached scsi disk sdd at scsi4, channel 0, id 0, lun 0
Attached scsi disk sde at scsi2, channel 0, id 0, lun 0
```

로컬 SCSI 디바이스 네임이 iSCSI 타겟의 호스트 ID와 LUN에 어떻게 매핑되는지 확인하기 위한 또 다른 방법으로 dmesg 커맨드가 있습니다:

```
# dmesg | sort | grep '^Attached scsi disk'
Attached scsi disk sda at scsi1, channel 0, id 0, lun 0
Attached scsi disk sdb at scsi0, channel 0, id 0, lun 0
Attached scsi disk sdc at scsi3, channel 0, id 0, lun 0
Attached scsi disk sdd at scsi4, channel 0, id 0, lun 0
Attached scsi disk sde at scsi2, channel 0, id 0, lun 0
```

이제 iSCSI 타겟 네임과 로컬 SCSI ID(아래에서는 HOST ID로 표시됩니다)의 매핑 정보를 얻기 위해, 클라이언트(linux 1)에서 iscsi-ls 커맨드를 실행합니다:

```
# iscsi-ls
*****
SFNet iSCSI Driver Version ...4:0.1.11-3(02-May-2006)
*****
TARGET NAME       : iqn.2006-01.com.openfiler:rac1.asm4
TARGET ALIAS      :
HOST ID           : 0
BUS ID            : 0
TARGET ID         : 0
TARGET ADDRESS    : 192.168.2.195:3260,1
SESSION STATUS    : ESTABLISHED AT Tue Oct 10 19:55:40 EDT 2006
SESSION ID        : ISID 00023d000001 TSIH c00
*****
TARGET NAME       : iqn.2006-01.com.openfiler:rac1.asm3
TARGET ALIAS      :
HOST ID           : 1
BUS ID            : 0
TARGET ID         : 0
TARGET ADDRESS    : 192.168.2.195:3260,1
SESSION STATUS    : ESTABLISHED AT Tue Oct 10 19:55:40 EDT 2006
SESSION ID        : ISID 00023d000001 TSIH b00
*****
TARGET NAME       : iqn.2006-01.com.openfiler:rac1.asm2
TARGET ALIAS      :
HOST ID           : 2
BUS ID            : 0
TARGET ID         : 0
TARGET ADDRESS    : 192.168.2.195:3260,1
SESSION STATUS    : ESTABLISHED AT Tue Oct 10 19:55:40 EDT 2006
SESSION ID        : ISID 00023d000001 TSIH d00
*****
TARGET NAME       : iqn.2006-01.com.openfiler:rac1.asm1
TARGET ALIAS      :
HOST ID           : 3
BUS ID            : 0
TARGET ID         : 0
TARGET ADDRESS    : 192.168.2.195:3260,1
SESSION STATUS    : ESTABLISHED AT Tue Oct 10 19:55:40 EDT 2006
SESSION ID        : ISID 00023d000001 TSIH e00
*****
TARGET NAME       : iqn.2006-01.com.openfiler:rac1.crs
TARGET ALIAS      :
HOST ID           : 4
```

```
BUS ID          : 0
TARGET ID       : 0
TARGET ADDRESS  : 192.168.2.195:3260,1
SESSION STATUS  : ESTABLISHED AT Tue Oct 10 19:55:40 EDT 2006
SESSION ID      : ISID 00023d000001 TSIH f00
```

로컬 SCSI ID와 iSCSI 타겟 호스트 ID/LUN 간의 매핑 정보, 그리고 iSCSI 타겟 네임과 SCSI ID 매핑 정보를 이용하여 linux1 호스트의 iSCSI 타겟 네임 / 로컬 SCSI 디바이스 네임 매핑을 생성할 수 있습니다.

iSCSI 타겟 네임 -> 로컬 SCSI 디바이스 네임		
iSCSI 타겟 네임	Host / SCSI ID	SCSI 디바이스 네임
iqn.2006-01.com.openfiler:rac1.asm4	0	/dev/sdb
iqn.2006-01.com.openfiler:rac1.asm3	1	/dev/sda
iqn.2006-01.com.openfiler:rac1.asm2	2	/dev/sde
iqn.2006-01.com.openfiler:rac1.asm1	3	/dev/sdc
iqn.2006-01.com.openfiler:rac1.crs	4	/dev/sdd

iSCSI 볼륨에 파티션을 생성

다음 단계에서는 각각의 iSCSI 볼륨에 primary partition을 생성합니다. 앞에서 설명한 것처럼 Oracle Cluster File System Release 2(OCFS2)을 이용하여 Oracle Clusterware 소프트웨어가 공유하는 두 가지 파일을 저장하게 될 것입니다. 그런 다음 Automatic Storage Management(ASM)를 이용하여 4개의 ASM 볼륨을 생성합니다. 4개 중 2개는 모든 물리적 데이터베이스 파일(데이터/인덱스 파일, 온라인 리두 로그 파일, 컨트롤 파일)을 위해, 나머지 2개는 *Flash Recovery Area : RMAN 백업 및 아카이브 리두 로그 파일*을 위해 사용됩니다.

아래 표는 5가지 iSCSI 볼륨이 지원하는 파일 시스템에 대해 설명하고 있습니다:

오라클 공유 드라이브 설정					
파일 시스템 타입	iSCSI 타겟 네임	사이즈	마운트 포인트	ASM 디스크그룹 네임	파일 타입
OCFS2	crs	2 GB	/u02/oradata/orcl		Oracle Cluster Regis Voting Disk - (~20M
ASM	asm1	118 GB	ORCL:VOL1	+ORCL_DATA1	Oracle Database Fil
ASM	asm2	118 GB	ORCL:VOL2	+ORCL_DATA1	Oracle Database Fil
ASM	asm3	118 GB	ORCL:VOL3	+FLASH_RECOVERY_AREA	Oracle Flash Recove
ASM	asm4	118 GB	ORCL:VOL4	+FLASH_RECOVERY_AREA	Oracle Flash Recove
합계		474 GB			

위 표에서 설명된 것처럼, 각각의 iSCSI 볼륨에 Linux primary partition을 생성해야 합니다. Linux에서는 파티션의 생성/제거를 위해 fdisk 커맨드를 사용합니다. 5개의 SCSI 볼륨에 대해 primary partition을 생성할 때 디폴트 설정을 그대로 사용합니다(디폴트 설정에서는 전체 디스크를 하나의 파티션으로 설정합니다). 디바이스가 올바른 DOS 파티션(또는 Sun, SGI, OSF 디스크레이블)을 포함하고 있지 않다는 경고가 뜰 수도 있지만 무시해도 무방합니다.

본 문서의 시나리오에서는, linux1에서 fdisk커맨드를 실행하여 앞 단계에서 확인한 각각의 로컬 SCSI 디바이스에 대해 primary partition을 생성합니다:

```
/dev/sda
/dev/sdb
/dev/sdc
/dev/sdd
/dev/sde
```

각 iSCSI 볼륨의 파티션을 생성하는 작업은 Oracle RAC 클러스터의 노드 중 하나에서만 실행되어야 함을 명심하십시오!

```
# -----
```

```
# fdisk /dev/sda
```

```

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-15134, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-15134, default 15134): 15134

```

```
Command (m for help): p
```

```

Disk /dev/sda: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	15134	121563823+	83	Linux

```

Command (m for help): w
The partition table has been altered!

```

```

Calling ioctl() to re-read partition table.
Syncing disks.

```

```
# -----
```

```
# fdisk /dev/sdb
```

```

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)

```

```

p
Partition number (1-4): 1
First cylinder (1-15134, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-15134, default 15134): 15134

```

```
Command (m for help): p
```

```

Disk /dev/sdb: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	15134	121563823+	83	Linux

```

Command (m for help): w
The partition table has been altered!

```

```

Calling ioctl() to re-read partition table.
Syncing disks.

```

```
# -----
```

```
# fdisk /dev/sdc
```

```

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)

```

```

p
Partition number (1-4): 1
First cylinder (1-15134, default 1): 1

```

Last cylinder or +size or +sizeM or +sizeK (1-15134, default 15134): **15134**

Command (m for help): **p**

Disk /dev/sdc: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		1	15134	121563823+	83	Linux

Command (m for help): **w**
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

fdisk /dev/sdd

Command (m for help): **n**

Command action

 e extended

 p primary partition (1-4)

p

Partition number (1-4): **1**

First cylinder (1-1009, default 1): **1**

Last cylinder or +size or +sizeM or +sizeK (1-1009, default 1009): **1009**

Command (m for help): **p**

Disk /dev/sdd: 2147 MB, 2147483648 bytes
67 heads, 62 sectors/track, 1009 cylinders
Units = cylinders of 4154 * 512 = 2126848 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sdd1		1	1009	2095662	83	Linux

Command (m for help): **w**
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

fdisk /dev/sde

Command (m for help): **n**

Command action

 e extended

 p primary partition (1-4)

p

Partition number (1-4): **1**

First cylinder (1-15134, default 1): **1**

Last cylinder or +size or +sizeM or +sizeK (1-15134, default 15134): **15134**

Command (m for help): **p**

Disk /dev/sde: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device Boot      Start          End      Blocks   Id  System
/dev/sde1        1            15134   121563823+  83  Linux

```

Command (m for help): **w**
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

파티션 생성 작업을 모두 완료했다면, 클러스터의 전체 Oracle RAC 노드에서 root 계정 권한으로 아래 커맨드를 실행하고 파티션 변경 사항을 커널에 통보합니다.

partprobe

fdisk -l

Disk /dev/hda: 40.0 GB, 40000000000 bytes
255 heads, 63 sectors/track, 4863 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device Boot      Start          End      Blocks   Id  System
/dev/hda1 *        1             13        104391    83  Linux
/dev/hda2          14           4863     38957625   8e  Linux LVM

```

Disk /dev/sda: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device Boot      Start          End      Blocks   Id  System
/dev/sda1        1            15134   121563823+  83  Linux

```

Disk /dev/sdb: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device Boot      Start          End      Blocks   Id  System
/dev/sdb1        1            15134   121563823+  83  Linux

```

Disk /dev/sdc: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device Boot      Start          End      Blocks   Id  System
/dev/sdc1        1            15134   121563823+  83  Linux

```

Disk /dev/sdd: 2147 MB, 2147483648 bytes
67 heads, 62 sectors/track, 1009 cylinders
Units = cylinders of 4154 * 512 = 2126848 bytes

```

Device Boot      Start          End      Blocks   Id  System
/dev/sdd1        1             1009     2095662    83  Linux

```

Disk /dev/sde: 124.4 GB, 124486942720 bytes
255 heads, 63 sectors/track, 15134 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device Boot      Start          End      Blocks   Id  System

```

```
/dev/sde1          1          15134   121563823+  83  Linux
```

페이지 1 페이지 2 페이지 3

Linux, iSCSI 환경에서 Oracle RAC 10g Release 2 클러스터 설치하기 (계속)

개발, 테스트 용도로만 사용할 수 있습니다. 운영 목적의 시스템에는 적용될 수 없습니다!

11. "oracle" 사용자 및 디렉토리 생성

클러스터의 모든 Oracle RAC 노드에서 아래 작업을 수행합니다!

예제 구성에서 Oracle Clusterware가 공유하는 모든 파일은 OCFS2에 저장됩니다. OCFS2를 사용하는 경우 모든 클러스터 노드의 *oracle* 계정과 *dba* 그룹은 동일한 UID/GID를 가져야 합니다. UID 또는 GID가 다른 경우, OCFS2 파일 시스템의 파일이 "unowned" 상태로 표시되거나 다른 사용자에게 의해 소유된 것처럼 표시될 수도 있습니다. 예제 구성에서는 *oracle* UID로 175를, *dba* GID로 115를 사용합니다.

오라클 그룹 및 사용자의 생성

dba 그룹, *oracle* 계정 및 관련 디렉토리를 아래와 같이 생성합니다.

```
# mkdir -p /u01/app
# groupadd -g 115 dba
# groupadd -g 116 oinstall
# useradd -u 175 -g 115 -d /u01/app/oracle -s /bin/bash -c "Oracle Software Owner" -p oracle ora
# chown -R oracle:dba /u01
# passwd oracle
# su - oracle
```

참고: 각 RAC 노드에 Oracle 환경 변수를 설정할 때, 노드 별로 서로 다른 Oracle SID를 할당해야 합니다! 예제 구성에서 사용된 설정이 다음과 같습니다:

```
linux1 : ORACLE_SID=orcl1
```

```
linux2 : ORACLE_SID=orcl2
```

oracle 사용자 계정을 위한 로그인 스크립트 생성

두 노드에서 "*oracle*" UNIX 사용자 계정을 생성한 뒤, *.bash_profile*을 실행하여 환경 변수가 올바르게 설정되었는지 확인합니다

```
.....
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

alias ls="ls -FA"

# User specific environment and startup programs
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1
export ORA_CRS_HOME=$ORACLE_BASE/product/crs
export ORACLE_PATH=$ORACLE_BASE/common/oracle/sql:.$ORACLE_HOME/rdbms/admin

# Each RAC node must have a unique ORACLE_SID. (i.e. orcl1, orcl2,...)
export ORACLE_SID=orcl1

export PATH=.:${PATH}:%HOME/bin:%ORACLE_HOME/bin
export PATH=${PATH}:/usr/bin:/bin:/usr/bin/X11:/usr/local/bin
export PATH=${PATH}:%ORACLE_BASE/common/oracle/bin
export ORACLE_TERM=xterm
export TNS_ADMIN=%ORACLE_HOME/network/admin
```

```

export ORA_NLS10=$ORACLE_HOME/nls/data
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$ORACLE_HOME/oracm/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
export CLASSPATH=$ORACLE_HOME/JRE
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/jlib
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/rdbms/jlib
export CLASSPATH=${CLASSPATH}:$ORACLE_HOME/network/jlib
export THREADS_FLAG=native
export TEMP=/tmp
export TMPDIR=/tmp
.....

```

OCFS2 / Clusterware를 위한 마운트 포인트 생성

마지막으로, 두 가지 Oracle Clusterware 파일을 저장할 OCFS2 파일시스템의 마운트 포인트를 생성합니다. root 계정 권한으로 아래와 같이 명령을 실행합니다:

```

$ su -
# mkdir -p /u02/oradata/orcl
# chown -R oracle:dba /u02

```

12. 오라클 환경을 위한 Linux 서버 설정

클러스터의 모든 Oracle RAC 노드에서 아래 작업을 수행합니다!

이 섹션에서 설명되는 명령 중 일부는 Oracle RAC 노드의 각 서버가 부팅될 때마다 반복적으로 실행되어야 합니다. 이 섹션은 오라클 환경의 Linux 커널 매개변수 설정에 관련한 매우 상세한 정보를 포함하고 있습니다. 명령을 스타트업 스크립트(/etc/sysctl.conf)에 포함시키는 방법은 섹션 15("Oracle RAC 노드를 위한 스타트업 커맨드")에서 설명되고 있습니다.

개요

이 섹션에서는 Oracle10g RAC 설치를 위한 사전 준비 단계로 Linux 서버를 설정하는 방법을 설명하고 있습니다. 이를 위해 서버의 스왑 공간 확인, 공유 메모리/세마포어 설정, 파일 핸들 최대 갯수 설정, IP 로컬 포트 영역 설정, oracle 계정의 shell limit 설정, 전체 커널 매개변수의 활성화, 클러스터 노드의 시간/날짜 설정 검증 등의 작업이 필요합니다.

매개변수들을 설정하는 방법에는 여러 가지가 있습니다. 예제 구성에서는 모든 커맨드를 /etc/sysctl.conf 파일에 포함시켜 변경 내용이 영구적으로 적용되도록 하는 방법이 사용됩니다.

Swap 공간의 설정

Oracle10g Release 2 설치 환경은 최소 512MB의 메모리를 요구합니다. 참고: 설치 과정에서 swap 공간을 잘못 설정하는 경우 Oracle Universal Installer의 실행 도중 행(hang) 현상이 발생하거나 비정상 종료될 수 있습니다.)

할당된 메모리 용량을 확인하는 방법이 아래와 같습니다:

```

# cat /proc/meminfo | grep MemTotal
MemTotal: 1034180 kB

```

swap 공간의 메모리 용량을 확인하는 방법은 아래와 같습니다:

```

# cat /proc/meminfo | grep SwapTotal
SwapTotal: 2031608 kB

```

512MB 이하의 메모리를 사용하는 경우에는 임시 swap 파일을 생성함으로써 swap 공간을 추가할 수 있습니다. 이 방법을 이용하면 로우 디바이스(raw device)를 사용하거나 시스템의 리빌드(rebuild)를 수행하는 수고를 덜 수도 있습니다.

root 계정에서 swap공간으로 사용될 파일(300MB)을 추가로 할당하기 위한 명령이 아래와 같습니다:

```

# dd if=/dev/zero of=tempswap bs=1k count=300000
다음으로 파일 권한을 변경합니다:

```

```
# chmod 600 tempswap
마지막으로 해당 "파티션"을 swap 공간으로 포맷하고, 기존 swap공간에 추가합니다:
```

```
# mke2fs tempswap
# mkswap tempswap
# swapon tempswap
```

공유 메모리의 설정

공유 메모리(shared memory)는 프로세스가 공유 메모리 세그먼트를 통해 공통 데이터 구조 및 데이터에 접근할 수 있게 합니다. 공유 메모리는 커널 작업을 수반하지 않으며 프로세스 간의 데이터 복제 작업이 불필요하기 때문에, IPC(inter-process communication)를 위한 가장 빠른 방법으로 선호되고 있습니다.

오라클은 오라클의 백업 및 포어그라운드 프로세스를 위한 공유 메모리 영역으로 Shared Global Area (SGA)를 제공합니다. SGA는 데이터베이스 버퍼 캐시, 공유 SQL, 접근 경로 등의 저장에 활용되므로, SGA 크기의 설정에 따라 오라클의 성능이 크게 달라질 수 있습니다.

공유 메모리 설정을 확인하려면 아래와 같이 실행합니다:

```
# ipcs -lm
----- Shared Memory Limits -----
max number of segments = 4096
max seg size (kbytes) = 32768
max total shared memory (kbytes) = 8388608
min seg size (bytes) = 1
```

Setting의 설정

SHMMAX 매개변수는 공유 메모리 세그먼트의 최대 크기(바이트 단위)를 정의하는데 사용 됩니다. 오라클 SGA는 공유 메모리로 구성되며, SHMMAX가 올바르게 설정되지 않은 경우 SGA의 크기가 제약될 수도 있습니다. 따라서 SGA의 크기보다 작지 않도록 SHMMAX를 설정해야 합니다. SHMMAX 매개변수가 잘못 설정된 경우에는 다음과 같은 에러가 발생합니다:

```
ORA-27123: unable to attach to shared memory segment
```

SHMMAX 매개변수의 설정값을 확인하려면 아래와 같이 실행합니다:

```
# cat /proc/sys/kernel/shmmax
33554432
```

SHMMAX의 디폴트 값은 32MB입니다. 이 사이즈는 Oracle SGA로 활용하기에는 너무 부족 합니다. 필자는 보통 SHMMAX 매개변수를 2GB로 설정합니다. 그 방법이 아래와 같습니다: 아래 커맨드를 이용하여 /proc 파일시스템(/proc/sys/kernel/shmmax)에 변경 사항을 직접 적용하면 서버를 리부팅 하지 않고도 SHMAX 설정을 변경할 수 있습니다:

```
# sysctl -w kernel.shmmax=2147483648
```

다음으로, /etc/sysctl.conf 스타트업 파일에 커널 매개변수를 추가하여 변경 사항을 영구적으로 적용합니다:

```
# echo "kernel.shmmax=2147483648" >> /etc/sysctl.conf
```

SHMMNI의 설정

이제 SHMMNI 매개변수를 검토할 차례입니다. 이 매개변수는 시스템 전체에서 사용하는 공유 메모리 세그먼트의 최대 숫자를 설정하는데 사용되며 디폴트 값은 4096입니다. SHMMNI 설정값을 확인하는 방법이 아래와 같습니다:

```
# cat /proc/sys/kernel/shmni
```

4096

SHMMNI의 디폴트 설정값은 예제 구성의 Oracle RAC 10g Release 2 설치를 위해 충분한 수준입니다.

SHMALL의 설정

마지막으로 SHMALL 공유 메모리 커널 매개변수를 살펴 봅시다. 이 매개변수는 특정 시점에 시스템에서 사용 가능한 공유 메모리의 최대 크기(페이지 단위)를 정의합니다. 따라서 이 매개변수는 최소한 아래 값보다 커야 합니다:

```
ceil(SHMMAX/PAGE_SIZE)
```

SHMALL의 디폴트 사이즈는 2097152이며 아래 명령을 이용하여 확인할 수 있습니다:

```
# cat /proc/sys/kernel/shmall
2097152
```

SHMALL의 디폴트 설정값은 예제 구성의 Oracle RAC 10g Release 2 설치를 위해 충분한 수준입니다.

참고: i386 플랫폼 기반 Red Hat Linux의 페이지 사이즈는 4,096 바이트입니다. 하지만 bigpages를 이용하면 메모리 페이지의 사이즈를 더 크게 설정할 수 있습니다.)

Semaphore의 설정

이것으로 공유 메모리 설정은 완료되었습니다. 다음은 세마포어를 설정할 차례입니다. 세마포어(semaphore)란 (공유 메모리와 같은) 공유 리소스의 사용 과정에서 프로세스(또는 프로세스 내 스레드) 간의 동기화를 위해 사용되는 일종의 카운터(counter)입니다. 세마포어 셋(semaphore set)은 Unix System V에서 지원됩니다. 애플리케이션이 세마포어를 요청하는 경우, 운영체제는 “셋(set)”을 통해 세마포어를 지원합니다

세마포어 설정값을 확인하는 방법이 아래와 같습니다:

```
# ipcs -ls
----- Semaphore Limits -----
max number of arrays = 128
max semaphores per array = 250
max semaphores system wide = 32000
max ops per semop call = 32
semaphore max value = 32767
```

또는 아래 명령을 사용할 수도 있습니다:

```
# cat /proc/sys/kernel/sem
250 32000 32 128
```

SEMMSL의 설정

SEMMSL 커널 매개변수는 세마포어 셋 당 세마포어의 최대 갯수를 정의합니다.

오라클은 init.ora 파일의 PROCESS 인스턴스 매개변수의 (전체 데이터베이스 중) 최대 값에 10을 더한 값을 SEMMSL의 설정값으로 사용할 것을 권장하고 있습니다. 또 SEMMSL의 값을 100 이상으로 설정하는 것이 권장됩니다.

SEMMNI의 설정

SEMMNI 커널 매개변수는 전체 Linux 시스템의 (“세마포어 셋”이 아닌) 세마포어의 최대 개수를 정의하는데 사용됩니다. 오라클은 SEMMNI를 100 이상의 값으로 설정할 것을 권장하고 있습니다.

SEMMNS의 설정

SEMMNS 커널 매개변수는 전체 Linux 시스템의 (“세마포어 셋”이 아닌) 세마포어의 최대 개수를 정의하는데 사용됩니다.

오라클은 각 데이터베이스의 PROCESSES 인스턴스 매개변수의 값을 모두 더한 뒤, 가장 큰 PROCESSES 값을 두 차례 더하고, 마지막으로 각 데이터베이스 별로 10을 더한 값으로 SEMMNS를 설정하도록 권고하고 있습니다.

Linux 시스템에 할당될 수 있는 세마포어의 최대 개수를 결정하기 위해 다음과 같은 공식을 사용합니다. 아래의 두 항목 중 낮은 쪽을 적용합니다:

```
SEMMNS -or- (SEMMSL * SEMMNI)
```

SEMOPM의 설정

SEMOPM 커널 매개변수는 semop 시스템 호출 (system call) 별로 수행될 수 있는 세마포어 작업의 수를 설정하는데 사용됩니다.

semop 시스템 호출(함수)은 하나의 semop 시스템 호출을 통해 여러 개의 세마포어를 지원합니다. 세마포어 셋에서 가질 수 있는 세마포어의 최대값은 SEMMSL 매개변수를 통해 정의됩니다. 따라서 SEMOPM을 SEMMSL과 동일하게 설정하는 것이 권장됩니다.

오라클은 SEMOPM을 100 이상의 값으로 설정할 것을 권장하고 있습니다.

세마포어 커널 매개변수의 설정

마지막으로 세마포어 매개변수를 설정하는 방법에 대해 알아보겠습니다. 여기에서는 SEMOPM 만을 변경하고, 다른 설정은 디폴트 값을 그대로 사용하는 것으로 합니다.

아래 커맨드를 이용하여 /proc 파일시스템(/proc/sys/kernel/sem)에 변경 사항을 직접 적용하면 서버를 리부팅 하지 않고도 세마포어 매개변수를 변경할 수 있습니다.

```
# sysctl -w kernel.sem="250 32000 100 128"
```

그런 다음, /etc/sysctl.conf 스타트업 파일에 커널 매개변수를 추가하여 변경 사항을 영구적으로 적용합니다:

```
# echo "kernel.sem=250 32000 100 128" >> /etc/sysctl.conf
```

파일 핸들의 처리

Red Hat Linux 서버를 구성하는 과정에서, 사용 가능한 파일 핸들(file handle)의 수가 충분한지 확인하는 것은 매우 중요합니다. 파일 핸들의 최대 개수 설정에 따라 Linux 시스템에서 동시에 오픈할 수 있는 파일의 수가 달라지게 됩니다.

전체 시스템에서 사용 가능한 파일 핸들의 최대값을 확인하기 위해 아래 명령을 실행합니다:

```
# cat /proc/sys/fs/file-max
102563
```

오라클은 전체 시스템의 파일 핸들 수를 최소 65536개 이상으로 설정할 것을 권고하고 있습니다.

아래 커맨드를 이용하여 /proc 파일시스템(/proc/sys/kernel/file-max)에 변경 사항을 직접 적용하면 서버를 리부팅 하지 않고도 매개변수를 변경할 수 있습니다:

```
# sysctl -w fs.file-max=65536
```

그런 다음, /etc/sysctl.conf 스타트업 파일에 커널 매개변수를 추가하여 변경 사항을 영구적으로 적용합니다:

```
# echo "fs.file-max=65536" >> /etc/sysctl.conf
```

현재 사용 중인 파일 핸들의 수는 아래와 같이 조회할 수 있습니다:

```
# cat /proc/sys/fs/file-nr
825 0 65536
```

file-nr 파일에서 표시되는 3 개의 매개변수가 다음과 같습니다: 할당된 파일 핸들의 수, 현재 사용 중인 파일 핸들의 수, 할당 가능한 최대 파일 핸들의 수.

(참고: /proc/sys/fs/file-max의 값을 증가시키고자 하는 경우, 먼저 ulimit이 올바르게 설정되었는지 확인할 필요가 있습니다. 2.4.20 버전에서는 ulimit이 "unlimited"로 설정되는 것이 일반적입니다. ulimit 커맨드를 사용하면 ulimit 설정을 확인할 수 있습니다:

```
# ulimit
unlimited
```

IP 로컬 포트 범위의 확인

시스템이 1024 - 65000 사이의 로컬 포트를 허용하도록 설정합니다.

ip_local_port_range의 값을 확인하기 위한 커맨드가 아래와 같습니다:

```
# cat /proc/sys/net/ipv4/ip_local_port_range
32768 61000
```

ip_local_port_range의 디폴트 값은 32768 - 61000으로 설정됩니다. 오라클은 1024 - 65000의 로컬 포트 범위를 사용할 것을 권장하고 있습니다.

아래 커맨드를 이용하여 /proc 파일시스템(/proc/sys/net/ipv4/ip_local_port_range)에 변경 사항을 직접 적용하면 서버를 리부팅 하지 않고도 매개변수를 변경할 수 있습니다:

```
# sysctl -w net.ipv4.ip_local_port_range="1024 65000"
```

그런 다음, /etc/sysctl.conf 스타트업 파일에 커널 매개변수를 추가하여 변경 사항을 영구적으로 적용합니다:

```
# echo "net.ipv4.ip_local_port_range = 1024 65000" >> /etc/sysctl.conf
```

oracle 사용자의 Shell Limit 설정

오라클은 Linux 시스템의 소프트웨어 성능 개선을 위해 아래와 같이 오라클 사용자의 shell limit을 증가시킬 것을 권고하고 있습니다.

Shell Limit	limits.conf 항목	Hard Limit
open file descriptor의 최대 수	nofile	65536
개별 사용자가 사용 가능한 프로세스의 최대 수	nproc	16384

변경 작업을 위해 root 계정으로 아래와 같이 실행합니다:

```
cat >> /etc/security/limits.conf <<EOF
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
EOF
```

```
cat >> /etc/pam.d/login <<EOF
session required /lib/security/pam_limits.so
EOF
```

"oracle" UNIX 계정의 디폴트 셸 스타트업 파일을 업데이트합니다.

Bourne, Bash, Korn 셸의 경우 아래와 같이 실행하여 /etc/profile 파일에 설정 항목을 추가합니다:

```
cat >> /etc/profile <<EOF
if [ $USER = "oracle" ]; then
  if [ $SHELL = "/bin/ksh" ]; then
    ulimit -p 16384
```

```

ulimit -n 65536
else
ulimit -u 16384 -n 65536
fi
umask 022
fi
EOF

```

C 셸(csh, tcsh)의 경우 아래와 같이 실행하여 /etc/csh.login 파일에 설정 항목을 추가합니다.

```

cat >> /etc/csh.login <<EOF
if ( W$USER == "oracle" ) then
  limit maxproc 16384
  limit descriptors 65536
endif
EOF

```

시스템의 전체 커널 매개변수 활성화

지금까지 오라클 설치 및 구성 과정에서 튜닝이 필요한 Linux 커널 매개변수들을 모두 살펴 보았습니다. 그리고 각각의 항목에 대해, /etc/sysctl.conf 파일에 매개변수 정의를 추가함으로써 변경 사항이 영구적으로 적용될 수 있도록 하였습니다.

이 시점에 리부트를 수행하여 모든 매개변수 변경 사항을 활성화할 수도 있습니다. 또는 root 계정으로 아래 커맨드를 이용하여 /etc/sysctl.conf 파일을 실행해도 됩니다. 이 작업은 클러스터의 모든 노드에서 실행되어야 합니다!

```

# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 262144
net.core.wmem_max = 262144
kernel.shmmax = 2147483648
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000

```

모든 클러스터 노드에 정확한 날짜 및 시간 설정

Oracle Clusterware, 데이터베이스, 컴패니언 CD의 설치 과정에서, Oracle Universal Installer(OUI)는 가장 먼저 인스톨러가 실행되는 로컬 노드(linux1)에 소프트웨어를 설치합니다. 설치된 소프트웨어는 클러스터의 나머지 노드(linux2)로 원격 복사됩니다. 원격 복사 과정에서, OUI는 각 원격 노드에 UNIX tar 커맨드를 실행하여 복사된 파일의 압축을 풉니다. 설치를 진행하는 로컬 노드의 날짜/시간이 원격 노드보다 빠르게 설정된 경우, OUI는 tar 커맨드 실행 과정에서 에러를 발생시키고 파일의 타임스탬프가 미래 시점으로 설정되어 있다는 메시지를 띄웁니다:

```

Error while copying directory
  /u01/app/oracle/product/crs with exclude file list 'null' to nodes 'linux2'.
[PRKC-1002 : All the submitted commands did not execute successfully]
-----
linux2:
  /bin/tar: ./bin/lsnodes: time stamp 2006-09-13 09:21:34 is 735 s in the future
  /bin/tar: ./bin/olsnodes: time stamp 2006-09-13 09:21:34 is 735 s in the future
  ...(more errors on this node)

```

이 메시지가 심각한 에러처럼 보일 수도 있습니다. 하지만 실제로는 단순한 경고에 불과합니다. 실제로 "tar" 커맨드는 정상적으로 실행됩니다. 하지만 (ls -l 커맨드를 이용하여) 원격 노드의 파일 목록을

확인해 보면, 파일의 타임스탬프가 미래가 아닌 현재 시점이 될 때까지 time 필드가 공백으로 비워져 있음을 확인할 수 있습니다.

아래에 설명된 설치 작업을 시작하기 전에, 클러스터의 각 멤버 노드가 동일한 날짜와 시간으로 설정 되어 있음을 확인해야 합니다. 오라클은 대부분의 운영 체제가 지원하는 *Network Time Protocol*을 사용할 것을 권고하고 있습니다.

하지만 경우에 따라서는 Network Time Protocol 서버의 사용이 불가능할 수도 있습니다. 클러스터의 날짜와 시간을 수동으로 설정해야 하는 경우에는, 소프트웨어 설치가 진행되는 로컬 노드(linux1)의 시간이 클러스터의 다른 노드(linx2)에 설정된 시간보다 늦도록 설정하는 것이 좋습니다. 필자의 경우 보통 20 초의 시간차를 두는 방법을 씁니다:

linux1의 날짜/시간 설정:

```
# date -s "9/13/2006 23:00:00"
```

linux2의 날짜/시간 설정:

```
# date -s "9/13/2006 23:00:20"
```

본 문서에서 설명된 2-노드 RAC 구성은 Network Time Protocol 서버를 사용하고 있지 않습니다.

13. hangcheck-timer 커널 모듈의 설정

클러스터의 모든 Oracle RAC 노드에서 아래 작업을 수행합니다!

Oracle 9i Release 1(9.0.1)과 Oracle 9i Release 2 (9.2.0.1)는 *watchdog*라는 이름의 데몬을 이용하여 클러스터의 상태를 모니터링하고 장애가 발생한 경우 RAC를 재시작합니다. Oracle 9i Release 2 (9.2.0.2)부터 이 데몬의 역할은 hangcheck-timer라는 Linux 커널 모듈에 의해 대체되었습니다. hangcheck 타이머는 Linux 커널에 로드된 상태로 타이머를 설정하고 일정 시간 이후 타이머를 점검하는 방식으로 시스템에 "hang"의 발생 여부를 점검합니다. hang-check의 임계치를 설정하고, 이 임계치를 넘어서는 경우 시스템을 리부트 하도록 설정하는 것이 가능합니다. Oracle Clusterware(*Cluster Manager*)가 hangcheck-timer 모듈을 반드시 요구하는 것은 아니지만, 오라클은 이 모듈의 사용을 강력하게 권장하고 있습니다.

hangcheck-timer.ko 모듈

hangcheck-timer 모듈은 커널 기반 타이머를 사용하여 시스템 태스크 스케줄러에 발생하는 딜레이(delay)를 정기적으로 점검하는 방식으로 시스템 상태를 확인합니다. 시스템이 중단되거나 "hang" 상태가 되는 경우 타이머는 노드를 리셋합니다. hangcheck-timer 모듈은 Time Stamp Counter (TSC) CPU 레지스터를 사용하며, 이 레지스터는 매 클럭 시그널(clock signal) 단위로 증가됩니다. TCS는 하드웨어에 의해 자동으로 업데이트되므로 매우 정확한 시간 측정이 가능합니다.

hangcheck-timer 프로젝트에 관련한 보다 상세한 정보는 [here](#) 에서 확인할 수 있습니다.

hangcheck-timer.ko 모듈의 설치

hangcheck-timer는 그간 오라클이 독자적으로 제공해 왔으나, 커널 버전 2.4.9-e.12 이후부터는 Red Hat Linux AS에도 기본 포함된 형태로 제공되기 시작했습니다. 현재 환경에도 hangcheck-timer가 이미 포함되어 있을 것입니다. 아래 명령을 사용하여 모듈이 포함되어 있는지 확인합니다. :

```
# find /lib/modules -name "hangcheck-timer.ko"
/lib/modules/2.6.9-42.EL/kernel/drivers/char/hangcheck-timer.ko
```

hangcheck timer 오브젝트(hangcheck-timer.ko)가 /lib/modules/2.6.9-42.EL/kernel/drivers/char 디렉토리에 위치하고 있음을 알 수 있습니다.

hangcheck-timer 모듈의 설정 및 로드

hangcheck-timer 모듈에는 두 가지 매개변수가 존재합니다:

hangcheck-tick: 이 매개변수는 시스템 상태의 모니터링 주기를 정의하는데 사용됩니다. 디폴트 값은 60초이며, 오라클은 30초의 설정을 권장합니다.

hangcheck-margin: 이 매개변수는 hangcheck-timer가 RAC 노드를 리셋하기 전까지 허용되는 최대 지연시간을 정의하는데 사용됩니다. 지연 시간은 초 단위로 설정됩니다. 디폴트 값은 180초이며, 오라클은 디폴트 값을 그대로 사용하는 것을 권장합니다.

참고: 두 가지 hangcheck-timer 매개변수를 통해 (hang이 발생한 상황에서) RAC 노드가 리셋 되기까지 걸리는 시간이 결정됩니다. 아래 조건을 만족하는 경우 노드 리셋이 발생하게 됩니다:

```
system hang time > (hangcheck_tick + hangcheck_margin)
```

Hangcheck 커널 모듈 매개변수의 설정

hangcheck-timer 커널 모듈이 (수동으로, 또는 오라클을 통해 자동으로) 로드 될 때마다, 위에서 설명한 두 가지 매개변수 (*hangcheck_tick*, *hangcheck_margin*)의 설정값을 확인하게 됩니다. 이 값은 Linux 서버가 리부트 될 때마다 확인됩니다. 따라서 */etc/modprobe.conf* 파일에 아래와 같이 정확한 값을 입력해 주어야 합니다:

```
# su -
# echo "options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180" >> /etc/modprobe.conf
```

이제 hangcheck-timer 커널 모듈이 로드될 때마다, */etc/modprobe.conf* 파일에 정의된 설정값이 사용됩니다.

테스트를 위해 Hangcheck 커널 모듈 수동으로 로드하기

오라클은 필요할 때마다 hangcheck-timer 커널 모듈을 자동으로 로드합니다. 그렇기 때문에, modprobe 또는 insmod 커맨드를 이용하여 스타트업 파일(예: */etc/rc.local*)에 hangcheck-timer 모듈을 별도로 추가할 필요는 없습니다.

필자는 (단순히 습관적으로) hangcheck-timer 커널 모듈의 modprobe를 */etc/rc.local* 파일에 추가하곤 합니다. 이것은 지극히 개인적인 습관의 문제이지만, 이처럼 hangcheck-timer의 modprobe를 스타트업 파일에 추가한다고 해서 문제가 될 것은 없습니다.

그래서 필자는 구성에 완벽을 기하고 밤잠을 편히 자자는 욕심에서 hangcheck-timer 커널 모듈을 모든 스타트업 파일에 아래와 같이 등록하는 방법을 사용합니다:

```
# echo "/sbin/modprobe hangcheck-timer" >> /etc/rc.local
```

(참고: 리부팅할 때마다 modprobe 또는 insmod를 이용하여 hangcheck-timer 커널 모듈을 수동으로 로드할 필요는 없습니다. 오라클은 필요할 때마다 hangcheck-timer 모듈을 자동으로 로드합니다.)

다음으로, */etc/modprobe.conf* 파일에 정의된 매개변수가 hangcheck-timer 커널 모듈에 올바르게 적용되고 있는지 테스트해 봅시다. 이를 위해 modprobe 커맨드를 사용합니다. hangcheck-timer 커널 모듈을 로드하는 시점에 매개변수를 전달하는 방법을 사용할 수도 있지만(예: `insmod hangcheck-timer hangcheck_tick=30 hangcheck_margin=180`), 여기에서는 */etc/modprobe.conf* 파일에 설정된 내용을 이용하는 방법을 사용하기로 합니다.

hangcheck-timer 커널 모듈을 수동으로 로드하고 */etc/modprobe.conf* 파일에 정의된 매개변수 값이 올바르게 적용되었는지 확인하기 위해, 아래와 같이 실행합니다:

```
# su -
# modprobe hangcheck-timer
# grep Hangcheck /var/log/messages | tail -2
Oct 10 19:56:24 linux1 kernel: Hangcheck: starting hangcheck timer 0.9.0 (tick is 30 seconds, margin is 180 seconds).
Oct 10 19:56:24 linux1 kernel: Hangcheck: Using monotonic_clock().
```

14. 원격 액세스를 위한 RAC 노드 설정

클러스터의 모든 Oracle RAC 노드에서 아래 작업을 수행합니다!

Oracle Real Application 클러스터를 설치하고 사용하기 전에, 모든 클러스터 노드의 "oracle" UNIX 사용자 계정에 대해 SSH(secure shell) 또는 RSH(remote shell)를 설정해 주어야 합니다. 여기서 목표는 "oracle" UNIX 사용자 계정과 "동일한 계정 (*user equivalence*)"을 셋업하는 것입니다. "oracle" UNIX 사용자 계정은 SSH 또는 RSH를 이용하여 (패스워드를 입력하지 않고도) 클러스터의 다른 노드에 접근하고 커맨드를 실행하거나 파일을 복사할 수 있습니다. SSH 또는 RSH 모두 사용 가능하지만, SSH가 보다 선호되는 방법으로 이용됩니다. 오라클은 10g Release 1부터 SSH 툴을 추가로 지원하고 있습니다. Oracle10g 이전 버전에서는 RSH를 사용해야 합니다.

10g의 Oracle Universal Installer가 SSH 툴(ssh, scp)의 존재를 감지하지 못하는 경우 RSH 툴(rsh, rcp)을 사용하게 됨을 참고하시기 바랍니다.

그렇다면 RSH 또는 SSH 설정이 왜 필요한 것일까요? Oracle Clusterware와 Oracle Database 소프트웨어의 설치 작업은 RAC 클러스터 노드 중 하나에서만 실행됩니다. Oracle Universal Installer(OUI)를 특정 노드에서 실행하면 ssh, scp 커맨드(또는 rsh, rcp 커맨드)를 이용하여 RAC 클러스터의 다른 노

드에 오라클 소프트웨어를 복사하는 작업이 수행됩니다. 따라서 OUI(runInstaller)를 실행하는 "oracle" UNIX 계정에 대해 다른 노드와의 트러스트 관계가 성립되어 있어야 합니다. 다시 말해 OUT가 실행되는 서버에서 별도의 패스워드를 입력하지 않고도 다른 모든 서버에 대해 SSH 커맨드(ssh, scp) 또는 RSH 커맨드(rsh, rcp)를 실행할 수 있어야 합니다.

일반적인 RAC 운영 모드에서는 SSH 또는 RSH가 사용되지 않는다는 점을 참고할 필요가 있습니다. 하지만 클러스터 데이터베이스 설치 뿐 아니라 패치 설치 과정에서도 SSH/RSH가 사용되기 때문에, 이 설정은 상시 유지되어야 합니다.

제일 먼저 SSH와 RSH 중 어떤 방법을 사용할 것인지 결정해야 합니다. 두 가지 모두 나름의 장점과 단점이 있습니다. RSH의 경우 설정이 매우 편리하다는 장점이 있습니다. RSH는 설정이 비교적 간단하며 트러스티드 노드(trusted node; 설치 작업을 진행하는 노드)에서 로그인할 때 언제나 터미널 세션을 사용할 수 있습니다. 하지만 원격 노드로의 연결을 위한 보안 기능이 제공되지 않는다는 점이 문제입니다. SSH는 연결을 위한 보안 메커니즘을 제공하지만, 작업 과정이 복잡합니다. 또 오라클 사용자가 트러스티드 노드에 접근할 때마다 매번 터미널 세션에서 SSH를 활성화해 주어야 합니다. 오라클의 공식제품문서에서는 SSH의 설정 방법만을 설명하고 있으며 SSH를 보다 선호되는 방법으로 권장하고 있습니다.

SSH, RSH의 설정 방법이 아래와 같습니다:

(SSH)Secure Shell의 사용

(RSH)Remote Shell의 사용

SSH(Secure Shell)의 사용

OpenSSH 버전 3의 설정 방법이 아래와 같습니다.

아래 커맨드를 사용하여 SSH가 설치되어 실행 중인지 확인합니다:

```
# pgrep sshd
2808
```

SSH가 실행되고 있다면 위에서처럼 sshd 프로세스의 ID 넘버가 표시될 것입니다. 이 커맨드를 클러스터의 모든 노드에서 실행하여 SSH 데몬의 실행 여부를 확인하십시오!

SSH에 대한 자세한 정보는 man 페이지를 통해 확인할 수 있습니다:

```
# man ssh
```

두 Oracle RAC 노드에 RSA, DSA 키 생성하기

SSH를 설정하려면 먼저 클러스터의 모든 Oracle RAC 노드에 RSA/DSA 키를 생성해야 합니다. 커맨드를 실행하면 RSA, DSA를 위한 퍼블릭/프라이빗 키가 생성됩니다. (각 노드별로 총 4 개의 키가 생성됩니다.) 그런 다음 RSA, DSA 퍼블릭 키의 콘텐츠를 *authorized key file*에 복사하고, 이 파일을 클러스터의 전체 노드에 배포해야 합니다.

RSA/DSA 키를 생성하는 방법이 아래와 같습니다. 이 작업을 클러스터의 모든 Oracle RAC 노드에서 실행해야 합니다.

1. "oracle" UNIX 사용자 계정으로 로그인합니다.

```
# su - oracle
```

2. 필요한 경우, "oracle" 사용자의 홈 디렉토리에 .ssh 디렉토리를 생성하고 권한을 설정해 줍니다.

```
$ mkdir -p ~/.ssh
$ chmod 700 ~/.ssh
```

3. 아래 커맨드를 실행하여 SSH 프로토콜 버전 3를 위한 RSA 키(퍼블릭/프라이빗 키)를 생성합니다.

```
$ /usr/bin/ssh-keygen -t rsa
```

프롬프트에서:

키 파일의 디폴트 위치를 승인합니다.

pass phrase를 입력하고 다시 확인 입력합니다. 패스워드는 "oracle" UNIX 사용자의 패스워드와 다르게 설정하는 것이 권장됩니다. (강제 사항은 아닙니다.)

이 커맨드를 실행하면 ~/.ssh/id_rsa.pub 파일에 퍼블릭 키가,

~/.ssh/id_rsa 파일에 프라이빗 키가 기록됩니다. 어떤 경에도 프라이빗 키는 배포해서는 안됨을 명심하십시오!

4. 아래 커맨드를 실행하여 SSH 프로토콜 버전 3를 위한 DSA 키(퍼블릭/프라이빗 키)를 생성합니다.

```
$ /usr/bin/ssh-keygen -t dsa
```

프롬프트에서:

키 파일의 디폴트 위치를 승인합니다.

pass phrase를 입력하고 다시 확인 입력합니다. 패스워드는 "oracle" UNIX 사용자의 패스워드와 다르게 설정하는 것이 권장됩니다. (강제 사항은 아닙니다.)

이 커맨드를 실행하면 ~/.ssh/id_dsa.pub 파일에 퍼블릭 키가,

~/.ssh/id_dsa 파일에 프라이빗 키가 기록됩니다. 어떤 경에도 프라이빗 키는 배포해서는 안됨을 명심하십시오!

5. 위 작업을 클러스터의 모든 Oracle RAC 노드에서 실행합니다. 모든 Oracle RAC 노드에서 RSA, DSA를 위한 퍼블릭/프라이빗 키를 생성한 다음, 노드 중 하나에서 *authorized key file*을 생성합니다. *authorized key file*이란 모든 노드의 RSA, DSA 퍼블릭 키 사본을 저장하는 파일을 의미합니다. *authorized key file*에 모든 퍼블릭 키를 저장한 뒤에는, 클러스터의 다른 노드에 이를 배포하는 작업이 수행됩니다. 클러스터의 노드 중 하나에서 아래 작업을 수행하고 작성된 *authorized key file*을 배포합니다. 본 예제에서는 linux1에서 작업을 수행하기로 합니다:
 1. 먼저, *authorized key file*이 노드에 이미 존재하는지 확인합니다 (~/.ssh/authorized_keys). 새로운 환경에 Oracle RAC를 설치하는 경우라면 이 파일이 존재하지 않을 가능성이 높습니다. 파일이 존재하지 않으면 새로 생성합니다:

```
$ touch ~/.ssh/authorized_keys
```

```
$ cd ~/.ssh
```

```
$ ls -l *.pub
```

```
-rw-r--r-- 1 oracle dba 603 Aug 31 23:40 id_dsa.pub
```

```
-rw-r--r-- 1 oracle dba 223 Aug 31 23:36 id_rsa.pub
```

위 목록에서 이제 앞 섹션에서 생성한 id_rsa.pub, id_dsa.pub 퍼블릭 키를 확인할 수 있을 것입니다.

2. SSH를 이용하여 양쪽 Oracle RAC 노드의 ~/.ssh/id_rsa.pub 콘텐츠와 ~/.ssh/id_dsa.pub 퍼블릭 키를 방금 생성한 *authorized key file* (~/.ssh/authorized_keys)에 복사합니다. 이 작업 역시 linux1에서 실행합니다. 두 Oracle RAC 노드의 "oracle" UNIX 사용자 패스워드를 묻는 프롬프트가 표시됩니다. SSH를 이용하여 현재 로그인 중인 노드(linux1)에 최초 접근을 시도하면, "oracle" UNIX 사용자 계정의 패스워드를 묻는 프롬프트가 표시됩니다. 하지만 두 번째로 노드 접근을 시도하면 프라이빗 키의 암호를 풀기 위한 pass phrase를 묻는 프롬프트가 뜹니다. 반면, 다른 노드에서는 언제나 "oracle" UNIX 사용자 계정의 패스워드를 묻습니다. 아래 실행 예제는 linux1에서 두 개의 클러스터 노드(linux1, linux2)에 대해 실행되는 환경을 가정하고 있습니다:

```
$ ssh linux1 cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

```
The authenticity of host 'linux1 (192.168.1.100)' can't be established.
```

```
RSA key fingerprint is 61:8a:f9:9e:28:a2:b7:d3:70:8d:dc:76:ca:d9:23:43.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'linux1,192.168.1.100' (RSA) to the list of known hosts
```

```
oracle@linux1's password: xxxxxx
```

```
$ ssh linux1 cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

```

Enter passphrase for key '/u01/app/oracle/.ssh/id_rsa': xxxxx
$ ssh linux2 cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
The authenticity of host 'linux2 (192.168.1.101)' can't be established.
RSA key fingerprint is 84:2b:bd:eb:31:2c:23:36:55:c2:ee:54:d2:23:6a:e4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'linux2,192.168.1.101' (RSA) to the list of known hosts
oracle@linux2's password: xxxxxx
$ ssh linux2 cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
oracle@linux2's password: xxxxxx

```

참고: SSH를 이용하여 처음으로 노드에 접근을 시도하는 경우 아래와 같은 메시지가 표시될 수 있습니다:

```

The authenticity of host 'linux1 (192.168.1.100)' can't be established.
RSA key fingerprint is 61:8a:f9:9e:28:a2:b7:d3:70:8d:dc:76:ca:d9:23:43.
Are you sure you want to continue connecting (yes/no)? yes

```

프롬프트에서 yes를 입력하여 다음 단계로 진행됩니다. 이후 같은 시스템에서 같은 노드로 접근하면 위 메시지가 다시 표시되지 않을 것입니다.

- 이제 linux1의 authorized key file(~/.ssh/authorized_keys)에 클러스터 내의 모든 노드를 위한 RSA/DSA 퍼블릭 키를 확보하였습니다. 이 파일을 클러스터의 다른 노드로 복사해야 합니다. 본 문서의 예제에서는 linux2에만 파일을 복사하면 됩니다. scp 커맨드를 이용하여 authorized key file을 클러스터의 나머지 노드에 복사합니다:

```

$ scp ~/.ssh/authorized_keys linux2:~/.ssh/authorized_keys
oracle@linux2's password: xxxxxx
authorized_keys 100% 1652 1.6KB/s 00:00

```

- 각각의 Oracle RAC 노드에 로그인 한 후 아래와 같이 실행하여 authorized key file의 권한을 변경합니다:

```
$ chmod 600 ~/.ssh/authorized_keys
```

- 이제 ssh를 이용하여 다른 노드에 로그인하거나 다른 노드서 커맨드를 실행하면 DSA 키를 생성할 때 정의한 pass phrase를 묻는 프롬프트가 뜹니다. linux1에서의 ssh 실행 예가 아래와 같습니다:

```

$ ssh linux1 hostname
Enter passphrase for key '/u01/app/oracle/.ssh/id_rsa': xxxxxx
linux1
$ ssh linux2 hostname
Enter passphrase for key '/u01/app/oracle/.ssh/id_rsa': xxxxxx
linux2

```

참고: 위에서 호스트네임을 제외한 다른 메시지나 텍스트가 표시되는 경우, 오라클 설치 작업이 실패할 수 있습니다. 따라서 위 커맨드를 실행했을 때 호스트네임 이외의 다른 텍스트가 표시되지 않도록 작업해 주어야 합니다. 로그인 스크립트를 통해 다른 출력이 표시되거나 프롬프트가 표시되는 경우, 인터랙티브 셸 환경에서만 이 스크립트가 실행되도록 수정해 줍니다.

현재 셸 세션을 위한 SSH User Equivalency의 활성화

OUI는 pass phrase를 입력하지 않고도 SSH 툴 커맨드(ssh, scp)를 실행할 수 있는 환경을 요구합니다. 클러스터의 모든 노드에 SSH가 설정되어 있더라도, SSH 툴 커맨드를 실행하면 pass phrase를 묻는 프롬프트가 표시됩니다. 그러므로 OUI를 실행하기 전에, OUI를 실행하는 터미널 세션에 대해 "user equivalence"를 활성화해 주어야 합니다. 본 문서의 예제에서는 모든

오라클 설치 작업이 linux1에서 실행되는 것으로 가정합니다.

오라클 설치 작업을 실행할 노드에서 로그아웃하거나 다시 로그인할때, 터미널 셸 세션에 대해 'user equivalence'를 활성화해 주어야 합니다(이 작업은 디폴트로 실행되지 않습니다).

현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행합니다:

1. OUI를 실행할 노드(linux1)에 "oracle" UNIX 사용자 계정으로 로그인합니다.

```
# su - oracle
```

2. 아래 커맨드를 실행합니다:

```
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
```

프롬프트에서 생성한 각 키에 대한 pass phrase를 입력합니다.

3. SSH가 정상적으로 설정되었다면, 앞으로 이 터미널 세션에서 패스워드나 pass phrase를 입력하지 않고도 ssh, scp 커맨드를 실행할 수 있을 것입니다:

```
$ ssh linux1 "date;hostname"
Wed Sep 13 17:35:30 EDT 2006
linux1
$ ssh linux2 "date;hostname"
Wed Sep 13 17:36:14 EDT 2006
linux2
```

참고: 위 커맨드를 실행하면 호스트네임과 함께 양쪽 Oracle RAC 노드에 설정된 시간이 표시됩니다. 노드 중 하나가 패스워드 또는 pass phrase 입력을 요구하는 경우, 해당 노드에서 퍼블릭 키가 저장된 ~/.ssh/authorized_keys 파일을 확인합니다. 위에서 시간과 호스트네임을 제외한 다른 메시지나 텍스트가 표시되는 경우, 오라클 설치 작업이 실패할 수 있습니다. 따라서 위 커맨드를 실행했을 때 호스트네임, 날짜 이외의 다른 텍스트가 표시되지 않도록 작업해 주어야 합니다. 로그인 스크립트를 통해 다른 출력이 표시되거나 프롬프트가 표시되는 경우, *인터랙티브 셸* 환경에서만 이 스크립트가 실행되도록 수정해 줍니다.

4. Oracle Universal Installer는 X Server를 이용한 GUI 인터페이스를 사용합니다. 'user equivalence'를 활성화한 터미널 세션(오라클 설치 작업을 수행할 노드)에서 DISPLAY 환경 변수를 X Windows 디스플레이가 가능하도록 설정해야 합니다:

Bourne, Korn, Bash 셸:

```
$ DISPLAY=<Any X-Windows Host>:0
$ export DISPLAY
```

C 셸:

```
$ setenv DISPLAY <Any X-Windows Host>:0
```

DISPLAY 변수의 설정을 변경한 후, 현재 터미널 세션에 *X11 forwarding*이 활성화되어 있지 않음을 확인합니다.

```
$ ssh linux1 hostname
linux1
```

```
$ ssh linux2 hostname
linux2
```

참고: 원격 클라이언트를 이용해서 설치를 진행할 노드에 연결하는 경우, 아래와 같은 메시지를 확인할 수 있습니다: "Warning: No xauth data: using fake authentication data for X11 forwarding." 이 메시지는 authorized keys file이 올바르게 설정되어 있지만 SSH 설정에서 X11 forwarding이 활성화되어 있음을 의미합니다. 실행 예가 아래와 같습니다:

```
$ export DISPLAY=melody:0
$ ssh linux2 hostname
Warning: No xauth data: using fake authentication data for X11 forwarding.
linux2
```

X11 Forwarding이 활성화되어 있으면 오라클 설치 작업이 실패함을 명심하시기 바랍니다. 이 문제를 해결하려면 "oracle" UNIX 사용자 계정을 위한 사용자 레벨 SSH 클라이언트 설정 파일을 생성하고 X11 Forwarding을 비활성화해야 합니다:

텍스트 편집기를 이용하여 ~/.ssh/config 파일을 편집 또는 생성합니다. ForwardX11 속성이 "no"로 설정되어 있는지 확인합니다. 한 예로, ~/.ssh/config 파일에 아래와 같이 입력할 수 있습니다:

```
Host *
  ForwardX11 no
```

- 반드시 이 터미널 세션에서 Oracle Universal Installer를 실행해야 합니다. 다른 터미널 세션에서 OUI를 실행하려 한다면 'user equivalence' 활성화 작업(이 섹션의 2,3,4 단계)을 반복해 주어야 합니다.

stty 커맨드의 제거

오라클 소프트웨어를 설치하는 과정에서, 시스템 상의 숨겨진 파일(예: .bashrc, .cshrc, .profile)에 stty 커맨드가 포함되어 있으면 설치 작업이 실패할 수 있습니다.

이 문제를 예방하려면, 이 파일들을 수정하여 STDERR를 통해 아무 것도 출력되지 않도록 조치해야 합니다:

Bourne, Bash, Korn 셸:

```
if [ -t 0 ]; then
  stty intr ^C
fi
```

C 셸:

```
test -t 0
if ($status == 0) then
  stty intr ^C
endif
```

참고:stty 커맨드를 포함하는 숨겨진 파일이 SSH에 의해 로드되면, OUI는 에러를 발생시키고 설치를 종료합니다.

RSH(Remote Shell)의 사용

대부분의 Linux 시스템에서는 RSH(remote shell)이 제공하는 서비스를 기본적으로 비활성화하고 있습니다. 이번 섹션에서는 Oracle Universal Installer에서 RSH 툴을 이용하여 원격 노드의 명령을 실행하거나 원격 노드에 파일을 복사하는 경우에 'user equivalence'를 활성화하는 방법을 설명하고 있습니다. 이번 작업의 목표는 Oracle

Universal Installer가 rsh, rcp를 이용하여 (패스워드를 별도로 입력하지 않고도) 원격 노드의 명령을 실행하고 파일을 복사할 수 있게 하는 것입니다. 하지만 RSH를 이용한 방법은 보안 측면에서 취약하다는 사실을 명심하시기 바랍니다.

rsh 데몬은 /etc/hosts.equiv 파일 또는 사용자(oracle)의 홈 디렉토리에 위치한 .rhosts 파일을 이용하여 사용자를 확인합니다.

먼저, RAC 클러스터의 각 노드에 rsh RPM이 설치되어 있는지 확인해 봅시다:

```
# rpm -q rsh rsh-server
rsh-0.17-25.4
rsh-server-0.17-25.4
```

위 실행 결과를 통해 rsh와 rsh-server가 설치되어 있음을 확인할 수 있습니다. 만일 rsh 이 설치되어 있지 않다면, RPM이 위치한 CD에서 아래 명령을 실행합니다:

```
# su -
# rpm -ivh rsh-0.17-25.4.i386.rpm rsh-server-0.17-25.4.i386.rpm
```

"rsh" 서비스를 활성화하려면, /etc/xinetd.d/rsh 파일의 "disable" 속성을 "no"로 설정하고 xinetd를 다시 로드해야 합니다. 아래 명령을 클러스터의 모든 노드에서 실행합니다:

```
# su -
# chkconfig rsh on
# chkconfig rlogin on
# service xinetd reload
Reloading configuration: [ OK ]
```

전체 RAC 노드에 대해 "oracle" UNIX 계정의 트러스트(trust) 관계를 설정하기 위해, 클러스터의 모든 노드에 /etc/hosts.equiv 파일을 생성합니다:

```
# su -
# touch /etc/hosts.equiv
# chmod 600 /etc/hosts.equiv
# chown root.root /etc/hosts.equiv
```

이제 클러스터의 모든 노드에서 아래와 같은 방법으로 /etc/hosts.equiv 파일을 편집합니다:

```
# cat /etc/hosts.equiv
+linux1 oracle
+linux2 oracle
+linux1-priv oracle
+linux2-priv oracle
```

참고: 위의 예에서는 oracle 계정만이 rsh 커맨드를 실행할 수 있도록 설정되었습니다. 보안상의 이유로, /etc/hosts.equiv 파일은 root에 의해 소유되며 permission은 600으로 설정됩니다. 일부 시스템의 경우 파일의 owner가 root이고 permission이 600으로 설정된 경우에만 /etc/hosts.equiv 파일의 내용을 참고하도록 설정되어 있기도 합니다.

rsh 커맨드를 테스트하기 전에, rsh의 올바른 버전을 사용하고 있는지 확인하시기 바랍니다. Red Hat Linux는 디폴트로 /usr/kerberos/sbin을 \$PATH 변수의 앞부분에 위치시킵니다. 따라서 rsh의 Kerberos 버전이 우선적으로 실행됩니다.

필자는 보통 Kerberos 버전의 rsh를 rename하는 방법으로 일반 rsh 커맨드를 사용합니다.

```
# su -
# which rsh
```

```

/usr/kerberos/bin/rsh
# mv /usr/kerberos/bin/rsh /usr/kerberos/bin/rsh.original
# mv /usr/kerberos/bin/rcp /usr/kerberos/bin/rcp.original
# mv /usr/kerberos/bin/rlogin /usr/kerberos/bin/rlogin.original
# which rsh
/usr/bin/rsh

```

이제 연결을 테스트하고 Oracle Clusterware와 10g RAC를 설치하는 노드에서 rsh 커맨드를 실행할 수 있습니다. 본 문서의 예제에서는 linux1 서버에서 설치를 진행하는 것으로 합니다:

```

# su - oracle
$ rsh linux1 ls -l /etc/hosts.equiv
-rw----- 1 root root 68 Sep 27 23:37 /etc/hosts.equiv
$ rsh linux1-priv ls -l /etc/hosts.equiv
-rw----- 1 root root 68 Sep 27 23:37 /etc/hosts.equiv
$ rsh linux2 ls -l /etc/hosts.equiv
-rw----- 1 root root 68 Sep 27 23:45 /etc/hosts.equiv
$ rsh linux2-priv ls -l /etc/hosts.equiv
-rw----- 1 root root 68 Sep 27 23:45 /etc/hosts.equiv

```

SSH를 이용하는 경우와 달리, RSH의 'user equivalence'를 활성화하기 위해 별도의 조치를 수행할 필요가 없습니다. 터미널 세션에 성공적으로 로그인하면 "oracle" UNIX 사용자 계정의 'user equivalence'가 자동으로 활성화됩니다.

15. RAC 노드 별로 스타트업 커맨드 설정하기

클러스터의 모든 Oracle RAC 노드에서 아래 작업을 수행합니다!

지금까지 Oracle RAC 10g 구성에 필요한 매개변수와 리소스 설정 방법에 대해 상세하게 살펴보았습니다. 이 섹션에서는 앞에서 설명한 매개변수, 커맨드 등을 RAC 클러스터의 각 Linux 노드의 스타트업 스크립트(startup script)에 추가하는 방법을 설명합니다.

아래에 회색으로 표시된 항목이 각 스타트업 파일에 포함되도록 편집 작업을 수행합니다.

/etc/modprobe.conf

(모든 매개변수와 설정값은 커널 모듈에 의해 사용됩니다.)

```

alias eth0 b44
alias eth1 tulip
alias snd-card-0 snd-intel8x0
options snd-card-0 index=0
alias usb-controller ehci-hcd
alias usb-controller1 uhci-hcd
options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180

```

/etc/sysctl.conf

(여기에서는 서버간 연결을 위한 디폴트/최대 send buffer size와 디폴트/최대 receive buffer size를 모두 조정하였습니다. 이 파일에는 그 밖에도 오라클 인스턴스가 사용하는 공유 메모리, 세마포어, 파일 핸들 등에 관련한 매개변수가 포함되어 있습니다.)

```

# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification

```

```

net.ipv4.conf.default.rp_filter = 1

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Default setting in bytes of the socket receive buffer
net.core.rmem_default=262144

# Default setting in bytes of the socket send buffer
net.core.wmem_default=262144

# Maximum socket receive buffer size which may be set by using
# the SO_RCVBUF socket option
net.core.rmem_max=262144

# Maximum socket send buffer size which may be set by using
# the SO_SNDBUF socket option
net.core.wmem_max=262144

# -----
# | SHARED MEMORY |
# -----
kernel.shmmax=2147483648

# -----
# | SEMAPHORES |
# | ----- |
# | SEMMSL_value SEMMNS_value SEMOPM_value SEMMNI_value |
# | |
# -----
kernel.sem=250 32000 100 128

# -----
# | FILE HANDLES |
# -----
fs.file-max=65536

# -----
# | LOCAL IP RANGE |
# -----
net.ipv4.ip_local_port_range=1024 65000

```

/etc/hosts

(RAC 클러스터 노드의 모든 호스트네임/IP 주소를 입력합니다.)

```

# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
# Public Network - (eth0)
192.168.1.100 linux1
192.168.1.101 linux2
# Private Interconnect - (eth1)
192.168.2.100 linux1-priv
192.168.2.101 linux2-priv
# Public Virtual IP (VIP) addresses for - (eth0)

```

```
192.168.1.200 linux1-vip
192.168.1.201 linux2-vip
192.168.1.106 melody
192.168.1.102 alex
192.168.1.105 bartman
```

/etc/hosts.equiv

(/etc/hosts.equiv 파일은 RSH를 사용하는 경우에만 필요합니다. 별도의 패스워드 입력 없이 oracle 계정을 사용하여 각 노드에 로그인할 수 있도록 설정해 줍니다.)

```
+linux1 oracle
+linux2 oracle
+linux1-priv oracle
+linux2-priv oracle
```

/etc/rc.local

(hangcheck-timer 커널 모듈을 로드합니다.)

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
touch /var/lock/subsys/local
# +-----+
# | HANGCHECK TIMER |
# | (I do not believe this is required, but doesn't hurt) |
# +-----+
/sbin/modprobe hangcheck-timer
```

16. Oracle Cluster File System(OCFS2)의 설치 및 설정

여기에서 설명된 작업의 대부분은 클러스터의 모든 노드에서 실행되어야 합니다! 단 OCFS2 파일시스템의 생성 작업은, 클러스터 노드 중 하나에서만 실행합니다

이제 Oracle Cluster File System Release 2(OCFS2)를 설치할 차례입니다. OCFS2는 로우 디바이스(raw device)의 관리에 수반되는 부담을 경감하기 위한 목적에서 오라클이 개발하여 제공하는 클러스터 파일 시스템(Cluster File System)입니다. OCFS2는 일반 파일시스템과 동일한 표준적 기능과 인터페이스를 제공하며, 클러스터 애플리케이션의 관리 편의성을 개선해 줍니다.

2002년 12월, 오라클은 Oracle RAC 사용자들이 RAW 디바이스를 사용하지 않고 클러스터 데이터베이스를 실행할 수 있도록 하기 위한 목적에서 OCFS Release 1을 출시하였습니다. 이 파일시스템은 데이터 파일, 컨트롤 파일, 리두 로그, 아카이브 로그 등의 데이터베이스 관련 파일을 저장할 수 있는 구조로 설계되었습니다. OCFS2는 Oracle Cluster File System의 차세대 버전입니다. OCFS2는 일반적인 용도의 범용 클러스터 파일 시스템으로 설계되었습니다. OCFS2에서는 데이터베이스 관련 파일 뿐 아니라, 오라클 바이너리와 구성 파일(공유 Oracle Home) 등을 저장하는 것이 가능하므로, RAC의 관리가 한층 편리해졌습니다.

이 가이드에서는 OCFS2의 최신 버전(본 문서가 작성되는 시점의 최신 버전은 (OCFS2 Release 1.2.3-1)를 사용하여 Oracle Clusterware에 의해 공유되는 두 개의 파일을 저장하는 구성을 사용하고 있습니다. 이 두 가지 파일 이외에도 전체 Oracle RAC 인스턴스를 위한 공유 SPFILE 또한 OCFS2에 함께 저장됩니다.

Linux 환경에서의 OCFS2 설치에 관련한 자세한 정보가 필요하신 경우 this page를 참고하십시오.

OCFS2 다운로드

먼저 OCFS2 배포판의 최신 버전을 다운로드합니다. OCFS 배포판은 커널 모듈과 툴, 두 종류의 RPM으로 구성되어 있습니다. 최신 커널 모듈은 <http://oss.oracle.com/projects/ocfs2/files/> 에서, 툴은 <http://oss.oracle.com/projects/ocfs2-tools/files/> 에서 다운로드할 수 있습니다. .

최신 OCFS2 커널 모듈(드라이버)을 포함하는 RPM을 다운로드합니다. 필자는 CentOS 4.4 Enterprise Linux에 2.6.9-42.EL 버전의 커널을 사용하였습니다. 본 문서가 작성되는 시점에, 우리는 OCFS2를

위한 OCFS2 커널 모듈이 제공됨을 확인할 수 있었습니다 (OCFS2 Release 1.2.3-1). Linux 커널 2.6.9-42.EL을 위한 OCFS2 커널 모듈은 아래에서 확인하실 수 있습니다. 반드시 배포판, 플랫폼, 커널 버전, 그리고 커널 종류 smp, hugemem, psmtp 등)에 맞는 OCFS2 커널 모듈을 다운로드하시기 바랍니다.

```
ocfs2-2.6.9-42.EL-1.2.3-1.i686.rpm - (for single processor)
```

or

```
ocfs2-2.6.9-42.ELsmp-1.2.3-1.i686.rpm - (for multiple processors)
```

or

```
ocfs2-2.6.9-42.ELhugemem-1.2.3-1.i686.rpm - (for hugemem)
```

툴의 경우 플랫폼과 배포판만을 확인하면 됩니다. OCFS2 툴과 함께 OCFS2 콘솔 애플리케이션도 함께 다운로드하는 것을 잊지 마시기 바랍니다.

```
ocfs2-tools-1.2.1-1.i386.rpm - (OCFS2 tools)
```

```
ocfs2console-1.2.1-1.i386.rpm - (OCFS2 console)
```

OCFS2 Console은 옵션이지만 강력하게 권장됩니다. ocfs2console 애플리케이션 e2fsprogs, glib2 2.2.3 이상, vte 0.11.10 이상, pygtk2 (EL4) 또는 python-gtk (SLES9) 1.99.16 이상, python 2.3 이상, ocfs2-tool 등을 요구합니다.

어떤 OCFS2 driver release를 다운로드해야 하는지 선택하기 어려운 경우, 커널 버전과 일치하는 OCFS2 release를 선택하도록 합니다. 커널 버전을 확인하는 방법은 아래와 같습니다:

```
$ uname -a
```

```
Linux linux1 2.6.9-42.EL #1 Sat Aug 12 09:17:58 CDT 2006 i686 i686 i386 GNU/Linux
```

```
Linux linux1 2.6.9-42.EL #1 Sat Aug 12 09:17:58 CDT 2006 i686 i686 i386 GNU/Linux
```

싱글 프로세서 환경에서 실행 가능한 버전은 "EL" 뒤에 "smp"가 따라붙지 않습니다. 멀티 프로세서 환경에서 실행 가능한 버전은 "smp"라는 텍스트가 포함되어 있습니다.

OCFS2 설치

본 예제에서는 OCFS2 파일을 두 대의 싱글-프로세서 머신에 설치합니다. root 계정으로 클러스터의 모든 노드에서 아래 커맨드를 실행합니다:

```
$ su -
```

```
# rpm -Uvh ocfs2-2.6.9-42.EL-1.2.3-1.i686.rpm W
      ocfs2console-1.2.1-1.i386.rpm W
      ocfs2-tools-1.2.1-1.i386.rpm
```

```
Prepar ing... ##### [100%]
  1:ocfs2-tools      ##### [ 33%]
  2:ocfs2-2.6.9-42.EL ##### [ 67%]
  3:ocfs2console    ##### [100%]
```

SELinux의 비활성화 (RHEL4 U2 및 상위 버전)

RHEL4 U2 이후 버전을 사용하는 경우(CentOS 4.2는 RHEL4 U2에 기반하고 있습니다), SELinux가 활성화되어 있는 환경에서는 OCFS2가 정상적으로 동작하지 않음을 명심해야 합니다. RHEL4 U2 (CentOS 4.2 포함) 및 이후 버전을 사용하고 있는 경우 (system-config-securitylevel 툴을 이용하여) SELinux를 비활성화하고, O2CB 서비스가 실행 가능한 상태로 만들어 주어야 합니다.

SELinux를 비활성화하려면 "Security Level Configuration" GUI 유틸리티를 실행합니다:

```
# /usr/bin/system-config-securitylevel &
```

실행 결과로 아래 화면이 표시됩니다:

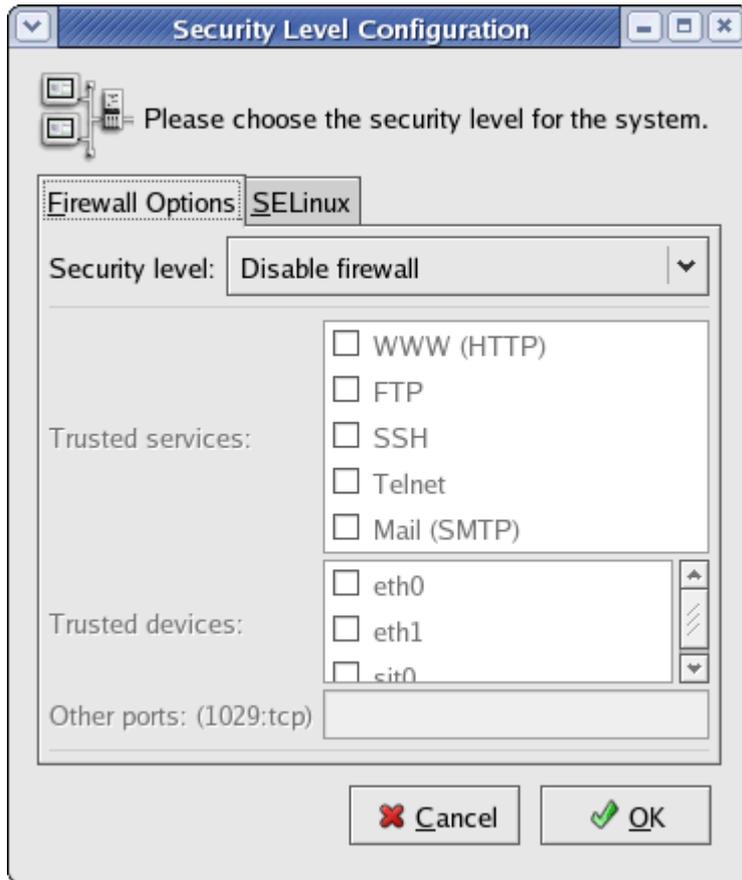


그림 13 Security Level Configuration 초기 화면

여기서 SELinux 탭을 클릭하고 “Enabled” 체크박스를 해제합니다. [OK]를 클릭하면 경고 대화상자가 뜹니다. [Yes]를 클릭하여 경고를 승인합니다. SELinux 옵션을 해제한 후의 화면이 아래와 같습니다:

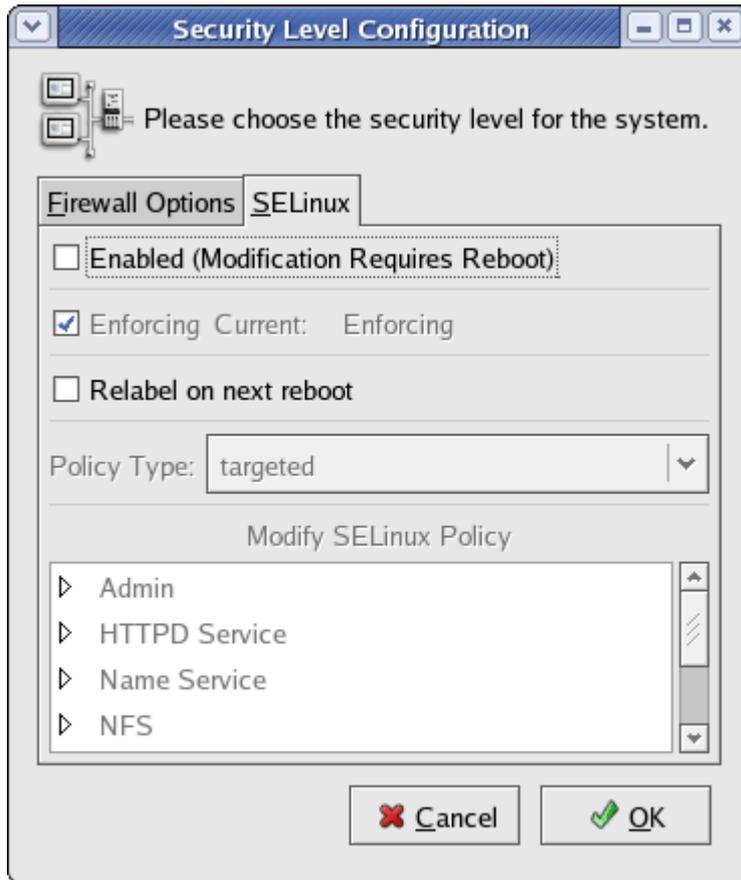


그림 14 SELinux 비활성화 화면

클러스터의 모든 노드에서 위 작업을 수행한 다음, 변경 사항을 적용하기 위해 시스템을 리부트합니다.

```
# init 6
```

OCFS2의 설정

다음으로 클러스터의 각 노드에 /etc/ocfs2/cluster.conf 파일을 생성하고 설정해 주어야 합니다. 가장 쉬운 방법은 GUI 툴 ocfs2console을 이용하는 것입니다. 본 구성 예제에서는, ocfs2console을 이용하여 /etc/ocfs2/cluster.conf 파일을 생성하고 설정하는 한편으로, 클러스터 스택 O2CB를 생성하고 시작하는 작업을 함께 수행하기로 합니다 (본 예제의 경우처럼) /etc/ocfs2/cluster.conf 파일이 존재하지 않는 경우, ocfs2console 툴은 ocfs2라는 디폴트 클러스터 이름을 사용하여 새로운 클러스터 스택 서비스(OC2B)와 함께 이 파일을 생성하는 작업을 수행합니다 이 작업은 루트 계정 권한으로 클러스터의 모든 노드에서 실행해 주어야 합니다:

```
$ su -
# ocfs2console &
```

실행 결과로 아래 화면이 표시됩니다:

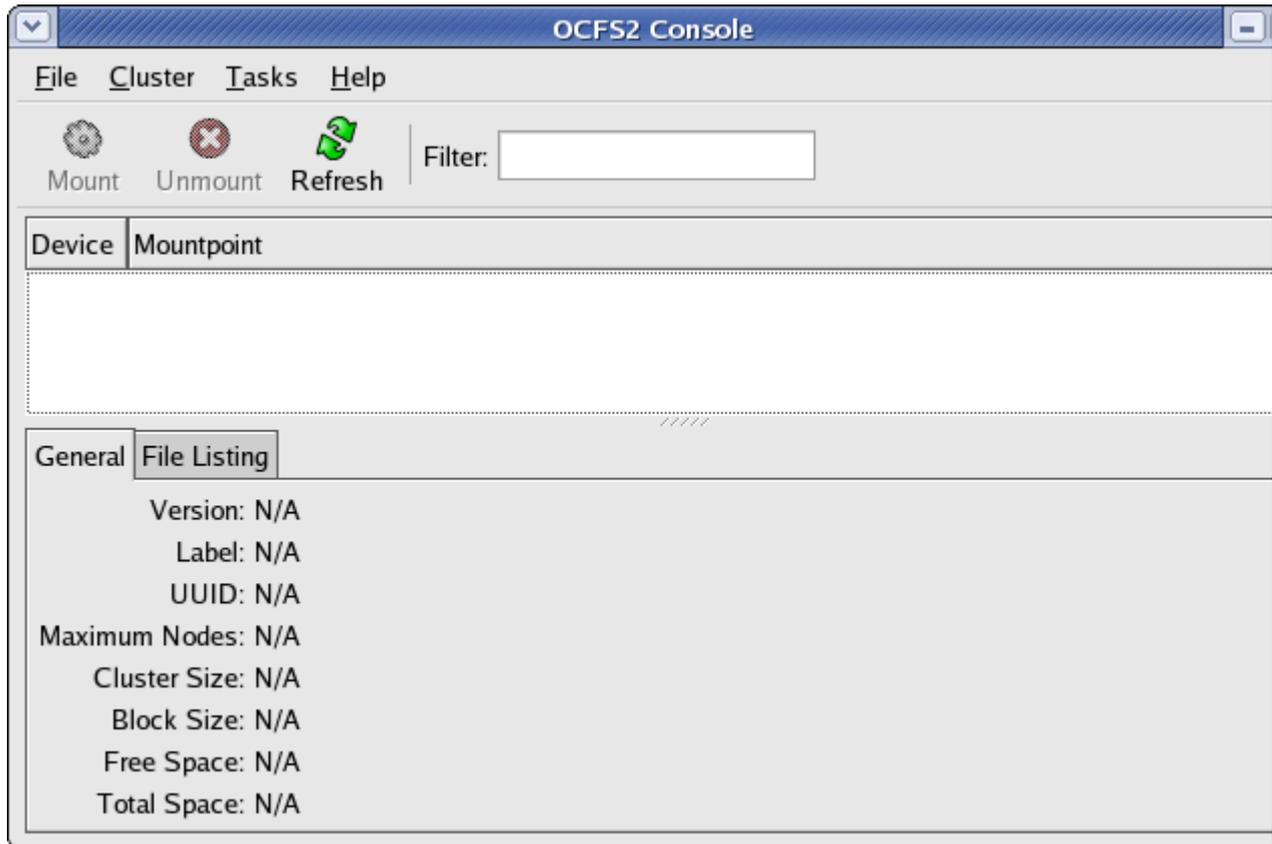


그림 15 ocfs2console GUI

ocfs2console GUI 툴을 이용하여 아래와 같이 작업해 줍니다:

1. [Cluster] -> [Configure Nodes...]를 선택합니다. OCFS2 Cluster Stack(그림 16)이 실행되고 "Node Configuration" 대화상자가 표시됩니다.
2. "Node Configuration" 대화상자에서 [Add] 버튼을 클릭합니다. "Add Node" 대화상자가 실행됩니다. "Add Node" 대화상자에서 클러스터 첫 번째 노드의 Host name과 IP address를 입력합니다. IP Port는 디폴트 값(7777)을 그대로 사용합니다. 본 예제에서는 첫 번째 노드에 linux1 / 192.168.1.100을 두 번째 노드에 linux2 / 192.168.1.101을 설정해 주었습니다. "Node Configuration" 대화상자의 [Apply]를 클릭합니다. 그림 17과 같이 모든 노드가 "Active"로 표시됩니다. "Node Configuration" 대화상자에서 [Close]를 클릭합니다." dialog.
3. 모든 값이 올바르게 입력되었음을 확인한 뒤, [File] -> [Quit] 메뉴를 사용하여 애플리케이션을 종료합니다. 클러스터의 모든 Oracle RAC 노드에 대해 같은 작업을 반복합니다.

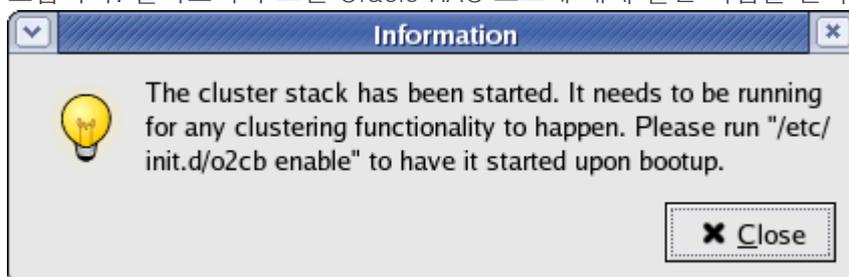


그림 16. OCFS2 Cluster Stack의 시작

아래 대화상자는 linux1 및 linux2 노드에 적용한 OCFS2 설정을 보여주고 있습니다

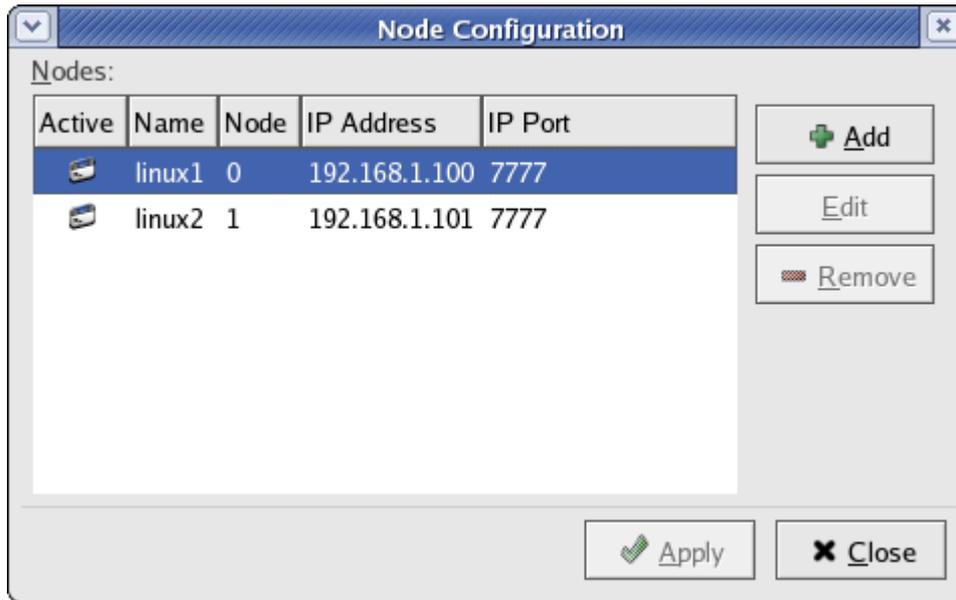


그림 17 OCFS2를 위한 노드 설정

ocfs2console을 종료한 후에, 아래와 유사한 /etc/ocfs2/cluster.conf 파일이 생성되었음을 확인할 수 있습니다. 이 작업은 클러스터의 모든 노드에 대해 수행되어야 하며, 모든 클러스터 노드의 OCFS2 구성 파일은 동일해야 합니다:

```
node:
  ip_port = 7777
  ip_address = 192.168.1.100
  number = 0
  name = linux1
  cluster = ocfs2
node:
  ip_port = 7777
  ip_address = 192.168.1.101
  number = 1
  name = linux2
  cluster = ocfs2
cluster:
  node_count = 2
  name = ocfs2
```

O2CB 클러스터 서비스

OCFS2를 이용하여 포맷, 파일 시스템 마운트 등의 작업을 시작하기 전에, 먼저 OCFS2 클러스터 스택(OC2B)를 실행해 주어야 합니다(위 구성 작업이 정상적으로 완료되면 자동 실행됩니다). OCFS2 클러스터 스택은 다음과 같은 서비스를 포함하고 있습니다:

NM: Node Manager – cluster.conf에 설정된 모든 노드의 상태를 추적

HB: Heart Beat 서비스 – 노드가 클러스터에 가입/탈퇴하는 경우 업/다운 통보를 전달

TCP: 노드 간의 커뮤니케이션을 관리

DLM: Distributed Lock Manager – 락, 락의 소유자 및 상태 정보를 추적

CONFIGFS: 사용자 공간(/config)에 마운트되는 구성 파일 시스템

DLMFS: 커널 스페이스 DLM을 위한 사용자 공간 인터페이스

위의 모든 클러스터 서비스는 *o2cb* 시스템 서비스(/etc/init.d/o2cb) 패키지에 포함되어 있습니다.

o2cb 시스템 서비스에서 유용한 커맨드와 옵션이 아래와 같습니다.

참고 : 아래 커맨드는 데모 용도로만 예시되고 있으며, 실제로 OCFS2를 설치/구성할 때에는 실행되지 않아야 합니다!

```
/etc/init.d/o2cb status
```

```
Module "configfs": Not loaded
Filesystem "configfs": Not mounted
Module "ocfs2_nodemanager": Not loaded
```

```
Module "ocfs2_dlm": Not loaded
Module "ocfs2_dlmfs": Not loaded
Filesystem "ocfs2_dlmfs": Not mounted
```

위 예제에서, 모든 서비스가 로드되지 않았음을 주목하시기 바랍니다. 이는 필자가 "status" 옵션을 실행하기 직전에 "unload" 작업을 수행했기 때문입니다. ocfs2console 유틸리티를 사용하여 OCFS를 설정한 직후에 o2cb 서비스의 상태를 확인하면 모든 서비스가 로드된 것으로 표시될 것입니다.

/etc/init.d/o2cb load

```
Loading module "configfs": OK
Mounting configfs filesystem at /config: OK
Loading module "ocfs2_nodemanager": OK
Loading module "ocfs2_dlm": OK
Loading module "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
```

전체 OCFS 모듈을 로드합니다.

/etc/init.d/o2cb online ocfs2

```
Starting cluster ocfs2: OK
```

생성된 ocfs2 클러스터를 온라인 상태로 전환합니다.

/etc/init.d/o2cb offline ocfs2

```
Unmounting ocfs2_dlmfs filesystem: OK
Unloading module "ocfs2_dlmfs": OK
Unmounting configfs filesystem: OK
Unloading module "configfs": OK
```

생성된 ocfs2 클러스터를 오프라인 상태로 전환합니다.

/etc/init.d/o2cb unload

```
Cleaning heartbeat on ocfs2: OK
Stopping cluster ocfs2: OK
```

모든 OCFS 모듈을 언로드합니다.

O2CB가 부팅 시 자동으로 시작되도록 설정

이제 OC2B 드라이브의 'on-boot' 속성을 설정하여 클러스터 스택이 부팅될 때마다 자동 실행되도록 해 주어야 합니다. 여기에서 설명되는 모든 작업은 클러스터의 모든 노드에서 실행되어야 합니다.

참고: OCFS2 1.2.1 이전 버전에는 'on-boot' 속성을 설정해 주었음에도 불구하고 부팅 시 드라이버가 로드되지 않는 버그가 존재합니다. 이 버그는 OCFS2 1.2.1에서 해결되었으며 따라서 본 예제 구성에서는 별도 작업이 불필요합니다. 만일 OCFS2 1.2.1 이전 버전을 사용하고 있다면 트러블슈팅 섹션을 참고하여 버그를 해결하시기 바랍니다.

아래와 같이 'on-boot' 속성을 설정합니다:

```
# /etc/init.d/o2cb offline ocfs2
```

```
# /etc/init.d/o2cb unload
```

```
# /etc/init.d/o2cb configure
```

```
Configuring the O2CB driver.
```

```
OC2B 드라이버의 on-boot 속성 설정을 위한 문답 작업이 시작됩니다..
```

```
아래 질문을 통해 부팅 과정에서 드라이버를 로드할 것인지 선택합니다
```

```
현재의 값은 대괄호( '[' ) 안에 표시되어 있습니다. 답을 입력하지 않고 <ENTER> 키를 누르면 현재 실행을 취소하고 중간에 빠져 나오려면 Ctrl-C를 누르면 됩니다.
```

```
Load O2CB driver on boot (y/n) [n]: y
```

```
Cluster to start on boot (Enter "none" to clear) [ocfs2]: ocfs2
```

```
Writing O2CB configuration: OK
```

```
Loading module "configfs": OK
```

```
Mounting configfs filesystem at /config: OK
Loading module "ocfs2_nodemanager": OK
Loading module "ocfs2_dlm": OK
Loading module "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Starting cluster ocfs2: OK
```

OCFS2 파일시스템 포맷

참고:본 예제의 다른 작업들과 달리, OCFS2 파일시스템의 생성 작업은 RAC 클러스터 노드 중 하나에 서만 실행해야 합니다. 이 섹션에서 설명되는 모든 커맨드는 linux1에서 실행되는 것으로 가정합니다. 이제 "iSCSI 볼륨에 파티션 생성" 섹션에서 생성한 iSCSI 볼륨 파티션을 포맷할 차례입니다.

앞에서 OCFS2(예: /dev/sdd1)에서 사용할 iSCSI 볼륨 파티션을 생성한 뒤, linux1을 여러 차례 리부팅하였을 것입니다. 이것은 매우 중요한 문제입니다. Openfiler의 iSCSI 타겟 네임과 linux1의 로컬 디바이스 네임의 매핑 설정이 달라져 있을 수 있기 때문입니다. linux1을 리부팅한 후 감지된 iSCSI 타겟 네임과 linux1의 로컬 SCSI 디바이스 네임이 서로 달라진 경우 "iSCSI 타겟의 존재 확인" 섹션에서 설명된 작업을 다시 반복하시기 바랍니다.

한 예로, 필자는 OCFS2가 사용할 iSCSI 볼륨의 primary partition을 생성하면서 iSCSI 타겟 네임 "iqn.2006-01.com.openfiler:rac1.crs"가 로컬 SCSI 디바이스 네임 /dev/sdd로 매핑되었음을 확인했습니다. 그런 다음, 앞 섹션의 SELinux를 비활성화하는 작업을 수행하면서 양쪽 노드를 모두 리부팅하였습니다. 그리고 나서 "iSCSI 타겟의 발견" 섹션에 설명된 작업을 수행하는 과정에서 "iqn.2006-01.com.openfiler:rac1.crs"이 로컬 SCSI 디바이스 네임 /dev/sde로 매핑이 변경되었음을 확인하였습니다. 이는 결국 /dev/sde1 파티션에 OCFS2가 생성됨을 의미합니다! 여러분이 확인하게 될 로컬 SCSI 디바이스 네임은 필자의 경우와 다를 가능성이 높다는 사실도 참고하시기 바랍니다.

O2CB 클러스터가 오프라인 상태라면 O2CB 클러스터를 시작합니다. 포맷 작업을 수행하려면 클러스터가 온라인 상태이어야 합니다. (볼륨이 클러스터의 다른 노드에 마운트되어 있지 않은지 확인하는 과정이 필요하기 때문입니다.)

앞 단계의 OCFS/Clusterware를 위한 마운트 포인트 생성 섹션에서 /u02/oradata/orcl 디렉토리를 생성해 두었습니다. 이 섹션에서는 *Cluster Manager*를 위해 사용될 파일시스템(/u02/oradata/orcl)을 생성하고 마운트하기 위한 커맨드를 소개하고 있습니다.

OCFS2 파일 시스템을 생성하고 마운트하기 위해 GUI 툴(ocfs2console)을 사용하거나 커맨드라인 툴(mkfs.ocfs2)을 사용할 수도 있습니다. ocfs2console 유틸리티의 경우, [Tasks] - [Format] 메뉴를 사용합니다.

mkfs.ocfs2 커맨드라인 툴을 이용하여 OCFS2 파일 시스템을 생성하는 방법은 아래 설명을 참고하시기 바랍니다.

파일 시스템을 생성하기 위해 오라클 실행파일 mkfs.ocfs2를 이용할 수 있습니다. 본 예제에서는 root 계정을 사용하여 linux1 노드에서만 아래 커맨드를 실행합니다. 이때 "crs" iSCSI 볼륨에 매핑된 로컬 SCSI 네임(/dev/sde1)을 사용합니다. 또 필자가 "oracrsfiles"라는 이름의 레이블(label)을 정의하였음을 참고하십시오. 이 레이블은 볼륨의 마운트/언마운트 작업에서 참조됩니다:

```
$ su -
# mkfs.ocfs2 -b 4K -C 32K -N 4 -L oracrsfiles /dev/sde1

mkfs.ocfs2 1.2.1
Filesystem label=oracrsfiles
Block size=4096 (bits=12)
Cluster size=32768 (bits=15)
Volume size=2145943552 (65489 clusters) (523912 blocks)
3 cluster groups (tail covers 977 clusters, rest cover 32256 clusters)
Journal size=67108864
Initial number of node slots: 4
Creating bitmaps: done
Initializing superblock: done
Writing system files: done
Writing superblock: done
Formatting Journals: done
Writing lost+found: done
mkfs.ocfs2 successful
```

OCFS2 파일시스템의 마운트

이제 생성된 파일시스템을 마운트할 차례입니다. 먼저 커맨드 라인을 이용한 마운트 방법을 설명하고,

부팅 시마다 자동 마운트 되도록 /etc/fstab 파일에 포함시키는 방법을 설명하겠습니다.

참고:파일 시스템의 마운트 작업은 루트 계정과 oracrsfiles OCFS2 레이블을 사용하여 Oracle RAC 클러스터의 모든 노드에서 실행되어야 합니다!

먼저, 커맨드 라인을 이용하여 OCFS2 파일시스템을 수동으로 마운트하는 방법이 아래와 같습니다. 이 작업은 반드시 root 계정으로 실행해야 합니다:

```
$ su -
# mount -t ocfs2 -o datavolume,nointr -L "oracrsfiles" /u02/oradata/orcl
```

마운트 작업이 성공적으로 완료되면, 별도의 메시지 없이 프롬프트가 다시 뜹니다. 하지만 파일시스템이 적절한 권한으로 올바르게 마운트 되었는지 점검해 주어야 할 것입니다.

먼저 mount 커맨드를 이용하여 새로운 파일시스템이 마운트 되었는지 여부를 확인합니다. 이 작업 역시 RAC 클러스터의 모든 노드에서 수행되어야 합니다:

```
# mount
/dev/mapper/VolGroup00-LogVol100 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
cartman:SHARE2 on /cartman type nfs (rw,addr=192.168.1.120)
configfs on /config type configfs (rw)
ocfs2_dlmfs on /dlm type ocfs2_dlmfs (rw)
/dev/sde1 on /u02/oradata/orcl type ocfs2 (rw,_netdev,datavolume,nointr,heartbeat=local)
```

참고: 새로운 파일 시스템의 마운트를 위해 datavolume 옵션을 사용하였음을 주목하십시오. 오라클 데이터베이스 사용자는 Voting Disk 파일, Cluster Registry (OCR), 데이터 파일, 리두 로그, 아카이브 로그, 컨트롤 파일을 포함하는 모든 볼륨을 마운트할 때, 반드시 datavolume 마운트 옵션을 사용함으로써 오라클 프로세스가 o_direct 플래그를 사용하여 프로세스를 오픈하도록 설정해야 합니다. nointr 옵션은 I/O가 시그널에 의해 중단(interrupt)되지 않음을 보장합니다.

Oracle Home(이 가이드에는 사용되지 않습니다)을 비롯한 다른 유형의 볼륨에는 이 마운트 옵션을 사용하면 안됩니다.

그렇다면 마운트에 왜 이렇게 오랜 시간이 걸리는 것일까요? 볼륨은 약 5 초 간의 지연 후 마운트 됩니다. 지연 시간은 heartbeat 쓰레드의 안정화를 위해 필요합니다. 오라클은 향후 릴리즈에 global heartbeat 지원 기능을 추가함으로써 즉각적인 마운트가 가능하도록 할 예정입니다.

OCFS가 부팅 시 자동 마운트되도록 설정

지금까지 작업한 내용을 다시 되짚어 보겠습니다. 먼저 *Cluster Manager* 파일을 저장하기 위한 OCFS2 파일시스템을 다운로드하고 설치하였습니다. 설치를 완료한 후에는 OCFS2 모듈을 커널에 로드하고 클러스터 파일시스템을 포맷하였습니다. 마지막으로 "oracrsfiles"라는 OCFS2 레이블을 사용하여 새로 생성된 파일시스템을 마운트하였습니다. 이 섹션에서는 머신이 부팅될 때마다 OCFS2 파일시스템이 마운트 되도록 설정하는 방법을 설명합니다.

먼저 Oracle RAC 클러스터의 모든 노드의 /etc/fstab 파일에 아래 라인을 추가합니다:

```
LABEL=oracrsfiles /u02/oradata/orcl ocfs2 _netdev,datavolume,nointr 0 0
```

마운트 과정에서 _netdev 옵션을 사용하였음을 주목하시기 바랍니다. _netdev 옵션은 OCFS2 볼륨에 필수적으로 적용됩니다. 이 옵션은 모든 네트워크 디바이스가 활성화된 이후에 볼륨이 마운트 되고, 네트워크가 섣다운되기 전에 볼륨을 디스마운트 하도록 합니다.

이제, 부팅 과정에서 ocfs2.ko 커널 모듈이 로드되고 파일시스템이 정상적으로 마운트 되는지 확인할 차례입니다.

이 가이드의 설명을 그대로 따라 했다면, 커널 모듈을 로드하고 OCFS2 파일시스템을 마운트 하는데 필요한 설정이 이미 완료되어 있을 것입니다. root 계정으로 RAC 클러스터의 모든 노드에서 아래 명령

을 실행하여 설정 내용을 확인합니다:

```
$ su -
# chkconfig --list o2cb
o2cb 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

굵은 글씨체로 표시된 플래그는 반드시 on으로 설정되어 있어야 합니다.

OCFS2 파일시스템의 permission 점검

다음으로, ls 커맨드를 사용하여 ownership을 점검합니다 owner는 oracle, group은 dba, permission은 0775로 설정되어 있어야 합니다.

먼저 permission을 확인해 봅시다:

```
# ls -ld /u02/oradata/orcl
drwxr-xr-x 3 root root 4096 Sep 29 12:11 /u02/oradata/orcl
```

위 결과에서, oracle 사용자 계정(dba 그룹)이 이 디렉토리에 쓰기 권한을 보유하고 있지 않은 것을 확인할 수 있습니다 아래와 같이 수정해 줍니다:

```
# chown oracle.dba /u02/oradata/orcl
# chmod 775 /u02/oradata/orcl
```

이제 다시 클러스터의 각 노드 별로 permission이 올바르게 설정되어 있는지 확인합니다:

```
# ls -ld /u02/oradata/orcl
drwxrwxr-x 3 oracle dba 4096 Sep 29 12:11 /u02/oradata/orcl
```

O2CB Heartbeat Threshold의 조정

이제 Oracle Clusterware 공유 파일이 사용하는 OCFS2를 설정하는 과정에서 가장 중요한 작업의 하나를 수행하도록 하겠습니다. 본 문서의 이전 버전(iSCSI가 아닌 FireWire를 공유 스토리지로 사용한 버전)을 작성하면서, 필자는 OCFS2의 설치/구성, 새로운 볼륨의 포매팅, Oracle Clusterware의 설치와 같은 작업을 성공적으로 수행할 수 있었습니다. Oracle Clusterware를 설치하고 FireWire를 통해 공유 드라이브 연결하는 것은 성공적이었지만, Clusterware 소프트웨어가 양쪽 노드에서 실행된 지 약 15 분 경과 후 많은 락업(lock-up)과 행(hang) 현상이 발생하기 시작했습니다. 행(hang) 현상이 발생하는 노드(linux1, linux2)는 경우에 따라 달랐습니다. 또 I/O 부하가 심하지 않은(또는 전혀 없는) 상황에서도 크래시 또는 행 현상이 발생했습니다.

OCFS2의 트레이스 파일을 점검한 뒤, Voting Disk의 속도가 너무 늦어 O2CB heartbeat 임계치보다 낮은 성능을 보이며, 이로 인해 Oracle Clusterware 소프트웨어가 크래시 현상을 발생시키는 것으로 확인되었습니다.

필자는 이 문제를 해결하기 위해 OC2B heartbeat threshold를 7(디폴트)에서 61로 변경하였습니다. 환경에 따라서는 더 높은 수치가 필요할 수도 있을 것입니다. 이 임계치 설정은 노드가 자신을 "차단(fence)"하는데 필요한 시간을 계산하기 위해 사용됩니다.

공유 스토리지로 FireWire, iSCSI 중 어느 것을 사용하든, 여기서 예시되는 구성이 교육 및 테스트를 위한 로우-엔드 셋업이라는 사실을 잊지 말기로 합니다. 이 구성은 어떤 면에서도 하이-엔드 환경으로 보이 어려우며, 어디에서든 예기치 못한 타임아웃 현상을 경험할 수 있습니다. 따라서 이 섹션에 설명된 내용을 따라 O2CB heartbeat threshold를 증가시키고 공유 드라이브의 느린 액세스 문제를 해결하는 것이 강력하게 권장됩니다. 다만, 필자가 O2CB heartbeat threshold를 증가시키지 않고도 (iSCSI를 이용한) 본 예제 구성을 성공적으로 실행할 수 있었다는 점을 언급해 둘 필요가 있어 보입니다. 하지만 필자가 과거에 경험한 것과 동일한 현상을 겪을지도 모르는 사용자들을 위해 이 섹션을 수정하지 않고 남겨 두기로 했습니다.

먼저 현재 설정된 OC2B heartbeat threshold를 확인해 봅시다. 이를 위해 /proc 파일시스템을 조회합니다:

```
# cat /proc/fs/ocfs2_nodemanager/hb_dead_threshold
7
```

7이라는 값이 조회되었습니다. 그런데 이 숫자가 무엇을 의미하는 것일까요? 실제로 "fence time"을

결정하기 위해 사용되는 공식은 아래와 같습니다 (단위는 초입니다):

$$[\text{fence time in seconds}] = (\text{O2CB_HEARTBEAT_THRESHOLD} - 1) * 2$$

O2CB heartbeat threshold가 7로 설정되어 있는 경우 fence time은 아래와 같이 계산됩니다:

$$(7 - 1) * 2 = 12 \text{ seconds}$$

사용 중인 외장형 디스크와 네트워크의 느린 속도를 감안하여 훨씬 더 높은 임계치(120초)를 적용해 주어야 할 수도 있습니다. fence time을 120초로 설정하기 위해서는 O2CB_HEARTBEAT_THRESHOLD를 61로 정의해 주어야 합니다:

$$(61 - 1) * 2 = 120 \text{ seconds}$$

이제 O2CB heartbeat threshold를 7에서 61로 증가시키는 작업을 수행할 차례입니다. 이 작업은 클러스터의 모든 노드에서 실행되어야 합니다. 먼저, /etc/sysconfig/o2cb 파일을 열고 O2CB_HEARTBEAT_THRESHOLD를 61로 수정합니다:

```
# O2CB_ENABLED: 'true' means to load the driver on boot.
O2CB_ENABLED=true
# O2CB_BOOTCLUSTER: If not empty, the name of a cluster to start.
O2CB_BOOTCLUSTER=ocfs2
# O2CB_HEARTBEAT_THRESHOLD: Iterations before a node is considered dead.
O2CB_HEARTBEAT_THRESHOLD=61
```

/etc/sysconfig/o2cb 파일의 수정작업이 끝났다면, 다음에는 o2cb 설정을 변경해 줍니다 이 작업 역시 클러스터의 모든 노드에서 수행되어야 합니다.

```
# umount /u02/oradata/orcl/
# /etc/init.d/o2cb unload
# /etc/init.d/o2cb configure
Load O2CB driver on boot (y/n) [y]: y
Cluster to start on boot (Enter "none" to clear) [ocfs2]: ocfs2
Writing O2CB configuration: OK
Loading module "configfs": OK
Mounting configfs filesystem at /config: OK
Loading module "ocfs2_nodemanager": OK
Loading module "ocfs2_dlm": OK
Loading module "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Starting cluster ocfs2: OK
```

이제 oc2b 클러스터 스택의 설정을 확인하면 변경된 값을 확인할 수 있습니다:

```
# cat /proc/fs/ocfs2_nodemanager/hb_dead_threshold
61
```

61의 O2CB heartbeat threshold 값은 구성에 따라 지원되지 않을 수도 있습니다. 또 경우에 따라서는 OCFS2 때문에 커널에 패닉 현상이 발생하는 것을 방지하기 위해 O2CB heartbeat threshold 값을 601로 증가시켜야 할 수 있습니다.

모든 노드를 리부트

다음 섹션으로 진행하기 전에, RAC 클러스터의 모든 노드를 리부팅합니다. 서버가 다시 온라인 상태가 되면, 클러스터 스택 서비스가 로드되고 OCFS2 파일시스템이 마운트되었는지 확인합니다:

```
# mount
/dev/mapper/VolGroup00-LogVol100 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
```

```

none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
configfs on /config type configfs (rw)
ocfs2_dlmfs on /dlm type ocfs2_dlmfs (rw)
cartman:SHARE2 on /cartman type nfs (rw,addr=192.168.1.120)
/dev/sdd1 on /u02/oradata/orcl type ocfs2 (rw,_netdev,datavolume,nointr,heartbeat=local)

```

O2CB heartbeat threshold를 수정했다면 올바르게 설정되었는지 확인해 줍니다:

```

# cat /proc/fs/ocfs2_nodemanager/hb_dead_threshold
61

```

OCFS2 버전의 확인

OCFS2의 버전을 확인하는 방법은 아래와 같습니다:

```

# cat /proc/fs/ocfs2/version
OCFS2 1.2.3 Thu Aug 10 18:16:03 PDT 2006 (build 6b798aaadf626d3b137c3952809b2f38)

```

17. Automatic Storage Management (ASMLib 2.0)의 설치 및 설정

여기에서 설명된 작업의 대부분은 클러스터의 모든 노드에서 실행되어야 합니다! 단, ASM 디스크의 생성 작업은 클러스터 노드 중 단 하나에서만 실행되어야 합니다.

이 섹션에서는 물리적 오라클 데이터베이스 파일(데이터, 온라인 리두 로그, 컨트롤 파일, 아카이브 리두 로그 등)과 Flash Recovery Area를 위한 파일시스템/볼륨 관리자에 ASM(Automatic Storage Management)을 사용하도록 설정하는 방법을 설명합니다

Oracle Database 10g에서 처음 소개된 기능인 ASM은 개별 파일 및 드라이브의 관리에 수반되는 DBA의 업무 부담을 최소화하기 위한 목적에서 개발되었습니다. ASM은 오라클 커널에 내장된 형태로 구현되었으며, 단일 인스턴스 또는 클러스터 환경에서 수천 개의 디스크를 24x7 기준으로 관리할 수 있는 환경을 제공합니다. 오라클이 사용하는 모든 파일과 디렉토리는 *디스크 그룹(disk group)*에 저장됩니다. ASM은 전체 디스크 드라이브에 대해 병렬적으로 로드 밸런싱을 수행함으로써 성능 병목을 해소하고 성능을 극대화하며, 수시로 변화하는 데이터 사용 패턴에 대응할 수 있게 합니다.

Linux 환경에서 ASM을 설정하는 방법은 크게 두 가지가 있습니다:

ASM with ASMLib I/O:ASMLib 호출을 사용하여 ASMLib이 관리하는 *로우 블록 디바이스(raw block device)*에 오라클 데이터베이스 파일을 생성하는 방법입니다 ASMLib은 블록 디바이스(block device)와 연동하므로, 이 방법을 사용할 때 반드시 로우 디바이스(raw device)를 사용할 필요는 없습니다.

ASM with Standard Linux I/O:표준 Linux I/O 시스템 호출을 사용하여 ASM이 관리하는 *로우 캐릭터 디바이스(raw character device)*에 오라클 데이터베이스 파일을 생성하는 방법입니다. 이 경우, ASM이 사용할 모든 디스크 파티션에 로우 디바이스를 생성해야 합니다.

여기에서는 ASM with ASMLib I/O 방식을 사용하기로 합니다. 오라클은 *ASMLib은 표준 UNIX I/O API의 한계에 제약 받지 않고 Linux 환경에서 ASM I/O를 제공한다고* 설명하고 있습니다(Metalink Note 275315.1). 필자는 ASMLib이 제공하는 성능적인 이점에 대해 몇 가지 테스트를 수행할 계획이지만, 성능 및 테스트에 대한 상세한 정보는 본 문서의 주제에서 벗어나므로 다루지 않는 것으로 하겠습니다.

먼저 예제 구성에서 사용하는 Linux 커널 버전에 맞는 ASM 드라이버(ASMLib Release 2.0)를 다운로드하고, ASM 2.0 드라이버를 설치하고 구성하는 방법을 설명한 후, 마지막으로 ASM 디스크를 생성하는 방법을 예시하도록 하겠습니다.

Oracle ASMLib 2.0에 관련한 자세한 정보는 <http://www.oracle.com/technology/tech/linux/asmlib/>에서 확인하시기 바랍니다.

ASMLib 2.0 패키지 다운로드

먼저 ASMLib 2.0 라이브러리와 드라이버를 OTN에서 다운로드합니다. 이 문서가 작성되는 시점의 ASMLib 드라이버 최신 버전은 2.0.3-1입니다. OCFS2의 경우와 마찬가지로, ASMLib 2.0 패키지 역시 Linux 커널 버전과 서버 프로세서 수에 따라 그 종류가 달라집니다. 여기에서는 싱글-프로세서 머

신과 커널 버전 2.6.9-42.EL #1에 대응되는 패키지를 선택하기로 합니다: 2.0.3-1.

```
# uname -a
Linux linux1 2.6.9-42.EL #1 Sat Aug 12 09:17:58 CDT 2006 i686 i686 i386 GNU/Linux
```

```
Oracle ASMLib for Red Hat Enterprise Linux 4 AS 다운로드
oracleasm-2.6.9-42.EL-2.0.3-1.i686.rpm - (for single processor)
-OR-
oracleasm-2.6.9-42.ELsmp-2.0.3-1.i686.rpm - (for multiple processors)
-OR-
oracleasm-2.6.9-42.ELhugemem-2.0.3-1.i686.rpm - (for hugemem)
또 아래 ASMLib 툴을 추가로 다운로드합니다:
oracleasm-lib-2.0.2-1.i386.rpm - (Userspace library)
oracleasm-support-2.0.3-1.i386.rpm - (Driver support files)
```

ASMLib 2.0 패키지의 설치

이 작업은 root 계정 권한으로 클러스터의 모든 노드에서 실행해 주어야 합니다:

```
$ su -
# rpm -Uvh oracleasm-2.6.9-42.EL-2.0.3-1.i686.rpm W
      oracleasm-lib-2.0.2-1.i386.rpm W
      oracleasm-support-2.0.3-1.i386.rpm
Preparing...##### [100%]
 1:oracleasm-support##### [ 33%]
 2:oracleasm-2.6.9-42.EL##### [ 67%]
 3:oracleasm-lib##### [100%]
```

ASMLib 2.0 패키지의 설정 및 로드

Linux 운영체제를 위한 ASMLib 패키지의 다운로드 및 설치를 완료했다면, 이제 ASM 커널 모듈을 설정하고 로드할 차례입니다. root 계정을 사용하여 모든 Oracle RAC 노드에 대해 아래 작업을 실행합니다:

```
$ su -
# /etc/init.d/oracleasm configure
Configuring the Oracle ASM library driver.
```

This will configure the on-boot properties of the Oracle ASM library driver. The following questions will determine whether the driver is loaded on boot and what permissions it will have. The current values will be shown in brackets ('[]'). Hitting <ENTER> without typing an answer will keep that current value. Ctrl-C will abort.

```
Default user to own the driver interface []: oracle
Default group to own the driver interface []: dba
Start Oracle ASM library driver on boot (y/n) [n]: y
Fix permissions of Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: [ OK ]
Creating /dev/oracleasm mount point: [ OK ]
Loading module "oracleasm": [ OK ]
Mounting ASMLib driver filesystem: [ OK ]
Scanning system for ASM disks: [ OK ]
```

오라클 환경을 위한 ASM 디스크 설정

ASM 디스크 생성 작업은 root 계정 권한으로 RAC 클러스터 노드 중 하나에서만 실행해야 합니다: 예제 구성에서는 이 작업을 linux1에서 실행하는 것으로 가정합니다. 다른 Oracle RAC 노드에서는 *scandisk* 명령을 실행하여 새로운 볼륨을 인식하도록 합니다. 이 작업이 완료되면, 모든 노드에서 *oracleasm listdisks* 커맨드를 실행하여 ASM 디스크의 사용 가능 여부를 확인합니다.

"iSCSI 볼륨에 파티션 생성하기" 섹션에서 ASM에 의해 사용될 4 개의 iSCSI 볼륨을 생성하였습니다. ASM은 온라인 리두 로그, 데이터베이스 파일, 컨트롤 파일, 아카이브 리두 로그 파일, 플래시 복구 영역 등의 오라클 데이터베이스 파일을 저장하기 위해 사용됩니다.

ASM에서 사용할 iSCSI 볼륨의 파티션(예: /dev/sda, /dev/sdb, /dev/sdc, /dev/sde)을 생성한 뒤 Linux1 서버를 여러 차례 리부팅하였을 것입니다. 이것은 매우 중요한 문제입니다. Openfiler의 iSCSI 타겟 네임과 Linux1의 로컬 디바이스 네임의 매핑 설정이 달라져 있을 수 있기 때문입니다. Linux1을 리부팅한 후 감지된 4개 ASM 볼륨의 iSCSI 타겟 네임과 Linux1의 로컬 SCSI 디바이스 네임이 서로 달린 경우 "iSCSI 타겟의 존재 확인" 섹션에서 설명된 작업을 다시 반복하시기 바랍니다. 한 예로, 필자는 OCFS2 설정 작업 후 Oracle RAC 노드들을 리부팅하였습니다. 리부팅 후, Linux1의 ASM iSCSI 타겟 네임 매핑은 아래와 같이 달라져 있었습니다:

iSCSI Target Name to local SCSI Device Name - (ASM)		
iSCSI Target Name	Host / SCSI ID	SCSI Device Name
iqn.2006-01.com.openfiler:rac1.asm4	0	/dev/sda
iqn.2006-01.com.openfiler:rac1.asm3	1	/dev/sdb
iqn.2006-01.com.openfiler:rac1.asm2	2	/dev/sde
iqn.2006-01.com.openfiler:rac1.asm1	3	/dev/sdc

ASM 디스크를 생성할 때 위의 iSCSI 타겟 네임을 사용하여 대응되는 로컬 SCSI 디바이스 네임을 확인해야 합니다.

참고: 동일한 하드웨어(동일한 공유 드라이브)를 사용하여 본 문서의 작업을 반복하면, ASM 디스크를 생성하는 과정에서 에러가 발생할 수 있습니다. 에러가 발생하면, 아래 명령을 실행하여 전체 ASM 디스크의 목록을 확인합니다:

```
# /etc/init.d/oracleasm listdisks
VOL1
VOL2
VOL3
VOL4
```

위에서 이미 4개의 ASM 볼륨이 정의되어 있음을 확인할 수 있습니다. 이미 이전 실행 과정에서 볼륨이 생성되었다면, 아래 명령을 사용하여 기존 볼륨을 제거합니다. 그런 다음 oracleasm createdisk 명령을 사용하여 새로운 볼륨을 다시 생성하도록 합니다.

```
# /etc/init.d/oracleasm deletedisk VOL1
Removing ASM disk "VOL1" [ OK ]
# /etc/init.d/oracleasm deletedisk VOL2
Removing ASM disk "VOL2" [ OK ]
# /etc/init.d/oracleasm deletedisk VOL3
Removing ASM disk "VOL3" [ OK ]
# /etc/init.d/oracleasm deletedisk VOL4
Removing ASM disk "VOL4" [ OK ]
```

iSCSI 타겟 네임과 로컬 SCSI 디바이스 네임의 매핑 정보를 이용하여 ASM 디스크를 생성하려면 아래와 같이 실행합니다:

```
$ su -
# /etc/init.d/oracleasm createdisk VOL1 /dev/sdc1
Marking disk "/dev/sdc1" as an ASM disk [ OK ]

# /etc/init.d/oracleasm createdisk VOL2 /dev/sde1
Marking disk "/dev/sde1" as an ASM disk [ OK ]

# /etc/init.d/oracleasm createdisk VOL3 /dev/sdb1
Marking disk "/dev/sdb1" as an ASM disk [ OK ]

# /etc/init.d/oracleasm createdisk VOL4 /dev/sda1
Marking disk "/dev/sda1" as an ASM disk [ OK ]
```

클러스터의 다른 모든 >노드에서, scandisk를 수행하여 새로운 볼륨을 인식하도록 합니다:

```
# /etc/init.d/oracleasm scandisks
Scanning system for ASM disks [ OK ]
```

이제 root 계정으로 RAC 클러스터의 모든 노드에 대해 아래 명령을 수행하여 ASM 디스크가 성공적으로 생성되었음을 확인합니다.

```
# /etc/init.d/oracleasm listdisks
VOL1
VOL2
VOL3
VOL4
```

18. Oracle RAC 10g 소프트웨어 다운로드

아래 다운로드 작업은 클러스터 노드 중 하나에서만 실행되어야 합니다!

다음은 Oracle Clusterware Release 2 (10.2.0.1.0), Oracle Database 10g Release 2 (10.2.0.1.0), Oracle Database 10g Companion CD Release 2 (10.2.0.1.0) for Linux x86 소프트웨어를 설치할 차례입니다 먼저 OTN에서 필요한 오라클 소프트웨어 패키지를 다운로드하여 압축을 풉니다.

이 작업은 클러스터 노드 중 하나에서만 실행되어야 합니다(여기에서는 모든 설치 작업을 linux1 서버에서 실행하기로 합니다). Oracle Installer는 필요한 소프트웨어 패키지를 섹션 14(원격 액세스를 위한 RAC 노드의 설정)에서 설정한 다른 RAC 노드에 자동으로 복사합니다.

oracle 사용자 계정으로 오라클 설치 작업을 시작할 서버(linux1)에 로그인합니다. 오라클 소프트웨어를 다운로드한 뒤 /u01/app/oracle/orainstall 디렉토리에 저장합니다.

소프트웨어 다운로드 및 압축 풀기

먼저, Oracle Clusterware Release 2 (10.2.0.1.0), Oracle Database 10g Release 2 (10.2.0.1.0), Oracle Database 10g Companion CD Release 2 (10.2.0.1.0) for Linux x86 소프트웨어를 download 합니다. (세 가지 소프트웨어를 모두 같은 페이지에서 다운로드할 수 있습니다.)

oracle 사용자 계정으로 두 패키지를 임시 디렉토리에 다운로드합니다. 예제 구성에서는 /u01/app/oracle/orainstall 디렉토리를 사용하기로 합니다.

아래와 같이 실행하여 Oracle Clusterware 패키지의 압축을 풉니다:

```
# su - oracle
$ cd ~oracle/orainstall
$ unzip 10201_clusterware_linux32.zip
```

다음으로 Oracle Database 소프트웨어의 압축을 풉니다:

```
$ cd ~oracle/orainstall
$ unzip 10201_database_linux32.zip
```

마지막으로 Oracle Companion CD의 압축을 풉니다:

```
$ cd ~oracle/orainstall
$ unzip 10201_companion_linux32.zip
```

19. Oracle10g Release 2 설치를 위한 사전 준비 작업

클러스터의 모든 Oracle RAC 노드에서 아래 작업을 수행합니다!

Linux O/S(CentOS Enterprise Linux 또는 RHEL 4)를 설치하면서 오라클 환경에 필요한 모든 RPM이 설치되었는지 확인해야 합니다. 앞에서 설명된 내용을 모두 따라 했다면, 모든 패키지가 올바르게 설치되어 있을 것입니다. 하지만 다른 설치 옵션(예: "Advanced Server")을 사용했다면 설치 과정에서

일부 패키지가 누락되었을 수도 있습니다. 필요한 모든 RPM은 Linux CD/ISO에서 찾을 수 있습니다.

오라클 설치를 위한 준비 작업의 일환으로 Cluster Verification Utility(CVU)를 실행합니다. CVU는 Oracle Clusterware 설치 미디어에 포함된 커맨드라인 유틸리티입니다. CVU는 Oracle RAC 노드가 Oracle Clusterware 및 Oracle Real Application Cluster 설치에 필요한 설정 요건을 만족하는지 점검하기 위한 도구로 활용됩니다. CVU는 오라클 설치 작업을 진행하는 노드(Linux1)에서만 실행하면 됩니다.

필요한 RPM의 확인

아래 패키지가 설치되어 있어야 합니다(Linux 배포본에 따라 버전 넘버가 조금씩 다를 수 있습니다).

```
binutils-2.15.92.0.2-21
compat-db-4.1.25-9
compat-gcc-32-3.2.3-47.3
compat-gcc-32-c++-3.2.3-47.3
compat-libstdc++-33-3.2.3-47.3
compat-libgcc-296-2.96-132.7.2
control-center-2.8.0-12.rhel4.5
cpp-3.4.6-3
gcc-3.4.6-3
gcc-c++-3.4.6-3
glibc-2.3.4-2.25
glibc-common-2.3.4-2.25
glibc-devel-2.3.4-2.25
glibc-headers-2.3.4-2.25
glibc-kernheaders-2.4-9.1.98.EL
gnome-libs-1.4.1.2.90-44.1
libaio-0.3.105-2
libstdc++-3.4.6-3
libstdc++-devel-3.4.6-3
make-3.80-6.EL4
openmotif-2.2.3-10.RHEL4.5
openmotif21-2.1.30-11.RHEL4.6
pdksh-5.2.14-30.3
setarch-1.6-1
sysstat-5.0.5-11.rhel4
xscreensaver-4.18-5.rhel4.11
```

openmotif RPM 패키지는 오라클 데모를 설치하는 경우에만 필요합니다. 본 문서에서는 오라클 데모를 설치하지 않습니다.

패키지 정보를 확인하려면, 아래와 같이 rpm -q <PackageName> [, <PackageName>] 커맨드를 사용합니다:

```
# rpm -q gcc glibc-devel
gcc-3.4.6-3
glibc-devel-2.3.4-2.25
```

위 목록의 패키지 중 하나를 별도로 설치해야 하는 경우에는 rpm -Uvh <PackageName.rpm>. 커맨드를 사용합니다 예를 들어 GCC gcc-3.4.6-3 패키지를 설치하는 방법이 아래와 같습니다:

```
# rpm -Uvh gcc-3.4.6-3.i386.rpm
```

Cluster Verification Utility 사용을 위한 사전 요구사항

JDK 1.4.2

CVU를 실행하려면 먼저 JDK 1.4.2가 시스템에 설치되어 있어야 합니다. JDK 1.4.2가 설치되어 있지 않은 상태에서 CVU를 실행하면 아래와 같은 에러 메시지가 표시됩니다:

```
ERROR. Either CV_JDKHOME environment variable should be set
or /stagepath/cluvfy/jrepack.zip should exist.
```

JDK 1.4.2가 설치되어 있지 않다면, Sun Web 사이트에서 다운로드하고 Sun의 매뉴얼을 참고하여 설치합니다. JDK 1.4.2는 다음 웹 사이트에서 다운로드할 수 있습니다:

<http://www.sun.com/java>.

JDK 1.4.2를 설치한 후에는 사용자 환경 변수 CV_JDKHOME를 정의하여 JDK의 경로를 설정해 주어야 합니다. 예를 들어, JDK 1.4.2가 /usr/local/j2re1.4.2_08에 설치되어 있다면, CVU 실행에 사용할 사용자로 로그인한 다음, 아래 명령을 실행합니다:

```
CV_JDKHOME=/usr/local/j2re1.4.2_08
export CV_JDKHOME
```

cvuqdisk RPM 설치 (RHEL 환경에만 해당)

아래 내용은 Red Hat Linux 환경에서 CVU를 실행하는 경우에만 적용됩니다. Red Hat Linux를 사용하는 경우 Red Hat 운영체제 패키지 중 하나인 cvuqdisk를 Oracle RAC 클러스터의 모든 노드에 설치해야 합니다. cvuqdisk가 설치되어 있지 않으면, CVU는 공유 디스크를 발견할 수 없으며 CVU 실행 시 "Package cvuqdisk not installed" 에러가 발생합니다.

cvuqdisk RPM은 Oracle Clusterware 설치 미디어의 rpm 디렉토리에 위치하고 있습니다. 본 구성 예제에서는 Oracle Clusterware 미디어를 linux1 노드의 /u01/app/oracle/orainstall/clusterware 디렉토리에 복사하였습니다. cvuqdisk RPM을 설치하기 전에 CVUQDISK_GRP 환경 변수를 cvuqdisk 유틸리티를 소유한 그룹으로 지정해야 함을 참고하십시오. 디폴트 그룹은 oinstall이며, 본 예제 구성에서 oracle UNIX 사용자 계정은 이 그룹에 포함되지 않습니다. 여기에서는 dba 그룹이 사용되고 있으므로, CVUQDISK_GRP=dba로 설정한 다음 cvuqdisk RPM 설치를 진행해야 합니다.

cvuqdisk RPM을 linux1에서 linux2로 복사하고, 아래와 같이 양쪽 Oracle RAC 노드에 RPM을 설치합니다:

```
$ su -
# cd /u01/app/oracle/orainstall/clusterware/rpm
# CVUQDISK_GRP=dba; export CVUQDISK_GRP

# rpm -iv cvuqdisk-1.0.1-1.rpm
Preparing packages for installation...
cvuqdisk-1.0.1-1

# ls -l /usr/sbin/cvuqdisk
-rwsr-x--- 1 root dba 4168 Jun  2 2005 /usr/sbin/cvuqdisk
```

원격 액세스 / User Equivalence의 검증

CVU는 오라클 설치 작업을 진행하는 노드(linux1)에서만 실행하면 됩니다. CVU를 실행하기 전에 oracle 사용자 계정으로 로그인하여 클러스터의 모든 노드에 대해 원격 액세스/user equivalence가 설정되었는지 검증합니다. SSH(secure shell) 을 사용하는 경우에는 CVU를 실행하기 전에 터미널 세션에 "user equivalence"를 활성화해야 합니다. 현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행하고 생성된 각 키를 위한 pass phrase가 입력될 수 있도록 합니다.

```
# su - oracle
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
```

RSH(Remote Shell)을 사용하는 경우에는 user equivalence가 /etc/hosts.equiv 파일에 정의되며, 새로운 터미널 셸 세션에 대해 기본적으로 활성화됩니다.

CVU를 이용한 CRS 사전 설치 요구사항 점검

CVU 사용을 위한 요구사항이 모두 만족되었다면, linux1 노드에서 "oracle" UNIX 계정을 이용하여 아래 커맨드를 실행하여 Oracle Clusterware(CRS)의 설치 전 요구사항을 점검합니다.:

```
$ cd /u01/app/oracle/orainstall/clusterware/cluvfy
$ ./runcluvfy.sh stage -pre crsinst -n linux1,linux2 -verbose
```

CVU 리포트를 검토합니다. 리포트에서 보고한 에러 중 몇 가지는 무시해도 좋은 것들입니다. 먼저 "oracle"이 "oinstall" 그룹에 포함되어 있지 않다는 에러가 있습니다.

본 문서에서 "oracle" 사용자 계정은 "dba" 그룹에만 할당되며, 따라서 이 에러는 무시해도 무방합니다.

두 번째로 VIP를 위한 인터페이스를 검색하는데 관련한 에러가 있습니다. 이 에러는 버그이며, 자세한 정보는 Metalink Note 338924.1에서 확인할 수 있습니다:

```
Suitable interfaces for the private interconnect on subnet "192.168.2.0":
linux2 eth1:192.168.2.101
linux1 eth1:192.168.2.100
```

```
ERROR:
Could not find a suitable set of interfaces for VIPs.
```

```
Result: Node connectivity check failed.
```

따라서 이 에러는 무시해도 무방합니다.

마지막으로, RHEL4 Update \$에 존재하지 않는 RPM 패키지 버전에 관련한 에러들이 있습니다. 그 예가 아래와 같습니다:

```
compat-gcc-7.3-2.96.128
compat-gcc-c++-7.3-2.96.128
compat-libstdc++-7.3-2.96.128
compat-libstdc++-devel-7.3-2.96.128
```

CVU 리포트 상에는 이 패키지들이 존재하지 않는다고 보고되지만, compat-* 패키지의 올바른 버전이 이미 클러스터의 모든 Oracle RAC 노드에 설치되어 있습니다. RHEL4 Update 4의 경우 각 패키지의 버전은 아래와 같습니다:

```
compat-gcc-32-3.2.3-47.3
compat-gcc-32-c++-3.2.3-47.3
compat-libstdc++-33-3.2.3-47.3
```

CVU를 이용한 하드웨어 및 운영체제 셋업의 점검

다음으로 CVU를 이용하여 하드웨어 및 운영체제 셋업을 검증해 봅시다. linux1 노드에서 "oracle" UNIX 계정을 사용하여 아래와 같이 실행합니다:

```
$ cd /u01/app/oracle/orainstall/clusterware/cluvfy
$ ./runcluvfy.sh stage -post hwos -n linux1,linux2 -verbose
```

CVU 리포트를 검토합니다. 앞에서와 마찬가지로, VIP 인터페이스에 관련한 에러는 무시해도 좋습니다.

또 공유 스토리지 접근 테스트 과정에서 에러가 발생할 것입니다.

```
Checking shared storage accessibility...
```

```
WARNING:
Unable to determine the sharedness of /dev/sde on nodes:
linux2,linux2,linux2,linux2,linux2,linux1,linux1,linux1,linux1,linux1
```

Shared storage check failed on nodes "linux2,linux1".

이 에러 역시 무시해도 좋습니다. 디스크가 양쪽 Oracle RAC 노드에 의해 공유되고 있음을 이미 확인한 바 있습니다. 그럼에도 불구하고 이 테스트가 실패하는 이유에 대해서는 별도 문서를 통해 설명되고 있습니다. 먼저 Metalink에서는 cluvfy가 SCSI 디바이스를 제외한 다른 디바이스 환경에서 정상적으로 동작하지 않음을 보고하고 있습니다. EMC PowerPath, 그리고 Openfiler에서 생성한 볼륨 그룹도 여기에 해당됩니다. 이 문서가 작성되는 시점에는, 공유 디바이스를 수동으로 탐지하기 위한 해결책이 존재하지 않는 상태입니다. 또 이 에러에 관련하여 오라클의 Bane Radulovic이 작성한 문서가 있습니다. 그의 연구 결과에 따르면 Linux 환경의 CVU가 smartctl을 호출하는 과정에서 smartctl가 iSCSI 디바이스의 시리얼 넘버를 반환하지 않는 문제가 있다고 합니다. 한 예로, /dev/sde에 대한 점검 결과가 아래와 같습니다:

```
# /usr/sbin/smartctl -i /dev/sde
smartctl version 5.33 [i686-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

Device: Openfile Virtual disk      Version: 0
Serial number:
Device type: disk
Local Time is: Wed Oct 25 23:41:20 2006 EDT
Device supports SMART and is Disabled
Temperature Warning Disabled or Not Supported
```

본 문서가 작성되는 시점에서, Openfiler 개발자들이 이 문제를 해결할 계획을 가지고 있는지는 확인되지 않았습니다.

페이지 1 **페이지 2** 페이지 3

Linux, iSCSI 환경에서 Oracle RAC 10g Release 2 클러스터 설치하기 (계속)

개발, 테스트 용도로만 사용할 수 있습니다. 운영 목적의 시스템에는 적용될 수 없습니다!

20. Oracle 10g Clusterware 소프트웨어의 설치

클러스터의 Oracle RAC 노드 중 하나(linux1)에서만 아래 작업을 수행합니다! Oracle Universal Installer는 클러스터의 다른 노드에 Oracle Clusterware 소프트웨어를 자동으로 설치합니다

이제 시스템 환경에 “클러스터(Oracle Clusterware 소프트웨어)”를 설치할 준비가 완료되었습니다. 앞 섹션에서 linux1 노드에서 오라클 소프트웨어를 다운로드한 뒤 /u01/app/oracle/orainstall 디렉토리에 저장한 후 압축을 풀었습니다. 이제 이 노드에서 설치 작업을 진행하면 됩니다.

Oracle Clusterware의 설치 과정에서, RAC 클러스터에 참여하는 노드에 대한 질문이 제공됩니다. 실제 설치작업이 시작되면, 필요한 소프트웨어를 원격 액세스를 통해 섹션 14 (원격 액세스를 위한 RAC 노드의 설정)에서 설정한 모든 RAC 노드에 자동으로 복사할 것입니다.

그렇다면 Oracle Clusterware가 담당하는 역할은 무엇일까요? Oracle Clusterware는 RAC의 클러스터 및 데이터베이스 구성에 관련한 메타데이터와, 일부 시스템 관리 기능을 포함하고 있습니다. DBA는 Oracle CRS를 이용하여 오라클 인스턴스를 클러스터에 등록합니다. CRS는 (ping 시그널을 통해) 클러스터의 모든 노드에 “하트비트(heartbeat)” 메시지를 전송합니다. 노드가 하트비트 메시지의 수신에 실패하는 경우, (공유 디스크에 저장된) CRS 구성을 점검하여 실제 노드에 장애가 발생했는지, 아니면 네트워크에 장애가 발생했는지 판별합니다.

Oracle Clusterware를 설치하고 난 뒤, Oracle Universal Installer (OUI)를 사용하여 Oracle Database 10g 소프트웨어를 설치하게 됩니다. 이때 Oracle Universal Installer는 CRS가 설치된 노드를 자동으로 인식합니다. CRS 설치작업과 마찬가지로, Oracle 10g 데이터베이스 소프트웨어 역시 클러스터 노드 중 하나에만 설치됩니다. OUI는 RAC 클러스터의 다른 노드에 소프트웨어 패키지를 자동으로 복사합니다.

Oracle Clusterware 공유 파일

Oracle Clusterware가 사용하는 두 가지 공유 파일은 앞에서 생성한 OCFS2에 저장됩니다. 공유되는 두 가지 Oracle Clusterware 파일이 다음과 같습니다:

Oracle Cluster Registry (OCR)

파일 1 : /u02/oradata/orcl/OCRFile

파일 2 : /u02/oradata/orcl/OCRFile_mirror

사이즈 : (2 * 100MB) = 200M

CRS Voting Disk

파일 1 : /u02/oradata/orcl/CSSFile
 파일 2 : /u02/oradata/orcl/CSSFile_mirror1
 파일 3 : /u02/oradata/orcl/CSSFile_mirror2
 사이즈 : (3 * 20MB) = 60MB

Note: 두 가지 Clusterware 파일(OCR, Clusterware Voting Disk)에 대해 ASM을 적용하는 것은 현재로서는 불가능합니다. 이 파일은 오라클 인스턴스가 시작되기 전에 액세스 가능해야 하기 때문입니다. ASM을 사용하려면, 먼저 ASM 인스턴스가 실행되어야만 합니다.

하지만 두 파일을 OCFS2, 공유 로우 디바이스(shared raw device), 또는 써드-파티 클러스터 파일 시스템에 저장하는 것은 가능합니다.

터미널 셸 환경의 검증

OUI를 시작하기 전에, 먼저 root 계정으로 콘솔에서 xhost 커맨드를 실행하여 X Server 연결을 허용합니다. 그런 다음 oracle 사용자 계정으로 로그인합니다. 원격 클라이언트를 사용하여 설치를 진행하는 노드에 접근하는 경우(예: X Server가 설치된 워크스테이션에서 SSH/Telnet을 이용하여 linux1에 접근하는 경우) 로컬 워크스테이션을 참조하도록 DISPLAY 변수를 설정해야 합니다. 마지막으로, 클러스터의 모든 노드에 대해 원격 액세스 / user equivalence를 확인합니다.

X Server 액세스 환경의 검증

```
# hostname
linux1
# xhost +
access control disabled, clients can connect from any host
```

oracle 사용자 계정으로 로그인 한 후 DISPLAY 설정 (필요한 경우)

```
# su - oracle
$ # IF YOU ARE USING A REMOTE CLIENT TO CONNECT TO THE
$ # NODE PERFORMING THE INSTALL
$ DISPLAY=<your local workstation>:0.0
$ export DISPLAY
```

원격 액세스 / User Equivalence의 검증

OUI가 실행되는 서버에서 별도의 패스워드를 입력하지 않고도 다른 모든 서버에 대해 *SSH(Secure Shell)* 커맨드(ssh, scp) 또는 *RSH(Remote Shell)* 커맨드(rsh, rcp)를 실행할 수 있는지 확인합니다.

SSH(secure shell)을 사용하는 경우에는 OUI를 실행하기 전에 터미널 세션에 "user equivalence"를 활성화해야 합니다. 현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행하고 생성된 각 키를 위한 pass phrase가 입력될 수 있도록 합니다.

```
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
$ ssh linux1 "date;hostname"
Wed Sep 13 18:27:17 EDT 2006
linux1
$ ssh linux2 "date;hostname"
Wed Sep 13 18:28:00 EDT 2006
linux2
```

RSH(remote shell)을 사용하는 경우에는 user equivalence가 /etc/hosts.equiv 파일에 정의되며, 새로운 터미널 셸 세션에 대해 기본적으로 활성화됩니다.

```
$ rsh linux1 "date;hostname"
```

```
Tue Sep 12 12:37:17 EDT 2006
linux1
$ rsh linux2 "date:hostname"
Tue Sep 12 12:38:00 EDT 2006
linux2
```

Cluster Ready Services의 설치

아래 명령을 실행하여 Oracle Clusterware를 설치합니다:

```
$ cd ~oracle
$ /u01/app/oracle/orainstall/clusterware/runInstaller -ignoreSysPrereqs
```

스크린 이름	작업 내용									
Welcome Screen	Next를 클릭합니다.									
Specify Inventory directory and credentials	디폴트 값을 그대로 사용합니다: Inventory directory: /u01/app/oracle/orainventory Operating System group name: dba									
Specify Home Details	ORACLE_HOME(\$ORA_CRS_HOME)의 Name과 Path를 다음과 같이 설정합니다: Name: OraCrs10g_home Path: /u01/app/oracle/product/crs									
Product-Specific Prerequisite Checks	인스톨러가 Oracle Cluster 소프트웨어의 설치 및 구성에 필요한 최소 요구사항을 검증하기 위해 몇 가지 점검 작업을 수행합니다. 점검 과정이 실패하는 경우, 해당 메시지를 클릭하는 방법으로 수작업 검증을 수행해 주어야 합니다. 필자의 경우 모든 점검 과정을 완료할 수 있었습니다. Next를 클릭하여 다음 단계로 진행합니다.									
Specify Cluster Configuration	Cluster Name: crs <table border="1"> <thead> <tr> <th>Public Node Name</th> <th>Private Node Name</th> <th>Virtual Node Name</th> </tr> </thead> <tbody> <tr> <td>linux1</td> <td>linux1-priv</td> <td>linux1-vip</td> </tr> <tr> <td>linux2</td> <td>linux2-priv</td> <td>linux2-vip</td> </tr> </tbody> </table>	Public Node Name	Private Node Name	Virtual Node Name	linux1	linux1-priv	linux1-vip	linux2	linux2-priv	linux2-vip
Public Node Name	Private Node Name	Virtual Node Name								
linux1	linux1-priv	linux1-vip								
linux2	linux2-priv	linux2-vip								
Specify Network Interface Usage	<table border="1"> <thead> <tr> <th>Interface Name</th> <th>Subnet</th> <th>Interface Type</th> </tr> </thead> <tbody> <tr> <td>eth0</td> <td>192.168.1.0</td> <td>Public</td> </tr> <tr> <td>eth1</td> <td>192.168.2.0</td> <td>Private</td> </tr> </tbody> </table>	Interface Name	Subnet	Interface Type	eth0	192.168.1.0	Public	eth1	192.168.2.0	Private
Interface Name	Subnet	Interface Type								
eth0	192.168.1.0	Public								
eth1	192.168.2.0	Private								
Specify OCR Location	Oracle Database 10g Release 2 (10.2)부터 Oracle Clusterware는 Mirror의 생성 기능을 추가하고 클러스터 안정성을 개선하였습니다. 본 문서에서 설명하는 "Normal Redundancy"를 사용하여 COR 파일을 미러링하도록 설정합니다: Specify OCR Location: /u02/oradata/orcl/OCRFile Specify OCR Mirror Location: /u02/oradata/orcl/OCRFile_mirror									
Specify Voting Disk Location	Oracle Database 10g RAC Release 2 (10.2)부터, 다수의 Voting Disk 기능이 RAC, CSS에 추가되었습니다. Release 1(10.1)에서는 단 하나의 Voting Disk만을 설정할 수 있었습니다. Voting Disk를 추가로 설정하는 기능을 여러 개의 Voting Disk를 서로 독립된 물리적 디스크에 분산시킬 수 있는 기능은 iSCSI 네트워크 프로토콜과 기타 NAS(Network Attached Storage) 솔루션을 지원합니다. 여러 개의 Voting Disk를 동시에 사용하는 경우의 Voting Disk가 필요함을 참고하십시오. 본 문서에서는 디폴트 설정인 "Normal Redundancy"를 사용하여 Voting Disk의 미러링을 수행하도록 선택합니다. Voting Disk Location: /u02/oradata/orcl/CSSFile Additional Voting Disk 1 Location: /u02/oradata/orcl/CSSFile_mirror Additional Voting Disk 2 Location: /u02/oradata/orcl/CSSFile_mirror									
Summary	Install을 눌러 설치 작업을 시작합니다!									
	설치가 완료되면 oraInstRoot.sh와 root.sh 스크립트를 실행하라는 메시지가 표시됩니다. root 계정을 사용하여 RAC 클러스터의 각 노드에서 새로운 콘솔을 실행합니다. RAC 클러스터의 모든 노드에서 /u01/app/oracle/orainventory 디렉토리를 생성합니다.									

	<p>후 orainstRoot.sh를 실행합니다</p> <hr/> <p>RAC 클러스터의 각 노드에서 열린 콘솔 윈도우에 "root" 사용자 계정으 상태를 그대로 유지합니다. /u01/app/oracle/product/crs 디렉토리로 이동하여 클러스터 각 노드의 파일을 찾습니다. root.sh 파일을 실행합니다. 이 작업을 RAC 클러스터 드에서 순차적으로 실행합니다. root.sh 스크립트를 실행할 때 여러 가지 경고 메시지가 뜨겠지만, 무시 다. root.sh 스크립트의 실행에는 다소 시간이 걸립니다. 마지막 노드에서 r 행하고 나면, 아래와 같은 critical error를 확인하게 됩니다:</p> <pre> ... Expecting the CRS daemons to be up within 600 seconds. CSS is active on these nodes. linux1 linux2 CSS is active on all nodes. Waiting for the Oracle CRSD and EVMD to start Oracle CRS stack installed and running under init(1M) Running vipca(silent) for configuring nodeapps The given interface(s), "eth0" is not public. Public interfaces should be used to configure virtual IPs. </pre> <p>이 문제는 Oracle 10.2.0.1에 한정된 문제(Metalink 문서 4437727)이며 로 진행하기 전에 해결하고 넘어가야 합니다. 가장 간단한 우회책은 에러 마지막 노드에서 root 계정으로 vipca(GUI)를 재실행하는 것입니다. . vi 이며, 따라서 DISPLAY 변수를 사용자의 X 서버 환경에 맞게 설정해 주 고하십시오: # \$ORA_CRS_HOME/bin/vipca "VIP Configuration Assistant"에서 필자가 입력한 내용이 아래와 같습니 Welcome:Next를 클릭 Network interfaces:두 인터페이스(eth0, eth1)를 모두 선택 Virtual IPs for cluster nodes: Node Name: linux1 IP Alias Name: linux1-vip IP Address: 192.168.1.200 Subnet Mask: 255.255.255.0 Node Name: linux2 IP Alias Name: linux2-vip IP Address: 192.168.1.201 Subnet Mask: 255.255.255.0 Summary:Finish를 클릭 Configuration Assistant Progress Dialog:설정이 완료되면 OK를 클릭 Configuration Results:Exit를 클릭 OUI로 돌아가서 "Execute Configuration scripts" 대화상자를 확인합니</p>
End of installation	설치가 완료되면 OUI를 종료합니다.

Oracle Clusterware 설치 결과의 검증

Oracle Clusterware의 설치가 완료되면, 몇 가지 테스트를 통해 설치가 성공적으로 완료되었는지 점검
합니다. RAC 클러스터의 모든 노드에 대해 아래 커맨드를 실행합니다.

클러스터 노드의 점검

```

$ /u01/app/oracle/product/crs/bin/olsnodes -n
linux1 1
linux2 2

```

Oracle Clusterware Auto-Start 스크립트의 확인

```

$ ls -l /etc/init.d/init.*
-r-xr-xr-x 1 root root 1951 Oct 4 14:21 /etc/init.d/init.crs*
-r-xr-xr-x 1 root root 4714 Oct 4 14:21 /etc/init.d/init.crsd*

```

```
-r-xr-xr-x 1 root root 35394 Oct 4 14:21 /etc/init.d/init.cssd*
-r-xr-xr-x 1 root root 3190 Oct 4 14:21 /etc/init.d/init.evmd*
```

21. Oracle Database 10g 소프트웨어의 설치

클러스터의 Oracle RAC 노드 중 하나(*linux1*)에서만 아래 작업을 수행합니다! Oracle Universal Installer는 클러스터의 다른 노드에 Oracle Database 소프트웨어를 자동으로 설치합니다

Oracle Clusterware 소프트웨어의 설치를 성공적으로 완료했다면, 다음은 Oracle Database 10g (10.2.0.1.0) RAC를 설치할 차례입니다.

본 구성 예제에서는 "Create Database" 옵션을 사용하지 않고, 그 대신 설치가 완료된 후 DBCA (Database Creation Assistant)를 이용하여 데이터베이스를 생성하는 방법을 사용합니다.

Oracle Clusterware 설치작업과 마찬가지로, Oracle 10g 데이터베이스 소프트웨어 설치 작업 역시 노드 중 하나에서만 실행됩니다. OUI는 RAC 클러스터의 다른 노드에 소프트웨어 패키지를 자동으로 복사합니다.

터미널 셸 환경의 검증

앞 섹션(*Oracle10g Clusterware 소프트웨어의 설치*)에서 설명한 것처럼, Oracle Universal Installer를 실행하기 전에 모든 노드에 대해 원격 액세스 및 user equivalence를 설정해 주어야 합니다. 앞 섹션에서 사용한 것과 동일한 터미널 셸 세션을 사용할 수 있으며, 같은 세션을 사용한 경우 아래의 작업(원격 액세스 설정 및 DISPLAY 변수 설정)을 수행할 필요가 없음을 참고하시기 바랍니다:

oracle 사용자 계정으로 로그인 한 후 DISPLAY 설정 (필요한 경우)

```
# su - oracle
$ # IF YOU ARE USING A REMOTE CLIENT TO CONNECT TO THE
$ # NODE PERFORMING THE INSTALL
$ DISPLAY=<your local workstation>:0.0
$ export DISPLAY
```

원격 액세스 / User Equivalence의 검증

OUI가 실행되는 서버에서 별도의 패스워드를 입력하지 않고도 다른 모든 서버에 대해 SSH(*Secure Shell*) 커맨드(ssh, scp) 또는 RSH(*Remote Shell*) 커맨드(rsh, rcp)를 실행할 수 있는지 확인합니다.

SSH(secure shell)을 사용하는 경우에는 OUI를 실행하기 전에 터미널 세션에 "user equivalence"를 활성화해야 합니다. 현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행하고 생성된 각 키를 위한 pass phrase가 입력될 수 있도록 합니다.

```
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
$ ssh linux1 "date;hostname"
Wed Sep 13 18:27:17 EDT 2006
linux1
$ ssh linux2 "date;hostname"
Wed Sep 13 18:28:00 EDT 2006
linux2
```

RSH(remote shell)을 사용하는 경우에는 user equivalence가 /etc/hosts.equiv 파일에 정의되며, 새로운 터미널 셸 세션에 대해 기본적으로 활성화됩니다.

```
$ rsh linux1 "date;hostname"
Tue Sep 12 12:37:17 EDT 2006
linux1
$ rsh linux2 "date;hostname"
Tue Sep 12 12:38:00 EDT 2006
linux2
```

Oracle Cluster Verification Utility의 실행

오라클 데이터베이스 소프트웨어를 설치하기 전에, Cluster Verification Utility(CVU)를 이용하여 사전 점검 작업을 수행해야 합니다.

참고: CVU 설정 방법은 "Cluster Verification Utility 실행을 위한 사전 요구사항 섹션을 참고하시기 바랍니다.

```
$ cd /u01/app/oracle/orainstall/clusterware/cluvfy
$ ./runcluvfy.sh stage -pre dbinst -n linux1,linux2 -r 10gR2 -verbose
```

CVU 리포트를 검토합니다. CRS를 위한 사전 점검 작업 결과에서와 동일한 에러가 표시됨을 확인할 수 있습니다(VIP 인터페이스 설정 관련 에러, RHEL4 Update에 존재하지 않는 RPM 패키지 관련 에러 등). 이 두 가지 에러는 무시해도 좋습니다.

Oracle Database 10g Release 2 소프트웨어 설치

Oracle Database 10g Release 2 소프트웨어를 설치합니다:

```
$ cd ~oracle
$ /u01/app/oracle/orainstall/database/runInstaller -ignoreSysPrereqs
```

스크린 이름	작업내용
Welcome Screen	Next를 클릭합니다.
Select Installation Type	필자의 경우 Enterprise Edition 옵션을 선택하였습니다.
Specify Home Details	ORACLE_HOME의 Name과 Path를 아래와 같이 설정합니다: Name: OraDb10g_home1 Path: /u01/app/oracle/product/10.2.0/db_1
Specify Hardware Cluster Installation Mode	Cluster Installation 옵션을 선택하고 모든 노드를 선택합니다. Select All을 클릭하여 모든 서버(linux1과 linux2)를 선택합니다. 만일 설치가 중단되고 RAC 노드 중 하나의 상태가 "Node not reachable"로 표시되는 경우, 다음과 같이 점검 작업을 수행합니다: 문제가 되는 노드에 Clusterware가 실행 중인지 확인합니다. 설치 작업을 수행하는 노드에서 문제가 되는 노드에 연결할 수 있는지 테스트합니다.
Product-Specific Prerequisite Checks	인스톨러는 오라클 데이터베이스 소프트웨어의 설치 및 구성에 필요한 최소요구조건을 노드가 만족하는지 확인하는 일련의 점검 작업을 수행합니다. 점검 과정이 실패하는 경우, 체크박스를 클릭하는 방법으로 수작업 검증을 수행해 주어야 합니다. Next를 클릭하여 다음 단계로 진행합니다.
Select Database Configuration	"Install database software only" 옵션을 선택합니다. 클러스터 데이터베이스의 생성 작업은 DBCA를 사용하여 별도로 수행하기로 했음을 기억하시기 바랍니다.
Summary	Install을 눌러 설치 작업을 시작합니다!
Root Script Window - Run root.sh	설치가 완료되면, root.sh 스크립트를 실행하라는 메시지가 표시됩니다. RAC 클러스터의 모든 노드에서 root.sh을 실행하되 (데이터베이스 설치를 시작한 서버로부터 시작하여) 한 번에 하나씩 실행하도록 합니다. 먼저, Oracle 10g 데이터베이스 소프트웨어를 설치한 노드(linux1)에서 새로운 콘솔 창을 엽니다. /u01/app/oracle/product/10.2.0/db_1 디렉토리로 이동하여 root.sh를 실행합니다. root.sh 스크립트의 실행이 완료되면, OUI로 돌아가 "Execute Configuration scripts" 대화상자를 확인합니다
End of installation	설치가 완료되면 OUI를 종료합니다.

22. Oracle 10g Companion CD 소프트웨어의 설치
클러스터의 Oracle RAC 노드 중 하나(linux1)에서만 아래 작업을 수행합니다! Oracle Universal Installer는 클러스터의 다른 노드에 Oracle10g Companion CD 소프트웨어를 자동으로 설치합니다.

Oracle Database 소프트웨어의 설치를 성공적으로 완료했다면, 다음은 Oracle Database 10g Release 2 Companion CD 소프트웨어(10.2.0.1.0)를 설치할 차례입니다.

이 작업은 필수적으로 요구되는 것은 아님을 참고하십시오. 본 구성 예제에서는 Java Virtual Machine (Java VM)과 Oracle *interMedia*를 사용하고 있기 때문에 Oracle Database 10g Companion CD를 설치하는 것입니다. 이때 설치 옵션으로는 “Oracle Database 10g Products”를 선택해야 합니다.

이 설치 옵션에는 Java 성능의 개선을 위한 NCOMP(Natively Compiled Java Libraries) 파일이 포함되어 있습니다. Java VM을 사용하는 데이터베이스에서 NCOMP 파일을 설치하지 않은 상태에서 패치를 적용하면 “ORA-29558:JAccelerator (NCOMP) not installed” 에러가 뜨게 됩니다. After successfully installing the Oracle Database software, the next step is to install the Oracle 10g Release 2 Companion CD software (10.2.0.1.0).

터미널 셸 환경의 검증

앞 섹션(Oracle10g Database 소프트웨어의 설치)에서 설명한 것처럼, Oracle Universal Installer를 실행하기 전에 모든 노드에 대해 원격 액세스 및 user equivalence를 설정해 주어야 합니다. 앞 섹션에서 사용한 것과 동일한 터미널 셸 세션을 사용할 수 있으며, 같은 세션을 사용한 경우 아래의 작업(원격 액세스 설정 및 DISPLAY 변수 설정)을 수행할 필요가 없음을 참고하시기 바랍니다:

oracle 사용자 계정으로 로그인 한 후 DISPLAY 설정 (필요한 경우)

```
# su - oracle
$ # IF YOU ARE USING A REMOTE CLIENT TO CONNECT TO THE
$ # NODE PERFORMING THE INSTALL
$ DISPLAY=<your local workstation>:0.0
$ export DISPLAY
```

원격 액세스 / User Equivalence의 검증

OUI가 실행되는 서버에서 별도의 패스워드를 입력하지 않고도 다른 모든 서버에 대해 SSH(Secure Shell) 커맨드(ssh, scp) 또는 RSH(Remote Shell) 커맨드(rsh, rcp)를 실행할 수 있는지 확인합니다.

SSH(secure shell)을 사용하는 경우에는 OUI를 실행하기 전에 터미널 세션에 "user equivalence"를 활성화해야 합니다. 현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행하고 생성된 각 키를 위한 pass phrase가 입력될 수 있도록 합니다.

```
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
$ ssh linux1 "date:hostname"
Wed Sep 13 18:27:17 EDT 2006
linux1
$ ssh linux2 "date:hostname"
Wed Sep 13 18:28:00 EDT 2006
linux2
```

RSH(remote shell)을 사용하는 경우에는 user equivalence가 /etc/hosts.equiv 파일에 정의되며, 새로운 터미널 셸 세션에 대해 기본적으로 활성화됩니다.

```
$ rsh linux1 "date:hostname"
Tue Sep 12 12:37:17 EDT 2006
linux1
$ rsh linux2 "date:hostname"
Tue Sep 12 12:38:00 EDT 2006
linux2
```

Oracle10g Companion CD 소프트웨어 설치

아래와 같이 Oracle10g Companion CD 소프트웨어를 설치합니다:

```
$ cd ~oracle
$ /u01/app/oracle/orainstall/companion/runInstaller -ignoreSysPrereqs
```

스크린 이름	작업 내용
Welcome Screen	Next를 클릭합니다.
Select a Product to Install	"Oracle Database 10g Products 10.2.0.1.0" 옵션을 선택합니다.
Specify Home Details	Oracle 10g Database 소프트웨어가 설치된 ORACLE_HOME의 이름과 위치를 입력합니다: Name: OraDb10g_home1 Path: /u01/app/oracle/product/10.2.0/db_1
Specify Hardware Cluster Installation Mode	클러스터의 모든 노드에 대해 Cluster 설치 옵션을 선택합니다. 디폴트 옵션을 그대로 사용하고 Next를 눌러 다음 단계로 진행합니다. 만일 설치가 중단되고 RAC 노드 중 하나의 상태가 "Node not reachable"로 표시되는 경우, 다음과 같이 점검 작업을 수행합니다: 문제가 되는 노드에 Clusterware가 실행 중인지 확인합니다. 설치 작업을 수행하는 노드에서 문제가 되는 노드에 연결할 수 있는지 테스트합니다.
Product-Specific Prerequisite Checks	인스톨러는 Companion CD 소프트웨어의 설치 및 구성에 필요한 최소요구조건을 노드가 만족하는지 확인하는 일련의 점검 작업을 수행합니다. 점검 과정이 실패하는 경우, 체크박스를 클릭하는 방법으로 수작업 검증을 수행해 주어야 합니다. 필자의 경우 아무런 문제 없이 모든 점검 과정을 완료할 수 있었습니다. Next를 클릭하여 다음 단계로 진행합니다.
Summary	Install을 눌러 설치 작업을 시작합니다!
End of installation	설치가 완료되면 OUI를 종료합니다.

23. TNS 리스너 프로세스 생성

클러스터의 Oracle RAC 노드 중 하나(linux1)에서만 아래 작업을 수행합니다! Network Configuration Assistant(NETCA)는 클러스터의 모든 노드에 대해 TNS listener의 설정 작업을 자동으로 수행합니다. DBCA를 이용하여 클러스터 데이터베이스를 생성하려면, Oracle TNS Listener 프로세스가 RAC 클러스터의 모든 노드에서 실행 중이어야 합니다.

TNS Listener를 생성하는 작업은 클러스터 노드 중 하나에서만 수행됩니다. 모든 변경 내역은 클러스터의 다른 노드에 자동으로 복제됩니다. 본 구성 예제에서는 linux1 서버에서 Network Configuration Assistant (NETCA)를 실행하고 새로운 TNS listener 프로세스를 생성하고 로컬 액세스를 위한 노드를 설정하기로 합니다.

터미널 셸 환경의 검증

앞 섹션(Oracle10g Companion CD 소프트웨어의 설치)에서 설명한 것처럼, Oracle Universal Installer를 실행하기 전에 모든 노드에 대해 원격 액세스 및 user equivalence를 설정해 주어야 합니다. 앞 섹션에서 사용한 것과 동일한 터미널 셸 세션을 사용할 수 있으며, 같은 세션을 사용한 경우 아래의 작업(원격 액세스 설정 및 DISPLAY 변수 설정)을 수행할 필요가 없음을 참고하시기 바랍니다:

oracle 사용자 계정으로 로그인 한 후 DISPLAY 설정 (필요한 경우)

```
# su - oracle
$ # IF YOU ARE USING A REMOTE CLIENT TO CONNECT TO THE
$ # NODE PERFORMING THE INSTALL
$ DISPLAY=<your local workstation>:0.0
$ export DISPLAY
```

원격 액세스 / User Equivalence의 검증

OUI가 실행되는 서버에서 별도의 패스워드를 입력하지 않고도 다른 모든 서버에 대해 SecureSSH(Secure Shell) 커맨드(ssh, scp) 또는 RSH(Remote Shell) 커맨드(rsh, rcp)를 실행할 수 있는지 확인합니다. SSH(secure shell)을 사용하는 경우에는 OUI를 실행하기 전에 터미널 세션에 "user

equivalence"를 활성화해야 합니다. 현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행하고 생성된 각 키를 위한 pass phrase가 입력될 수 있도록 합니다.

```
$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
$ ssh linux1 "date;hostname"
Wed Sep 13 18:27:17 EDT 2006
linux1
$ ssh linux2 "date;hostname"
Wed Sep 13 18:28:00 EDT 2006
linux2
```

RSH(remote shell)을 사용하는 경우에는 user equivalence가 /etc/hosts.equiv 파일에 정의되며, 새로운 터미널 셸 세션에 대해 기본적으로 활성화됩니다.

```
$ rsh linux1 "date;hostname"
Tue Sep 12 12:37:17 EDT 2006
linux1
$ rsh linux2 "date;hostname"
Tue Sep 12 12:38:00 EDT 2006
linux2
```

Network Configuration Assistant의 실행

아래와 같이 NETCA를 실행합니다:

```
$ netca &
```

RAC 환경을 위한 새로운 Oracle 리스너를 생성하는 방법이 아래와 같습니다.

스크린 이름	작업 내용
Select the Type of Oracle Net Services Configuration	Cluster Configuration을 선택합니다.
Select the nodes to configure	모든 노드(linux1, linux2)를 선택합니다.
Type of Configuration	Listener configuration을 선택합니다.
Listener Configuration – Next 6 Screens	Oracle Listener의 설정 방법은 일반적인 listener 설정 방법과 동일합니다. 아래 6개 항목에 대해 디폴트 매개변수를 그대로 사용합니다: What do you want to do: Add Listener name: LISTENER Selected protocols: TCP Port number: 1521 Configure another listener: No Listener configuration complete! [Next] 모든 과정이 완료되면 다시 Welcome (Type of Configuration) 스크린이 표시됩니다.
Type of Configuration	Naming Methods configuration을 선택합니다.
Naming Methods Configuration	아래와 같이 입력합니다: Selected Naming Methods: Local Naming Naming Methods configuration complete! [Next] 다시 Welcome (Type of Configuration) 스크린이 표시됩니다.
Type of Configuration	Finish를 클릭하고 NETCA를 종료합니다.

이제 RAC 클러스터의 모든 노드에서 Oracle TNS listener 프로세스가 실행되고 있음을 확인할 수 있습니다.

```

$ hostname
linux1
$ ps -ef | grep lsnr | grep -v 'grep' | grep -v 'ocfs' | awk '{print $9}'
LISTENER_LINUX1
=====
$ hostname
linux2
$ ps -ef | grep lsnr | grep -v 'grep' | grep -v 'ocfs' | awk '{print $9}'
LISTENER_LINUX2

```

24. 오라클 클러스터 데이터베이스의 생성

클러스터의 Oracle RAC 노드 중 하나(linux1)에서만 데이터베이스 생성 작업을 수행합니다!

여기에서는 클러스터 데이터베이스 생성을 위해 DBCA를 사용하기로 합니다.

DBCA를 실행하기 전에, \$ORACLE_HOME, \$PATH 환경변수가 \$ORACLE_BASE/product/10.2.0/db_1 환경으로 설정되었는지 확인합니다.

또 지금까지 설치한 모든 서비스(Oracle TNS listener, Clusterware 프로세스 등)가 정상적으로 실행 중인지 확인합니다.

터미널 셸 환경의 검증

앞 섹션(*TNS 리스너 소프트웨어의 설치*)에서 설명한 것처럼, DBCA(*Database Configuration Assistance*)를 실행하기 전에 모든 노드에 대해 원격 액세스 및 user equivalence를 설정해 주어야 합니다. 앞 섹션에서 사용한 것과 동일한 터미널 셸 세션을 사용할 수 있으며, 같은 세션을 사용한 경우 아래의 작업(원격 액세스 설정 및 DISPLAY 변수 설정)을 수행할 필요가 없음을 참고하시기 바랍니다:

oracle 사용자 계정으로 로그인 한 후 DISPLAY 설정 (필요한 경우)

```

# su - oracle
$ # IF YOU ARE USING A REMOTE CLIENT TO CONNECT TO THE
$ # NODE PERFORMING THE INSTALL
$ DISPLAY=<your local workstation>:0.0
$ export DISPLAY

```

원격 액세스 / User Equivalence의 검증

OUI가 실행되는 서버에서 별도의 패스워드를 입력하지 않고도 다른 모든 서버에 대해 SSH(*Secure Shell*) 커맨드(ssh, scp) 또는 RSH(*Remote Shell*) 커맨드(rsh, rcp)를 실행할 수 있는지 확인합니다.

SSH(secure shell)을 사용하는 경우에는 OUI를 실행하기 전에 터미널 세션에 "user equivalence"를 활성화해야 합니다. 현재 터미널 셸 세션에 대해 'user equivalence'를 활성화하려면 아래와 같이 실행하고 생성된 각 키를 위한 pass phrase가 입력될 수 있도록 합니다.

```

$ exec /usr/bin/ssh-agent $SHELL
$ /usr/bin/ssh-add
Enter passphrase for /u01/app/oracle/.ssh/id_rsa: xxxxx
Identity added: /u01/app/oracle/.ssh/id_rsa (/u01/app/oracle/.ssh/id_rsa)
Identity added: /u01/app/oracle/.ssh/id_dsa (/u01/app/oracle/.ssh/id_dsa)
$ ssh linux1 "date:hostname"
Wed Sep 13 18:27:17 EDT 2006
linux1
$ ssh linux2 "date:hostname"
Wed Sep 13 18:28:00 EDT 2006
linux2

```

RSH(remote shell)을 사용하는 경우에는 user equivalence가 /etc/hosts.equiv 파일에 정의되며, 새로운 터미널 셸 세션에 대해 기본적으로 활성화됩니다.

```

$ rsh linux1 "date:hostname"
Tue Sep 12 12:37:17 EDT 2006

```

```
linux1
$ rsh linux2 "date:hostname"
Tue Sep 12 12:38:00 EDT 2006
linux2
```

Oracle Cluster Verification Utility의 실행

클러스터 데이터베이스를 생성하기 전에, Cluster Verification Utility(CVU)를 이용하여 사전 점검 작업을 수행해야 합니다.

참고: CVU 설정 방법은 "Cluster Verification Utility 실행을 위한 사전 요구사항 섹션을 참고하시기 바랍니다.

```
$ cd /u01/app/oracle/orainstall/clusterware/cluvfy
$ ./runcluvfy.sh stage -pre dbcfg -n linux1,linux2 -d ${ORACLE_HOME} -verbose
```

CVU 리포트를 검토합니다. CRS를 위한 사전 점검 작업 결과에서와 동일한 에러가 표시됨을 확인할 수 있습니다(VIP 인터페이스 설정 관련 에러, RHEL4 Update에 존재하지 않는 RPM 패키지 관련 에러 등). 이 에러 역시 무시해도 좋습니다.

클러스터 데이터베이스의 생성

아래와 같이 데이터베이스 생성 작업을 시작합니다:

```
$ dbca &
```

스크린 이름	작업 내용
Welcome Screen	"Oracle Real Application Clusters database"를 선택합니다.
Operations	Create a Database를 선택합니다.
Node Selection	Select All을 클릭하여 모든 서버(linux1과 linux2)를 선택합니다.
Database Templates	Custom Database를 선택합니다.
Database Identification	다음과 같이 입력합니다: Global Database Name: orcl.idevelopment.info SID Prefix: orcl 필자의 경우 데이터베이스 도메인으로 idevelopment.info을 사용하였습니다. 도메인은 어떻게 설정하든 상관이 없습니다. 도메인이 반드시 DNS 도메인과 일치할 필요가 없음을 참고하시기 바랍니다
Management Option	디폴트 옵션을 그대로 사용합니다 ("Configure the Database with Enterprise Manager / Use Database Control for Database Management").
Database Credentials	필자의 경우 "Use the Same Password for All Accounts"을 선택하였습니다 패스워드를 2회 입력하되, 패스워드가 숫자로 시작되어서는 안 된다는 점에 유의하시기 바랍니다.
Storage Options	본 구성 예제에서는 use Automatic Storage Management (ASM)을 선택하기로 합니다.
Create ASM Instance	인스턴스에서 사용할 SYS 패스워드를 입력합니다. Oracle10g Release 2부터는 ASM 인스턴스의 SPFILE(server parameter file)을 공유 디스크에 저장할 것을 요구하고 있습니다. "Create server parameter file (SPFILE)" 항목에서 OCFS2 파티션에 대한 디폴트 엔트리를 다음과 같이 수정합니다: /u02/oradata/orcl/dbs/spfile+ASM.ora. 다른 모든 옵션은 디폴트 값을 그대로 사용합니다. 다음으로 ASM 인스턴스를 생성하고 시작할 것인지 묻는 질문에 OK 버튼을 클릭하고 대화상자를 닫습니다. OUI가 RAC 클러스터의 모든 노드에 대해 ASM 인스턴스를 생성하고 시작합니다.
	Create New 버튼을 누릅니다. "Create Disk Group" 윈도우에 앞에서 ASMLib을 사용하여 설정한 4개 볼륨이 표시됩니다 "Select Member Disks" 윈도우에 앞에서 설정한 볼륨이 아래와 같이 표시되지 않는 경우: (ORCL:VOL1, ORCL:VOL2, ORCL:VOL3, ORCL:VOL4) "Change Disk Discovery Path" 버튼을 클

<p>ASM Disk Groups</p>	<p>릭하고 "ORCL:VOL*"을 입력합니다.</p> <p>필자의 경우 "Disk Group Name"으로 "ORCL_DATA1"을 사용하였습니다. "Select Member Disks" 윈도우에서 처음 두 개의 ASM 볼륨(ORCL:VOL1, ORCL:VOL2)을 선택합니다. "Redundancy" 세팅은 "Normal"로 설정합니다.</p> <p>윈도우의 모든 설정값을 확인한 뒤 [OK] 버튼을 클릭합니다. 그러면 "ASM Disk Group Creation" 윈도우가 뜨게 됩니다. ASM Disk Group Creation 프로세스가 완료되면 다시 "ASM Disk Groups" 윈도우가 표시됩니다.</p> <p>Create New 버튼을 다시 누릅니다. 두 번째 "Disk Group Name"으로는 "FLASH_RECOVERY_AREA"라는 문자열을 사용하였습니다 "Select Member Disks" 윈도우에서 나머지 두 개의 ASM 볼륨(ORCL:VOL3, ORCL:VOL4)을 선택합니다. "Redundancy" 세팅은 "Normal"로 설정합니다.</p> <p>윈도우의 모든 설정값을 확인한 뒤 [OK] 버튼을 클릭합니다. 그러면 "ASM Disk Group Creation" 윈도우가 표시됩니다.</p> <p>ASM Disk Group Creation 프로세스가 완료되면 다시 "ASM Disk Groups" 윈도우를 통해 생성된 두 개의 디스크 그룹이 표시됩니다. 새로 생성된 Disk Group Name "ORCL_DATA1"의 체크박스를 클릭하고 (이때 FLASH_RECOVERY_AREA가 선택되지 않았음을 확인합니다) [Next]를 클릭하여 다음 단계로 진행합니다.</p>
<p>Database File Locations</p>	<p>필자의 경우 디폴트(Oracle Managed Files) 설정을 그대로 사용하였습니다: Database Area: +ORCL_DATA1</p>
<p>Recovery Configuration</p>	<p>"Specify Flash Recovery Area" 옵션을 체크합니다. Flash Recovery Area에 대해 [Browse] 버튼을 클릭하고 Disk Group Name을 "+FLASH_RECOVERY_AREA"로 설정합니다. 필자가 구성한 디스크 그룹의 사이즈는 118GB입니다. 필자는 117GB(117760 MB)의 Flash Recovery Area를 사용하였습니다.</p>
<p>Database Content</p>	<p>필자의 경우 모든 Database Component(destination tablespace 포함)를 디폴트 값으로 설정하였습니다. 하지만 Sample Schemas를 선택해도 무방합니다 이 옵션은 Oracle Companion CD 소프트웨어가 설치된 경우에만 사용 가능합니다.</p>
<p>Database Services</p>	<p>Add를 클릭하고 "Service Name"으로 orcltest를 입력합니다. 두 인스턴스 모두 "Preferred" 설정을 유지하고, "TAF Policy"의 경우 Basic을 선택합니다.</p>
<p>Initialization Parameters</p>	<p>환경에 맞게 매개변수를 수정합니다. 필자의 경우 디폴트 값을 그대로 사용하였습니다.</p>
<p>Database Storage</p>	<p>환경에 맞게 매개변수를 수정합니다. 필자의 경우 디폴트 값을 그대로 사용하였습니다.</p>
<p>Creation Options</p>	<p>디폴트 옵션(Create Database)을 그대로 사용하고 Finish를 클릭하여 데이터베이스 생성 프로세스를 시작합니다. "Summary" 스크린에서 OK를 클릭합니다.</p>
<p>End of Database Creation</p>	<p>데이터베이스 생성이 완료되면 DBCA를 종료합니다. DBCA를 종료하는 과정에서, 모든 오라클 인스턴스와 HA 서비스 "orcltest"를 시작한다는 메시지가 표시됩니다. 이 과정은 수 분이 걸릴 수도 있습니다. 모든 작업이 완료되면 윈도우와 대화상자가 사라집니다.</p>

DBCA의 실행을 완료하고 나면, 이제 Oracle RAC 클러스터가 정상적으로 실행됩니다!

orcltest 서비스의 생성

오라클 클러스터 데이터베이스를 생성하는 과정에서 TAF가 활성화된 데이터베이스에 연결하기 위해 "orcltest"라는 이름의 서비스를 추가하였습니다. 이 서비스는 tnsnames.ora 파일에 추가되었지만, 각 오라클 인스턴스에 서비스로서 업데이트되지는 않은 상태입니다.

아래 명령을 실행하여, orcltest 서비스가 성공적으로 추가되었는지 확인합니다:

```
SQL> show parameter service
```

NAME	TYPE	VALUE
service_names	string	orcl.idevelopment.info, orcltest

orcl.idevelopment.info에 대해서만 서비스가 정의되었다면, 양쪽 인스턴스에 서비스를 수동으로 추가해 주어야 합니다:

```
SQL> show parameter service
```

NAME	TYPE	VALUE
service_names	string	orcl.idevelopment.info

```
SQL> alter system set service_names =
  2 'orcl.idevelopment.info, orcltest.idevelopment.info' scope=both;
```

25. TNS 네트워킹 파일의 검증

클러스터의 모든 Oracle RAC 노드에 대해 TNS 네트워킹 파일을 설정해 주어야 합니다!

listener.ora

클러스터 환경에서 TNS listener 구성 파일(listener.ora)을 설정하는 방법에 대해서는 이미섹션 23에서 설명했습니다. 생성된 listener.ora 파일에 대해 별도의 수정 작업은 필요하지 않습니다.

여러분이 참고하실 수 있도록, linux1 서버의 listener.ora 파일 복사본을 가이드의 support 파일에 포함시켜 놓았습니다. 또 오라클에 의해 자동 설정된 tnsnames.ora(Transparent Application Failover (TAF)의 테스트에 사용됩니다) 파일의 복사본도 함께 포함시켜 두었습니다. 이 작업 역시 클러스터의 모든 노드에서 수행되어야 합니다.

클러스터 데이터베이스에 액세스해야 하는 다른 클라이언트에 관련한 항목도 이 파일에 포함시킬 수 있습니다.

외부 클라이언트로부터 클러스터 데이터베이스에 연결하기

이 과정은 반드시 필요한 것은 아니지만, 여기에서는 TNS 파일이 올바르게 설정되었는지 확인하기 위한 목적에서 설명하기로 합니다. 오라클(9i 또는 10g)이 설치된 다른 머신(예: 네트워크에 연결된 Windows 머신)에 관련한 항목을 클러스터 노드의 tnsnames.ora 파일에 추가합니다.

그런 다음, tnsnames.ora 파일에 정의된 service name을 사용하여 클러스터 데이터베이스에 연결할 수 있는지 테스트합니다:

```
C:\W> sqlplus system/manager@orcl2
C:\W> sqlplus system/manager@orcl1
C:\W> sqlplus system/manager@orcltest
C:\W> sqlplus system/manager@orcl
```

26. 테이블스페이스의 생성 변경

클러스터 데이터베이스를 생성하는 과정에서 모든 테이블스페이스를 디폴트 크기로 설정해 두었습니다. 공유 스토리지에 대용량 드라이브를 사용하는 경우라면, 테스트 데이터베이스의 크기를 늘려 잡아야 할 수도 있을 것입니다.

테스트 데이터베이스의 테이블 스페이스를 생성/변경하기 위한 SQL 명령이 아래와 같습니다. 본 문서에서 사용된 데이터베이스 파일명(OMF 파일)이 여러분이 생성한 오라클 환경과 다를 수 있음을 명심하시기 바랍니다. 환경에 필요한 파일명을 확인하기 위한 쿼리가 아래와 같습니다:

```
SQL> select tablespace_name, file_name
  2 from dba_data_files
  3 union
  4 select tablespace_name, file_name
  5 from dba_temp_files;
```

TABLESPACE_NAME	FILE_NAME
EXAMPLE	+ORCL_DATA1/orcl/datafile/example.257.570913311
INDX	+ORCL_DATA1/orcl/datafile/indx.270.570920045
SYSaux	+ORCL_DATA1/orcl/datafile/sysaux.260.570913287

```

SYSTEM      +ORCL_DATA1/orcl/datafile/system.262.570913215
TEMP        +ORCL_DATA1/orcl/tempfile/temp.258.570913303
UNDOTBS1    +ORCL_DATA1/orcl/datafile/undotbs1.261.570913263
UNDOTBS2    +ORCL_DATA1/orcl/datafile/undotbs2.265.570913331
USERS       +ORCL_DATA1/orcl/datafile/users.264.570913355

```

```
$ sqlplus "/ as sysdba"
```

```

SQL> create user scott identified by tiger default tablespace users;
SQL> grant dba, resource, connect to scott;

```

```

SQL> alter database datafile '+ORCL_DATA1/orcl/datafile/users.264.570913355' resize 1024m;
SQL> alter tablespace users add datafile '+ORCL_DATA1' size 1024m autoextend off;

```

```

SQL> create tablespace indx datafile '+ORCL_DATA1' size 1024m
  2 autoextend on next 50m maxsize unlimited
  3 extent management local autoallocate
  4 segment space management auto;

```

```
SQL> alter database datafile '+ORCL_DATA1/orcl/datafile/system.262.570913215' resize 800m;
```

```
SQL> alter database datafile '+ORCL_DATA1/orcl/datafile/sysaux.260.570913287' resize 500m;
```

```

SQL> alter tablespace undotbs1 add datafile '+ORCL_DATA1' size 1024m
  2 autoextend on next 50m maxsize 2048m;

```

```

SQL> alter tablespace undotbs2 add datafile '+ORCL_DATA1' size 1024m
  2 autoextend on next 50m maxsize 2048m;

```

```
SQL> alter database tempfile '+ORCL_DATA1/orcl/tempfile/temp.258.570913303' resize 1024m;
```

필자가 테스트 데이터베이스 환경을 위해 정의한 테이블스페이스의 설정 정보가 아래와 같습니다:

Status	Tablespace Name	TS Type	Ext. Mgt.	Seg. Mgt.	Tablespace Size	Used (in bytes)	Pct. Used
ONLINE	UNDOTBS1	UNDO	LOCAL	MANUAL	1,283,457,024	85,065,728	7
ONLINE	SYSAUX	PERMANENT	LOCAL	AUTO	524,288,000	275,906,560	53
ONLINE	USERS	PERMANENT	LOCAL	AUTO	2,147,483,648	131,072	0
ONLINE	SYSTEM	PERMANENT	LOCAL	MANUAL	838,860,800	500,301,824	60
ONLINE	EXAMPLE	PERMANENT	LOCAL	AUTO	157,286,400	83,820,544	53
ONLINE	INDX	PERMANENT	LOCAL	AUTO	1,073,741,824	65,536	0
ONLINE	UNDOTBS2	UNDO	LOCAL	MANUAL	1,283,457,024	3,801,088	0
ONLINE	TEMP	TEMPORARY	LOCAL	MANUAL	1,073,741,824	27,262,976	3
avg							22
sum					8,382,316,544	976,355,328	

8 rows selected.

27. RAC 클러스터 및 데이터베이스 설정의 검증

클러스터의 모든 Oracle RAC 노드에서 RAC 검증 작업을 수행해야 합니다! 하지만 본 문서에서는 Linux 서버에서만 점검 작업을 수행하기로 합니다.

이 섹션에서는 srvctl 커맨드와 SQL 쿼리를 사용하여 Oracle RAC 10g 구성을 검증하는 방법을 설명합니다.

SRVCTL을 위해 정의된 다섯 가지 노드-레벨 (node-level) 작업이 아래와 같습니다:

노드-레벨 애플리케이션의 추가 및 삭제

노드-레벨 애플리케이션 관련 환경변수의 설정 및 해제

노드 애플리케이션의 관리

ASM 인스턴스의 관리

프로그램 그룹의 시작 및 중단 (가상 IP 주소, 리스너, Oracle Notification Services, Oracle Enterprise Manager 에이전트 등).

모든 인스턴스 및 서비스의 상태 확인

```
$ srvctl status database -d orcl
Instance orcl1 is running on node linux1
Instance orcl2 is running on node linux2
```

단일 인스턴스의 상태 확인

```
$ srvctl status instance -d orcl -i orcl2
Instance orcl2 is running on node linux2
```

데이터베이스의 named service 글로벌 상태 확인

```
$ srvctl status service -d orcl -s orcltest
Service orcltest is running on instance(s) orcl2, orcl1
```

특정 노드의 노드 애플리케이션 상태 확인

```
$ srvctl status nodeapps -n linux1
VIP is running on node: linux1
GSD is running on node: linux1
Listener is running on node: linux1
ONS daemon is running on node: linux1
```

ASM 인스턴스의 상태 확인

```
$ srvctl status asm -n linux1
ASM instance +ASM1 is running on node linux1.
```

구성된 데이터베이스의 목록 확인

```
$ srvctl config database
orcl
```

RAC 데이터베이스의 설정 확인

```
$ srvctl config database -d orcl
linux1 orcl1 /u01/app/oracle/product/10.2.0/db_1
linux2 orcl2 /u01/app/oracle/product/10.2.0/db_1
```

특정 클러스터 데이터베이스에 관련된 모든 서비스를 확인

```
$ srvctl config service -d orcl
orcltest PREF: orcl2 orcl1 AVAIL:
```

노드 애플리케이션의 설정 확인 - (VIP, GSD, ONS, Listener 등)

```
$ srvctl config nodeapps -n linux1 -a -g -s -l
VIP exists.: /linux1-vip/192.168.1.200/255.255.255.0/eth0:eth1
GSD exists.
ONS daemon exists.
Listener exists.
```

ASM 인스턴스의 설정 확인

```
$ srvctl config asm -n linux1
```

```
+ASM1 /u01/app/oracle/product/10.2.0/db_1
```

클러스터에서 실행 중인 모든 인스턴스 확인

```
SELECT
  inst_id
  , instance_number inst_no
  , instance_name inst_name
  , parallel
  , status
  , database_status db_status
  , active_state state
  , host_name host
FROM gv$instance
ORDER BY inst_id;
```

INST_ID	INST_NO	INST_NAME	PAR	STATUS	DB_STATUS	STATE	HOST
1	1	orcl1	YES	OPEN	ACTIVE	NORMAL	linux1
2	2	orcl2	YES	OPEN	ACTIVE	NORMAL	linux2

디스크 그룹에 포함된 모든 데이터 파일 확인

```
select name from v$datafile
union
select member from v$logfile
union
select name from v$controlfile
union
select name from v$tempfile;
```

```
NAME
```

```
+FLASH_RECOVERY_AREA/orcl/controlfile/current.258.570913191
+FLASH_RECOVERY_AREA/orcl/onlinelog/group_1.257.570913201
+FLASH_RECOVERY_AREA/orcl/onlinelog/group_2.256.570913211
+FLASH_RECOVERY_AREA/orcl/onlinelog/group_3.259.570918285
+FLASH_RECOVERY_AREA/orcl/onlinelog/group_4.260.570918295
+ORCL_DATA1/orcl/controlfile/current.259.570913189
+ORCL_DATA1/orcl/datafile/example.257.570913311
+ORCL_DATA1/orcl/datafile/indx.270.570920045
+ORCL_DATA1/orcl/datafile/sysaux.260.570913287
+ORCL_DATA1/orcl/datafile/system.262.570913215
+ORCL_DATA1/orcl/datafile/undotbs1.261.570913263
+ORCL_DATA1/orcl/datafile/undotbs1.271.570920865
+ORCL_DATA1/orcl/datafile/undotbs2.265.570913331
+ORCL_DATA1/orcl/datafile/undotbs2.272.570921065
+ORCL_DATA1/orcl/datafile/users.264.570913355
+ORCL_DATA1/orcl/datafile/users.269.570919829
+ORCL_DATA1/orcl/onlinelog/group_1.256.570913195
+ORCL_DATA1/orcl/onlinelog/group_2.263.570913205
+ORCL_DATA1/orcl/onlinelog/group_3.266.570918279
+ORCL_DATA1/orcl/onlinelog/group_4.267.570918289
+ORCL_DATA1/orcl/tempfile/temp.258.570913303
```

```
21 rows selected.
```

'ORCL_DATA1' 디스크 그룹에 포함된 모든 ASM 디스크 확인

```
SELECT path
FROM v$asm_disk
WHERE group_number IN (select group_number
                        from v$asm_diskgroup
                        where name = 'ORCL_DATA1');
```

```
PATH
-----
```

```
ORCL:VOL1
ORCL:VOL2
```

28. 클러스터의 시작 / 종료

이것으로 Oracle RAC 10g의 설치 및 설정 작업을 모두 완료하였습니다.

이 시점에서 여러분은 이렇게 질문할 지도 모릅니다. “다 좋은데, 서비스를 어떻게 시작하고 종료하나요?” 이 가이드의 설명대로 작업을 완료하였다면 모든 서비스(Clusterware, 오라클 인스턴스, Enterprise Manager Database Console)는 Linux 노드가 리부트 될 때 자동으로 시작됩니다.

하지만 수동으로 노드를 셧다운하고 다시 시작해야 할 경우도 있습니다. 또는 Enterprise Manager가 실행되고 있지 않음을 확인하고, 이를 시작해야 할 수 있습니다. 이 섹션에서는 SRVCTL 커맨드를 사용하여 클러스터 환경을 시작하고 종료하는 방법을 설명합니다.

먼저 oracle UNIX 계정으로 로그인하였는지 확인합니다. 이 섹션에서 설명되는 모든 커맨드는 linux1에서 실행되는 것으로 가정합니다:

```
# su - oracle
$ hostname
linux1
```

Oracle RAC 10g 환경의 종료

가장 먼저 오라클 인스턴스를 종료합니다. 인스턴스와 관련 서비스가 종료된 이후에 ASM 인스턴스를 종료합니다. 마지막으로 노드 애플리케이션(Virtual IP, GSD, TNS Listener, ONS)을 종료합니다.

```
$ export ORACLE_SID=orcl1
$ emctl stop dbconsole
$ srvctl stop instance -d orcl -i orcl1
$ srvctl stop asm -n linux1
$ srvctl stop nodeapps -n linux1
```

Oracle RAC 10g 환경의 시작

먼저 노드 애플리케이션(Virtual IP, GSD, TNS Listener, ONS)을 시작합니다. 노드 애플리케이션이 성공적으로 실행되었다면, ASM 인스턴스를 시작합니다. 마지막으로, 오라클 인스턴스와 관련 서비스, Enterprise Manager Database 콘솔을 시작합니다.

```
$ export ORACLE_SID=orcl1
$ srvctl start nodeapps -n linux1
$ srvctl start asm -n linux1
$ srvctl start instance -d orcl -i orcl1
$ emctl start dbconsole
```

SRVCTL을 이용한 전체 인스턴스의 시작/종료

모든 인스턴스와 서비스를 동시에 시작/종료하는 방법이 아래와 같습니다.

```
$ srvctl start database -d orcl
$ srvctl stop database -d orcl
```

29. Transparent Application Failover (TAF)

기업이 엔터프라이즈 애플리케이션에 99.99% (또는 99.999%)의 가용성 수준을 요구하는 것은 드문 일이 아닙니다. 1 년 동안 5 시간 이내의 다운타임을 보장하려면 얼마나 많은 노력이 필요할 것인지 생각해 보십시오. 고가용성 요구사항을 만족하기 위해서는, 특정 시스템에 장애가 발생하는 경우 자동으로 페일오버를 수행하기 위한 메커니즘이 기본적으로 필요합니다. Oracle RAC 10g는 오라클 데이터베이스에 대해 매우 뛰어난 수준의 페일오버 메커니즘을 제공하는 솔루션입니다. Oracle RAC 10g는 클러스터 환경에서 지속적인 가용성을 제공하는데 필요한 모든 컴포넌트를 제공하고 있습니다. 클러스터 내의 특정 시스템에 장애가 발생하면, 사용자의 연결은 자동으로 다른 시스템에 자동으로 페일오버 됩니다.

페일오버 프로세싱에 관련한 Oracle RAC 10g의 핵심 컴포넌트로 Transparent Application Failover (TAF) 옵션을 들 수 있습니다. 장애가 발생한 시스템의 모든 데이터베이스 연결 및 프로세스는 클러스터 내의 다른 노드를 통해 재연결되며, 이 과정은 사용자 관점에서 볼 때 완전히 투명한 형태로 진행됩니다.

이번 섹션에서는 TAF가 Oracle RAC 10g에서 동작하는 원리를 간단하게 예시하고 있습니다. Oracle RAC 10g의 페일오버를 제대로 설명하려면 별도의 문서를 따로 작성해야 할 것입니다. 여기에서는 개론적인 지식만을 제공하는 것을 목적으로 합니다.

TAF는 OCI 라이브러리를 통해 자동적으로 수행됩니다. 따라서 TAF를 사용하기 위해 애플리케이션 (클라이언트) 코드를 변경할 필요는 없습니다. 하지만 Oracle TNS 파일(tnsnames.ora) 파일에 몇 가지 설정 작업이 필요합니다. (Java 실행 클라이언트의 경우 아직까지는 tnsnames.ora에 접근하기 위한 기능이 제공되지 않기 때문에 TAF에 참여할 수 없습니다.)

tnsnames.ora 파일의 설정

TAF의 실행을 테스트하기 전에, "non-RAC" 클라이언트 머신의 tnsnames.ora 파일 설정을 확인할 필요가 있습니다. (주변에 굴러다니는 Windows 머신을 활용하면 됩니다). 이 클라이언트에는 Oracle RDBMS 소프트웨어가 설치되어 있어야 합니다. (Oracle 소프트웨어의 "client install"만으로도 충분합니다.)

앞에서 클러스터 데이터베이스를 생성하는 과정에서, TAF 테스트를 위해 ORCLTEST라는 이름의 서비스를 생성한 바 있습니다. ORCLTEST는 로드 밸런싱 및 페일오버에 필요한 모든 구성 매개변수를 제공합니다. 새로 구성된 오라클 클러스터 데이터베이스에 연결하기 위해, 이 콘텐츠를 클라이언트 머신의 %ORACLE_HOME%\network\admin\tnsnames.ora 파일에 복사합니다.

```
...
ORCLTEST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux1-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = linux2-vip)(PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcltest.idevelopment.info)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = BASIC)
        (RETRIES = 180)
        (DELAY = 5)
      )
    )
  )
...

```

세션의 페일오버 정보를 확인하기 위한 SQL 쿼리

세션의 failover type 및 method, 페일오버의 발생 여부 등을 확인하려면 아래 쿼리 구문을 사용합니다.

```
COLUMN instance_name    FORMAT a13
COLUMN host_name        FORMAT a9
COLUMN failover_method  FORMAT a15

```

```

COLUMN failed_over          FORMAT a11

SELECT
  instance_name
  , host_name
  , NULL AS failover_type
  , NULL AS failover_method
  , NULL AS failed_over
FROM v$instance
UNION
SELECT
  NULL
  , NULL
  , failover_type
  , failover_method
  , failed_over
FROM v$session
WHERE username = 'SYSTEM';

```

TAF 데모

Window 머신 (또는 다른 플랫폼의 “non-RAC” 클라이언트 머신)에서 orcltest 서비스를 사용하여 SYSTEM 사용자로 클러스터 데이터베이스에 로그인합니다:

```
C:\W> sqlplus system/manager@orcltest
```

```

COLUMN instance_name      FORMAT a13
COLUMN host_name          FORMAT a9
COLUMN failover_method    FORMAT a15
COLUMN failed_over        FORMAT a11

```

```

SELECT
  instance_name
  , host_name
  , NULL AS failover_type
  , NULL AS failover_method
  , NULL AS failed_over
FROM v$instance
UNION
SELECT
  NULL
  , NULL
  , failover_type
  , failover_method
  , failed_over
FROM v$session
WHERE username = 'SYSTEM';

```

```

INSTANCE_NAME HOST_NAME FAILOVER_TYPE FAILOVER_METHOD FAILED_OVER
-----
orcl1         linux1
              SELECT          BASIC          NO

```

위에서 설정한 SQL*Plus 세션에서 로그아웃 하지 마십시오!

위 쿼리를 수행한 다음, abort 옵션을 사용하여 linux1 노드의 orcl1 인스턴스를 셧다운 합니다. 이 작업을 수행하기 위해 아래와 같이 srvctl 커맨드라인 유틸리티를 사용합니다:

```

# su - oracle
$ srvctl status database -d orcl
Instance orcl1 is running on node linux1
Instance orcl2 is running on node linux2

```

```
$ srvctl stop instance -d orcl -i orcl1 -o abort
```

```
$ srvctl status database -d orcl
Instance orcl1 is not running on node linux1
Instance orcl2 is running on node linux2
```

다시 앞의 SQL 세션으로 돌아가, 버퍼에 저장된 SQL 구문을 재실행합니다:

```
COLUMN instance_name    FORMAT a13
COLUMN host_name        FORMAT a9
COLUMN failover_method  FORMAT a15
COLUMN failed_over      FORMAT a11
```

```
SELECT
  instance_name
, host_name
, NULL AS failover_type
, NULL AS failover_method
, NULL AS failed_over
FROM v$instance
UNION
SELECT
  NULL
, NULL
, failover_type
, failover_method
, failed_over
FROM v$session
WHERE username = 'SYSTEM';
```

```
INSTANCE_NAME HOST_NAME FAILOVER_TYPE FAILOVER_METHOD FAILED_OVER
-----
orcl2          linux2
                SELECT          BASIC          YES
```

```
SQL> exit
```

위 실행 결과에서, 세션이 linux2 노드의 orcl2 인스턴스로 페일오버 되었음을 확인할 수 있습니다.

30. 트러블슈팅

OCFS2 - O2CB가 부팅 시 자동으로 시작되도록 설정

OCFS2 1.2.1 이전 버전에는 'on-boot' 속성을 설정해 주었음에도 불구하고 부팅 시 드라이버가 로드되지 않는 버그가 존재합니다. OCFS2 공식제품문서의 지시 사항 대로 'on-boot' 속성을 설정한 후, 부팅 과정에서 다음과 같은 에러가 표시됨을 확인할 수 있을 것입니다:

```
...
Mounting other filesystems:
  mount.ocfs2: Unable to access cluster service

Cannot initialize cluster mount.ocfs2:
  Unable to access cluster service Cannot initialize cluster [FAILED]
...
```

Red Hat 운영체제의 버전이 chkconfig-1.3.11.2-1에서 chkconfig-1.3.13.2-1로 업그레이드

레이드되면서 서비스 등록 방법이 변경되었습니다. O2CB 스크립트는 chkconfig-1.3.11.2-1과 호환합니다.
on-boot 속성을 설정하기 전에,
/etc/init.d/o2cb 파일에서 아래 라인을 제거합니다.

```
### BEGIN INIT INFO
# Provides: o2cb
# Required-Start:
# Should-Start:
# Required-Stop:
# Default-Start: 2 3 5
# Default-Stop:
# Description: Load O2CB cluster services at system boot.
### END INIT INFO
```

그리고 o2cb 서비스를 다시 등록합니다.

```
# chkconfig --del o2cb
# chkconfig --add o2cb
# chkconfig --list o2cb
o2cb          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

```
# ll /etc/rc3.d/*o2cb*
lrwxrwxrwx  1 root root 14 Sep 29 11:56 /etc/rc3.d/S24o2cb -> ../init.d/o2cb
```

S24o2cb 서비스가 디폴트 런레벨에서 실행 중이어야 합니다.
버그를 해결하고 난 다음에는, 이제 아래와 같이 on-boot 속성을 설정할 수 있습니다.

```
# /etc/init.d/o2cb offline ocfs2
# /etc/init.d/o2cb unload
# /etc/init.d/o2cb configure
Configuring the O2CB driver.
```

This will configure the on-boot properties of the O2CB driver.
The following questions will determine whether the driver is loaded on boot. The current values will be shown in brackets ('[]'). Hitting
without typing an answer will keep that current value. Ctrl-C
will abort.

```
Load O2CB driver on boot (y/n) [n]: y
Cluster to start on boot (Enter "none" to clear) [ocfs2]: ocfs2
Writing O2CB configuration: OK
Loading module "configs": OK
Mounting configs filesystem at /config: OK
Loading module "ocfs2_nodemanager": OK
Loading module "ocfs2_dlm": OK
Loading module "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Starting cluster ocfs2: OK
```

OCFS2 - O2CB Heartbeat Threshold의 조정

본 문서의 이전 버전(iSCSI가 아닌 FireWire를 공유 스토리지로 사용한 버전)을 작성하면서, 필자는 OCFS2의 설치/구성, 새로운 볼륨의 포매팅, Oracle Clusterware의 설치와 같은 작업을 성공적으로 수행할 수 있었습니다. Oracle Clusterware를 설치하고 FireWire를 통해 공유 드라이브 연결하는 것은 성공적이었지만, Clusterware 소프트웨어가 양쪽 노드에서 실행된 지 약 15 분 경과 후 많은 락업(lock-up)과 행(hang) 현상이 발생하기 시작했습니다. 행(hang) 현상이 발생하는 노드(linux1, linux2)는 경우에 따라 달랐습니다. 또 I/O 부하가 심하지 않은(또는 전혀 없는) 상황에서도 크래시 또는 행 현상이 발생했습

니다.

OCFS2의 트레이스 파일을 점검한 뒤, Voting Disk의 속도가 너무 늦어 O2CB heartbeat 임계치보다 낮은 성능을 보이며, 이로 인해 Oracle Clusterware 소프트웨어가 크래시 현상을 발생시키는 것으로 확인되었습니다.

필자는 이 문제를 해결하기 위해 OC2B heartbeat threshold를 7(디폴트)에서 61로 변경하였습니다. 환경에 따라서는 더 높은 수치가 필요할 수도 있을 것입니다. 이 임계치 설정은 노드가 자신을 "차단(fence)"하는데 필요한 시간을 계산하기 위해 사용됩니다.

O2CB heartbeat threshold의 설정 방법은 "O2CB Heartbeat Threshold의 조정" 섹션에서 자세히 설명되고 있습니다.

Oracle Clusterware (10g RAC 10.1.0.3 및 이후 버전) – CSS 타임아웃 시간의 계산

이 섹션은 Oracle10g R1(10.1.0.3 이상)과 Oracle10g R2에 해당됩니다. 필자의 경우 본 구성 예제와 관련하여, 아래에 설명된 작업을 수행할 필요가 없었다는 점을 참고하시기 바랍니다. 아래에 설명된 문제가 RAC 시스템에서 발생할 가능성은 낮습니다. RAC 10g 환경에서의 CSS 타임아웃 값의 계산에 대한 자세한 정보는 Metalink Note294430.1을 참고하시기 바랍니다.

환경에 따라, Oracle Clusterware를 설치한 뒤 Clusterware의 CSS 타임아웃 값을 변경해야 할 수도 있습니다. 하지만 Oracle10g R2 및 이후 버전에서는 이 작업이 불필요합니다. Oracle10g R2 사용자는 패치 4896338을 적용하고 디폴트 misscount 값을 수정하지 말아야 합니다. 10.1.0.3 및 이후 버전의 CSS 타임아웃 계산 방식은 10.1.0.2와 다릅니다. Linux 플랫폼(IA32, IA64, x86-64)에 설치된 Oracle 10.1.0.3 및 이후 버전에서 CSS 데몬 타임아웃과 관련한 여러 가지 문제가 보고되었습니다. 이 이슈는 DBCA를 이용한 데이터베이스 생성, FireWire를 이용한 공유 스토리지 구성 과정에서 특히 큰 문제가 되었습니다. 예를 들어, 데이터베이스 생성 과정에서 데이터베이스 생성 작업이 다음과 같은 에러와 함께 실패하는 경우가 많았습니다: ORA-03113: end-of-file on communication channel. \$ORA_CRS_HOME/css/log/ocssd1.log 로그 파일에는 다음과 같은 내용이 기록됩니다:

clssnmDiskPingMonitorThread: voting device access hanging (45010 miliseconds)

이 문제의 근본적인 원인은 디스크의 낮은 성능과 CSS misscount의 디폴트 값에 있습니다. CSS misscount 값은 CSS가 노드를 탈퇴(evict) 처리하기 전에 무시(miss)되는 하트 비트 시그널의 수를 의미합니다. CSS는 이 숫자를 기준으로 Voting Disk에 대한 I/O 작업이 *타임아웃(timed out)* 되었음을 판단하고 스플릿 브레인(split brain) 현상을 방지하기 위해 자신을 종료합니다. Linux 기반 Oracle 10.1.0.2 및 이후 버전의 경우 CSS misscount의 디폴트 값은 60으로 설정되어 있습니다. 하지만 타임아웃 값을 계산하기 위한 공식은 10.1.0.2에서 10.1.0.3으로 업그레이드되면서 변경되었습니다.

10.1.0.2의 타임아웃 계산 방법이 아래와 같습니다:

```
time_in_secs > CSS misscount, then EXIT
```

예를 들어 디폴트 값인 60이 사용되는 경우 타임아웃 기간은 60 초로 설정됩니다. 10.1.0.3에서는 공식이 아래와 같이 변경 되었습니다:

```
disktimeout_in_secs = MAX((3 * CSS misscount)/4, CSS misscount - 15)
```

여기에 디폴트 CSS misscount 값인 60을 적용하면, 45 초가 결과로 계산됩니다. 이 변경 사항은 노드 장애 시 보다 신속한 클러스터 재구성 작업이 가능하도록 지원하기 위해 적용되었습니다. 10.1.0.2 버전에서 디폴트 CSS misscount 값(60)을 적용하는 경우에는 타임아웃이 발생하기까지 60초를 기다려야 하는 반면, 10.1.0.3 버전에서는 45 초로, 15 초가 단축되었습니다.

그럼 여기서 CSS misscount에 대해 설명하고 있는 이유가 무엇일까요? 앞에서 설명한 것처럼, Oracle Clusterware의 크래시 현상(또는 시스템의 높은 I/O 부하) 때문에 데이터베이스 생성 작업이 실패하는 경우가 종종 있습니다. 높은 I/O 부하가 발생하면 Voting Disk에 쿼리를 수행하는 과정에서 CSS에 장시간의 타임아웃이 발생합니다. 지정된 타임아웃 시간을 초과하면 Oracle Clusterware에 크래시(crash) 현상이 발생합니다. 본 구성

예제에서 사용되는 네트워크와 외장형 드라이브의 성능이 그리 뛰어나지 않기 때문에 이런 문제가 수시로 발생할 수 있습니다. 공유 드라이브에 대한 액세스가 느리면 느릴 수록 문제가 발생할 가능성은 높아집니다.

다행스러운 사실은, CSS `misscount`의 값을 수정하여 타임아웃 시간을 증가시킬 수 있다는 것입니다. 속도가 느린 드라이브가 사용된 경우, 필자는 CSS `misscount`를 360으로 설정하여 문제를 방지할 수 있었습니다. 분명히 확인할 수는 없었지만 CSS `misscount`를 600으로 설정해도 아무런 문제가 없는 것처럼 보였습니다.

단, 오라클이 `misscount` 값을 수정하는 것을 권장하지 않는다는 사실을 참고하시기 바랍니다. 오라클은 CSS 동작 방식을 변경한 최신 패치 셋을 적용하는 것을 권장하고 있습니다.

또 Oracle10g R2에서는 `misscount`를 증가시킬 필요가 없다는 점을 명심할 필요가 있습니다. 10.2.0.1 + Bug 4896338 버전부터, CSS는 (최초 클러스터 구성 시, 또는 재구성 작업 이전 상황을 제외하고는) Voting Disk에 대한 I/O 때문에 노드를 클러스터에서 탈퇴시키지 않도록 수정되었습니다. 따라서 클러스터에 N 개의 노드가 존재하고 그 중 하나가 Voting Disk 접근 과정에서 `misscount`에 설정된 것보다 많은 시간을 사용하는 경우, Voting Disk에 대한 액세스가 `disktimeout` 설정값을 초과하지 않는 이상 노드는 탈퇴 (evict) 처리되지 않습니다. 따라서 이 패치를 적용했다면 `misscount`를 증가시킬 아무런 이유가 없습니다.

그렇다면 10g R1 사용자들은 CSS `misscount`의 디폴트 값을 어떻게 수정할 수 있을까요? 여러 가지 방법이 있습니다. 가장 쉬운 방법은 Oracle Clusterware를 설치하고 나서 `root.sh`를 클러스터의 모든 노드에서 실행하기 전에 `$ORA_CRS_HOME/install/rootconfig` 스크립트를 수정하는 것입니다. 예를 들어, CSS `misscount`를 60에서 360으로 변경하려면 Oracle RAC 클러스터의 모든 노드에 대해 `$ORA_CRS_HOME/install/rootconfig` 파일을 수정해 주어야 합니다. 356번 라인의 아래 항목을 변경해 줍니다:

```
CLSCFG_MISCNT="-misscount 60"
```

위 항목을 다음과 같이 변경합니다:

```
CLSCFG_MISCNT="-misscount 360"
```

Oracle Clusterware가 이미 설치되어 있다면 `$ORA_CRS_HOME/bin/crsctl` 커맨드를 이용하여 CSS `misscount`를 변경해 줄 수 있습니다. 먼저 CSS `misscount`의 현재값을 조회합니다:

```
# $ORA_CRS_HOME/bin/crsctl get css misscount
360
```

설정된 값이 60임을 확인했다면, 아래와 같이 360으로 수정해 줍니다:

클러스터의 노드 중 하나만을 실행합니다. 예를 들어 `linux2`를 섀다운 처리하고 `linux1`만을 실행합니다.

실행 중인 노드(`linux1`)에서 `root` 계정으로 로그인하고 아래와 같이 실행합니다:

```
$ORA_CRS_HOME/bin/crsctl set css misscount 360
```

노드(`linux1`)를 리부트합니다.

마지막으로 클러스터의 다른 모든 노드를 시작합니다.

31. 결론

Oracle10g RAC는 데이터베이스 환경에 뛰어난 폴트 톨러런스 및 로드 밸런싱 기능을 제공합니다. 하지만 Oracle 10g RAC의 기능을 체험해 보고자 하는 DBA들에게 클러스터의 구성 비용은 큰 부담이 될 수 있습니다. 아무리 작은 규모의 RAC 클러스터를 구성하더라도 15,000 달러에서 20,000 달러의 비용이 듭니다.

본 문서를 통해 CentOS 4.2(또는 Red Hat Enterprise Linux 4)와 iSCSI 테크놀로지를 이용하여 저렴

한 비용으로 Oracle RAC 10g Release 2 클러스터를 구축하는 방법을 설명하였습니다. 여기에서 설명된 RAC 솔루션을 구축하는 데에는 약 2,200 달러의 비용이 필요하며, 개발용 Oracle RAC 10g Release 2 클러스터로 충분히 활용 가능합니다. 단 본 문서에서 사용된 하드웨어 구성을 절대 운영 환경에 적용해서는 안됩니다.

32. 도와주신 분들

이처럼 방대하고 복잡한 문서가 한 사람의 힘으로 만들어지는 경우는 거의 없습니다. 이 문서의 경우 필자가 주도적으로 참여하여 내용을 작성하고 구성을 검증하였지만, 문서 작업을 성공적으로 완료하기까지 도와주신 여러분들께 감사의 마음을 표하는 것이 도리일 듯 합니다.

먼저, 오라클 Server BDE 팀의 Bane Radulovic에게 감사의 뜻을 표하고자 합니다. Bane은 필자에게 Openfiler를 소개해 주었을 뿐 아니라, 이 제품을 Oracle 10g RAC에서 활용하는 방법에 대한 그의 지식과 경험을 나누어 주었습니다. 그의 연구와 노력 덕분에 Openfiler의 설정 작업을 문제 없이 마칠 수 있었습니다. 그는 또 하드웨어 선정 및 테스트 과정에도 참여하였습니다.

다음으로, Werner Puschitz가 작성한 문서("Installing Oracle Database 10g with Real Application Clusters (RAC) on Red Hat Enterprise Linux Advanced Server 3")에서 많은 도움을 얻었음을 밝힙니다. 이 문서와, 그가 작성한 기타 여러 문서를 통해 다른 오라클 제품 문서에서 확인할 수 없었던 Oracle RAC 10g 정보를 얻을 수 있었습니다. hangcheck-timer 커널 모듈의 설치, Unix shared memory의 설정, ASMLib의 설정 등에 관련한 그의 연구와 노력이 없었다면 이 가이드도 완성되지 못했을 것입니다. Werner Puschitz가 집필한 Linux 및 오라클 관련 테크니컬 아티클을 참고하시려면 그의 웹 사이트(www.puschitz.com)를 방문하시기 바랍니다.

또 Oracle Cluster Verification Utility(CVU)의 사용 방법에 대한 조언을 제공해 준 Tzvika Lemel에게도 감사하다는 말씀을 전합니다.

마지막으로 본 문서에서 사용된 하드웨어를 너그럽게 지원해 준 하드웨어 벤더 업체(Maxtor, D-Link, SIIG, LaCie)에 감사의 뜻을 표합니다.

Jeffrey Hunter [www.idevelopment.info] 는 12년이 넘는 경력을 보유한 선임 DBA이자 소프트웨어 엔지니어로, 터록의 스타니슬로스 주립 대학에서 컴퓨터 과학 학사 학위를 취득하였습니다. Jeff는 Oracle Certified Professional, Java Development Certified Professional 자격증을 보유하고 있으며, 현재 DBA Zone, Inc.의 집필진으로 활동 중입니다. Jeff는 Unix, Linux, Windows NT 환경의 고급 성능 튜닝, Java 프로그래밍, 용량 계획, 데이터베이스 보안, 물리적/논리적 데이터베이스 설계 등을 전문 분야로 하고 있습니다. Jeff는 그 밖에도 수학 암호화 이론, Java/C 프로그래밍 언어 처리기(컴파일러 및 인터프리터), LDAP, 웹 기반 데이터베이스 관리 툴의 개발, Linux 등에 많은 관심을 갖고 있습니다.

여러분의 의견을 보내 주십시오