

Oracle® VM Server for SPARC 3.4 관리 설 명서

ORACLE®

부품 번호: E71810
2016년 8월

부품 번호: E71810

Copyright © 2007, 2016, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 합의서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 합의서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. 사용자와 오라클 간의 합의서에 별도로 규정되어 있지 않는 한 Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다. 단, 사용자와 오라클 간의 합의서에 규정되어 있는 경우는 예외입니다.

설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=d0cacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

오라클 고객센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

목차

이 설명서 사용	17
1 Oracle VM Server for SPARC 소프트웨어 개요	19
Oracle VM Server for SPARC 및 Oracle Solaris OS 버전 정보	19
하이퍼바이저 및 Logical Domains	20
Logical Domains Manager	22
도메인의 역할	22
명령줄 인터페이스	23
가상 입/출력	23
리소스 구성	25
지속적 구성	25
Oracle VM Server for SPARC Management Information Base	25
Oracle VM Server for SPARC 문제 해결	25
2 Oracle VM Server for SPARC 보안	27
권한을 사용하여 논리적 도메인의 관리 위임	27
권한 프로파일 및 역할 사용	28
Logical Domains Manager 프로파일 콘텐츠	30
확인된 부트 사용	31
3 서비스 및 컨트롤 도메인 설정	33
출력 메시지	33
기본 서비스 만들기	34
▼ 기본 서비스를 만드는 방법	34
컨트롤 도메인의 초기 구성	35
컨트롤 도메인 구성	35
컨트롤 도메인의 초기 factory-default 구성에서 CPU 및 메모리 리소스 줄이기	36
도메인을 사용하도록 재부트	37

▼ 재부트하는 방법	37
Oracle Solaris 10 서비스 도메인과 다른 도메인 사이의 네트워킹 사용으로 설정	38
▼ 가상 스위치를 주 인터페이스로 구성하는 방법	38
가상 네트워크 터미널 서버 데몬 사용으로 설정	39
▼ 가상 네트워크 터미널 서버 데몬을 사용으로 설정하는 방법	39
ILOM 상호 연결이 사용으로 설정되었는지 확인	39
▼ ILOM 상호 연결 구성을 확인하는 방법	40
▼ ILOM 상호 연결 서비스를 다시 사용으로 설정하는 방법	40
4 게스트 도메인 설정	43
게스트 도메인 만들기 및 시작	43
▼ 게스트 도메인을 만들고 시작하는 방법	43
게스트 도메인에 Oracle Solaris OS 설치	46
메모리 크기 요구사항	46
▼ DVD에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법	47
▼ Oracle Solaris ISO 파일에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법	48
▼ Oracle Solaris 10 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법	50
5 도메인 콘솔 사용	53
권한을 사용하여 도메인 콘솔에 대한 액세스 제어	53
▼ 역할을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법	54
▼ 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법	55
▼ 역할을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법	56
▼ 권한 프로파일을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법	58
도메인 콘솔 로깅 사용	58
▼ 콘솔 로깅을 사용 또는 사용 안함으로 설정하는 방법	59
도메인 콘솔 로깅에 대한 서비스 도메인 요구사항	59
네트워크를 통해 게스트 콘솔에 연결	59
콘솔 그룹 사용	60
▼ 하나의 그룹에 다중 콘솔을 결합하는 방법	60
6 I/O 도메인 구성	63
I/O 도메인 개요	63
일반적인 I/O 도메인 만들기 지침	64

7 PCIe 버스를 지정하여 루트 도메인 만들기	65
PCIe 버스를 지정하여 루트 도메인 만들기	65
정적 PCIe 버스 지정	66
동적 PCIe 버스 지정	67
▼ PCIe 버스를 지정하여 루트 도메인을 만드는 방법	68
8 PCIe SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기	73
SR-IOV 개요	73
SR-IOV 하드웨어 및 소프트웨어 요구사항	76
현재 SR-IOV 기능 제한 사항	79
정적 SR-IOV	80
정적 SR-IOV 소프트웨어 요구사항	80
동적 SR-IOV	81
동적 SR-IOV 소프트웨어 요구사항	81
동적 SR-IOV 구성 요구사항	81
I/O 가상화 사용으로 설정	83
▼ PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법	83
PCIe SR-IOV 가상 기능 사용 계획	84
이더넷 SR-IOV 가상 기능 사용	85
이더넷 SR-IOV 하드웨어 요구사항	85
이더넷 SR-IOV 제한 사항	85
이더넷 SR-IOV 가상 기능 사용 계획	86
이더넷 장치 관련 등록 정보 및 네트워크 관련 등록 정보	86
이더넷 가상 기능 만들기	86
이더넷 가상 기능 삭제	90
이더넷 SR-IOV 가상 기능 수정	93
I/O 도메인에서 이더넷 SR-IOV 가상 기능 추가 및 제거	95
고급 SR-IOV 항목: 이더넷 SR-IOV	97
SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기	100
InfiniBand SR-IOV 가상 기능 사용	103
InfiniBand SR-IOV 하드웨어 요구사항	103
InfiniBand 가상 기능 만들기 및 삭제	103
I/O 도메인에서 InfiniBand 가상 기능 추가 및 제거	107
InfiniBand 가상 기능을 루트 도메인에 추가 및 루트 도메인에서 제거	110
고급 SR-IOV 항목: InfiniBand SR-IOV	111
광 섬유 채널 SR-IOV 가상 기능 사용	115
광 섬유 채널 SR-IOV 하드웨어 요구사항	115
광 섬유 채널 SR-IOV 요구사항 및 제한 사항	116

광 섬유 채널 장치 클래스 관련 등록 정보	116
광 섬유 채널 SR-IOV 가상 기능 만들기	118
광 섬유 채널 SR-IOV 가상 기능 삭제	122
광 섬유 채널 SR-IOV 가상 기능 수정	124
I/O 도메인에서 광 섬유 채널 SR-IOV 가상 기능 추가 및 제거	125
고급 SR-IOV 항목: 광 섬유 채널 SR-IOV	127
I/O 도메인 복원성	128
복원형 I/O 도메인 요구사항	129
I/O 도메인 복원성 제한 사항	129
복원형 I/O 도메인 구성	130
예제 - 복원형 및 비복원형 구성 사용	133
비복원형 I/O 도메인이 구성된 상태로 루트 도메인 재부트	134
9 직접 I/O를 사용하여 I/O 도메인 만들기	135
PCIe 끝점 장치를 지정하여 I/O 도메인 만들기	135
직접 I/O 하드웨어 및 소프트웨어 요구사항	137
현재 직접 I/O 기능 제한 사항	138
PCIe 끝점 장치 구성 계획	139
PCIe 끝점이 구성된 상태로 루트 도메인 재부트	140
PCIe 하드웨어 변경	141
PCIe 카드를 제거할 때 게스트 도메인 가동 중단 최소화	142
PCIe 끝점 장치를 지정하여 I/O 도메인 만들기	144
▼ PCIe 끝점 장치를 지정하여 I/O 도메인을 만드는 방법	144
직접 I/O 문제	148
ldm list-io -l 출력에서 PCI Express 이중 10Gbit 이더넷 파이버 카드에 4개 의 하위 장치가 표시됨	148
10 비primary 루트 도메인 사용	151
비primary 루트 도메인 개요	151
비primary 루트 도메인 요구사항	152
비primary 루트 도메인 제한 사항	153
비primary 루트 도메인 예	154
PCIe 버스에 대한 I/O 가상화 사용	154
비primary 루트 도메인에서 직접 I/O 장치 관리	156
비primary 루트 도메인에서 SR-IOV 가상 기능 관리	157
11 가상 디스크 사용	161
가상 디스크 소개	161

가상 디스크 식별자 및 장치 이름	163
가상 디스크 관리	164
▼ 가상 디스크를 추가하는 방법	165
▼ 가상 디스크 백엔드를 여러 번 내보내는 방법	165
▼ 가상 디스크 옵션을 변경하는 방법	166
▼ 시간 초과 옵션을 변경하는 방법	166
▼ 가상 디스크를 제거하는 방법	166
가상 디스크 표시	166
전체 디스크	167
단일 슬라이스 디스크	167
가상 디스크 백엔드 옵션	168
읽기 전용(ro) 옵션	168
배타적(excl) 옵션	168
슬라이스(slice) 옵션	169
가상 디스크 백엔드	169
물리적 디스크 또는 디스크 LUN	170
▼ 물리적 디스크를 가상 디스크로 내보내는 방법	170
물리적 디스크 슬라이스	171
▼ 물리적 디스크 슬라이스를 가상 디스크로 내보내는 방법	171
▼ 슬라이스 2를 내보내는 방법	172
파일 및 볼륨 내보내기	172
가상 디스크 다중 경로 구성	176
가상 디스크 다중 경로 및 NFS	178
가상 디스크 다중 경로 및 가상 디스크 시간 초과	178
▼ 가상 디스크 다중 경로를 구성하는 방법	178
동적 경로 선택	180
CD, DVD 및 ISO 이미지	181
▼ CD 또는 DVD를 서비스 도메인에서 게스트 도메인으로 내보내는 방법 ...	182
▼ 컨트롤 도메인에서 ISO 이미지를 내보내서 게스트 도메인을 설치하는 방 법	183
가상 디스크 시간 초과	184
가상 디스크 및 SCSI	185
가상 디스크 및 format 명령	185
가상 디스크에서 ZFS 사용	186
서비스 도메인에서 ZFS 풀 구성	186
ZFS를 사용하여 디스크 이미지 저장	186
디스크 이미지의 스냅샷 만들기	187
복제본을 사용하여 새 도메인 프로비전	188
Oracle VM Server for SPARC 환경에서 볼륨 관리자 사용	190

Volume Manager에서 가상 디스크 사용	190
가상 디스크에서 Volume Manager 사용	192
가상 디스크 문제	193
특정 조건에서 게스트 도메인의 Solaris Volume Manager 구성 또는 메타 장치가 손실될 수 있음	193
Oracle Solaris 부트 디스크 호환성	194
12 가상 SCSI 호스트 버스 어댑터 사용	197
가상 SCSI 호스트 버스 어댑터 소개	197
가상 SCSI HBA의 작동 모델	199
가상 SCSI HBA 식별자 및 장치 이름	200
가상 SCSI HBA 관리	201
물리적 SCSI HBA 정보 얻기	201
가상 저장 영역 네트워크 만들기	202
가상 SCSI 호스트 버스 어댑터 만들기	202
가상 SCSI HBA의 존재 여부 확인	203
가상 SCSI HBA 시간 초과 옵션 설정	203
가상 SCSI 호스트 버스 어댑터 제거	203
가상 저장 영역 네트워크 제거	204
LUN 추가 또는 제거	204
게스트 도메인에서 가상 LUN의 모양	204
가상 SCSI HBA 및 가상 SAN 구성	205
가상 SCSI HBA 다중 경로 구성	205
▼ 가상 SCSI HBA 다중 경로를 구성하는 방법	208
▼ 게스트 도메인에서 가상 SCSI HBA에 대한 다중 경로를 사용으로 설정하는 방법	209
▼ 게스트 도메인에서 가상 SCSI HBA에 대해 다중 경로를 사용 안함으로 설정하는 방법	209
▼ 서비스 도메인에서 가상 SCSI HBA에 대한 다중 경로를 사용으로 설정하는 방법	210
▼ 서비스 도메인에서 가상 SCSI HBA에 대한 다중 경로를 사용 안함으로 설정하는 방법	211
SCSI 장치에서 부트	212
가상 LUN에서 부트	212
SCSI DVD 장치에서 부트	213
가상 LUN 설치	213
가상 SCSI HBA 시간 초과	213
가상 SCSI HBA 및 SCSI	214
게스트 도메인에서 단편화가 심한 I/O 버퍼 지원	214

13 가상 네트워크 사용	217
가상 네트워크에 대한 소개	218
Oracle Solaris 11 네트워킹 개요	218
Oracle Solaris 10 네트워킹 개요	221
가상 네트워크 성능 최대화	222
하드웨어 및 소프트웨어 요구사항	222
가상 네트워크의 성능 최대화를 위한 도메인 구성	223
가상 스위치	223
가상 네트워크 장치	225
Vnet 간 LDC 채널	225
논리적 도메인에 있는 네트워크 확인	228
네트워크 장치 구성 및 통계 보기	228
가상 네트워크 장치에서 소비되는 물리적 네트워크 대역폭의 양 제어	231
네트워크 대역폭 제한	232
네트워크 대역폭 제한 설정	232
가상 장치 식별자 및 네트워크 인터페이스 이름	234
Oracle Solaris 11 네트워크 인터페이스 이름 찾기	235
자동 또는 수동 MAC 주소 지정	237
도메인에 지정된 MAC 주소의 범위	237
자동 지정 알고리즘	238
중복된 MAC 주소 감지	238
Oracle Solaris 10을 실행하는 도메인에서 네트워크 어댑터 사용	239
▼ GLDv3 준수 네트워크 어댑터인지 여부를 확인하는 방법	239
NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성	240
Oracle Solaris 11 시스템에서 NAT 구성	240
Oracle Solaris 10 시스템에서 NAT 구성	242
Oracle VM Server for SPARC 환경에서 IPMP 구성	243
Oracle Solaris 11 도메인에서 가상 네트워크 장치를 IPMP 그룹으로 구성 ...	243
Oracle Solaris 10 도메인에서 가상 네트워크 장치를 IPMP 그룹으로 구성 ...	245
서비스 도메인에서 IPMP 구성 및 사용	246
Oracle VM Server for SPARC 가상 네트워킹에서 링크 기반 IPMP 사용	248
VLAN 태그 지정 사용	252
포트 VLAN ID	252
VLAN ID	253
VLAN 지정 및 사용	253
▼ 설치 서버가 VLAN에 있을 때 게스트 도메인을 설치하는 방법	255
전용 VLAN 사용	256
PVLAN 요구사항	257

PVLAN 구성	258
패킷 처리량 성능 조정	261
가상 스위치에서 링크 통합 사용	262
점보 프레임 구성	264
▼ 점보 프레임을 사용하도록 가상 네트워크 및 가상 스위치 장치를 구성하는 방법	265
vnet 및 vsw 드라이버의 이전(점보 미인식) 버전과의 호환성(Oracle Solaris 10)	267
가상 네트워크에서 가상 NIC 사용	268
가상 네트워크 장치에 가상 NIC 구성	270
도메인에 Oracle Solaris 11 영역 만들기	270
신뢰할 수 있는 가상 네트워크 사용	271
신뢰할 수 있는 가상 네트워크 요구사항 및 제한 사항	271
신뢰할 수 있는 가상 네트워크 구성	272
신뢰할 수 있는 가상 네트워크 정보 보기	275
Oracle Solaris 11 네트워킹 관련 기능 차이점	277
14 도메인 마이그레이션	279
도메인 마이그레이션 소개	279
마이그레이션 작업 개요	280
소프트웨어 호환성	281
마이그레이션 작업 보안	281
마이그레이션용 SSL 인증서 구성	282
SSL 인증서 제거	283
도메인 마이그레이션용 FIPS 140-2 모드	283
▼ Logical Domains Manager를 FIPS 140-2 모드로 실행하는 방법	283
▼ Logical Domains Manager를 FIPS 140-2 모드에서 기본 모드로 되돌리 는 방법	285
도메인 마이그레이션 제한 사항	285
마이그레이션에 대한 버전 제한 사항	285
perf-counters 설정을 위한 마이그레이션 제한 사항	286
linkprop=phys-state 설정을 위한 마이그레이션 제한 사항	286
많은 수의 가상 장치가 있는 도메인에 대한 마이그레이션 제한 사항	287
도메인 마이그레이션	287
Dry Run 수행	287
비대화식 마이그레이션 수행	288
활성 도메인 마이그레이션	288
CPU에 대한 도메인 마이그레이션 요구사항	289
메모리에 대한 마이그레이션 요구사항	290

물리적 I/O 장치에 대한 마이그레이션 요구사항	291
가상 I/O 장치에 대한 마이그레이션 요구사항	291
PCIe 끝점 장치에 대한 마이그레이션 요구사항	292
PCIe SR-IOV 가상 기능에 대한 마이그레이션 요구사항	293
암호화 장치에 대한 마이그레이션 요구사항	293
활성 도메인의 지연된 재구성	293
활성 도메인에 전원 관리 탄력적 정책이 적용되는 동안 마이그레이션	293
다른 도메인에 대한 작업	294
OpenBoot PROM의 도메인 또는 커널 디버거에서 실행 중인 도메인 마이그레이션	294
바인드된 도메인 또는 비활성 도메인 마이그레이션	294
가상 I/O 장치에 대한 마이그레이션 요구사항	295
PCIe 끝점 장치에 대한 마이그레이션 요구사항	295
PCIe SR-IOV 가상 기능에 대한 마이그레이션 요구사항	295
진행 중인 마이그레이션 모니터	295
진행 중인 마이그레이션 취소	296
실패한 마이그레이션 복구	297
마이그레이션 예	297
15 리소스 관리	299
리소스 재구성	299
동적 재구성	300
지연된 재구성	300
리소스 할당	301
CPU 할당	302
▼ 전체 코어 제약 조건을 적용하는 방법	302
▼ 최대 코어 제약 조건을 적용하는 방법	303
전체 코어 제약 조건과 다른 도메인 기능 사이의 상호 작용	304
하드 분할을 사용하여 시스템 구성	305
도메인의 구성 확인	306
CPU 전체 코어로 도메인 구성	306
하드 분할된 시스템과 다른 Oracle VM Server for SPARC 기능과의 상호 작용	309
도메인에 물리적 리소스 지정	311
▼ physical-bindings 제약 조건을 제거하는 방법	313
▼ 물리적으로 바인드되지 않은 모든 리소스를 제거하는 방법	314
컨트롤 도메인에서 물리적 리소스 관리	314
도메인에서 물리적 리소스 관리 제한 사항	315

메모리 동적 재구성 사용	315
메모리 추가	316
메모리 제거	316
일부 메모리 DR 요청	316
컨트롤 도메인의 메모리 재구성	317
동적 및 지연된 재구성	317
메모리 정렬	318
메모리 DR 예	320
리소스 그룹 사용	322
리소스 그룹 요구사항 및 제한 사항	323
전원 관리 사용	323
동적 리소스 관리 사용	324
도메인 리소스 나열	326
시스템에서 읽을 수 있는 출력	327
플래그 정의	327
활용률 통계 정의	328
다양한 목록 보기	328
제약 조건 나열	331
리소스 그룹 정보 나열	331
Perf-Counter 등록 정보 사용	332
리소스 관리 문제	334
도메인에서 많은 수의 CPU를 제거할 경우 실패할 수 있음	334
때때로 동적으로 추가된 메모리 블록을 전체로서만 동적으로 제거할 수 있음	335
16 도메인 구성 관리	337
도메인 구성 관리	337
사용 가능한 구성 복구 방법	338
자동 저장을 사용하여 구성 복원	338
자동 복구 정책	339
도메인 구성 저장	340
도메인 구성 복원	341
서비스 프로세서 연결 문제 해결	343
구성 관리 문제	344
init-system이 저장된 XML 파일에서 게스트 도메인의 이름이 지정된 코어 제약 조건을 복원하지 않음	344
17 하드웨어 오류 처리	345

하드웨어 오류 처리 개요	345
FMA를 사용하여 결함이 있는 리소스 차단 또는 구성 해제	345
결함이 있거나 누락된 리소스 감지 후 도메인 복구	347
복구 모드 하드웨어 및 소프트웨어 요구사항	349
저하된 구성	349
복구 모드 제어	350
도메인을 저하됨으로 표시	350
I/O 리소스를 비워짐으로 표시	350
18 기타 관리 작업 수행	351
CLI에서 이름 입력	351
/etc/system 파일에서 등록 정보 값 업데이트	352
▼ 조정 등록 정보 값을 추가하거나 수정하는 방법	352
부하가 높은 도메인을 중지할 때 시간 초과가 발생할 수 있음	353
Oracle VM Server for SPARC와 함께 Oracle Solaris OS 작동	354
Oracle Solaris OS가 시작된 후 OpenBoot 펌웨어를 사용할 수 없음	354
서버의 전원을 껐다 켜	354
Oracle Solaris OS 중단 결과	354
컨트롤 도메인 재부트 결과	355
서비스 프로세서와 함께 Oracle VM Server for SPARC 사용	355
도메인 종속성 구성	356
도메인 종속성 예	357
종속성 주기	358
CPU 및 메모리 주소를 매핑하여 오류 발생 위치 확인	360
CPU 매핑	360
메모리 매핑	360
CPU 및 메모리 매핑 예	361
UUID(Universally Unique Identifier) 사용	362
가상 도메인 정보 명령 및 API	362
논리적 도메인 채널 사용	363
대량의 도메인 부트	366
Oracle VM Server for SPARC 시스템의 정상 종료 및 전원 껐다 켜기	367
▼ 여러 활성 도메인이 연결된 시스템 전원 끄는 방법	367
▼ 시스템 전원 껐다 켜는 방법	367
Logical Domains 변수 지속성	367
인터럽트 제한 조정	369
도메인 I/O 종속성 나열	371
Logical Domains Manager 데몬 사용으로 설정	372

▼ Logical Domains Manager 데몬을 사용으로 설정하는 방법	372
자동 저장 구성 데이터 저장 및 복원	373
자동 저장 구성 디렉토리 저장 및 복원	373
Logical Domains 제약 조건 데이터베이스 파일 저장 및 복원	374
출하 시 기본 구성과 도메인 사용 안함으로 설정	374
▼ 모든 게스트 도메인을 제거하는 방법	374
▼ 모든 도메인 구성을 제거하는 방법	375
▼ 출하 시 기본 구성을 복원하는 방법	375
▼ Logical Domains Manager를 사용 안함으로 설정하는 방법	375
▼ 서비스 프로세서에서 출하 시 기본 구성을 복원하는 방법	376
A 전원 관리 사용	377
전원 관리 사용	377
전원 관리 기능	378
전원 소비 데이터 보기	379
용어해설	385
색인	395

이 설명서 사용

- 개요 – Oracle Solaris OS 시스템 관리자에게 Oracle VM Server for SPARC 3.4 소프트웨어의 설치, 구성, 사용을 설명하는 세부정보와 절차를 제공합니다.
- 대상 – SPARC 서버에서 가상화를 관리하는 시스템 관리자.
- 필요한 지식 - 이러한 서버의 시스템 관리자는 UNIX 시스템 및 Oracle Solaris OS(Oracle Solaris 운영체제)를 사용할 수 있는 실제적인 지식을 보유하고 있어야 합니다.

제품 설명서 라이브러리

이 제품과 관련 제품들에 대한 설명서 및 리소스는 <http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html>에서 사용할 수 있습니다.

피드백

<http://www.oracle.com/goto/docfeedback>에서 이 설명서에 대한 피드백을 보낼 수 있습니다.

◆◆◆ 1 장

Oracle VM Server for SPARC 소프트웨어 개요

이 장에서는 Oracle VM Server for SPARC 소프트웨어의 개요를 제공합니다.

Oracle VM Server for SPARC에서는 Oracle SPARC T-Series 서버, SPARC M-Series 서버, SPARC S-Series 서버 및 Fujitsu M10 플랫폼에 매우 효율적인 엔터프라이즈급 가상화 기능을 제공합니다. Oracle VM Server for SPARC 소프트웨어를 사용하여 단일 시스템에 논리적 도메인이라는 가상 서버를 최대 128개까지 만들 수 있습니다. 이 종류의 구성을 통해 SPARC T-Series 서버, SPARC M-Series 서버, SPARC S-Series 서버와 Fujitsu M10 플랫폼, Oracle Solaris OS에서 제공하는 대규모 스레드를 활용할 수 있습니다.

이 장에서는 다음 주제를 다룹니다.

- “Oracle VM Server for SPARC 및 Oracle Solaris OS 버전 정보” [19]
- “하이퍼바이저 및 Logical Domains” [20]
- “Logical Domains Manager” [22]
- “Oracle VM Server for SPARC Management Information Base” [25]
- “Oracle VM Server for SPARC 문제 해결” [25]

주 - 이 설명서에 설명된 기능은 *Oracle VM Server for SPARC 3.4 설치 설명서*에 나와 있는 지원되는 모든 시스템 소프트웨어와 하드웨어 플랫폼에서 사용할 수 있습니다. 하지만 일부 기능은 지원되는 시스템 소프트웨어 및 하드웨어 플랫폼의 하위 세트에서만 사용할 수 있습니다. 이러한 예외사항에 대한 자세한 내용은 *Oracle VM Server for SPARC 3.4 릴리스 노트*의 “이 릴리스의 새로운 기능” 및 *What's New in Oracle VM Server for SPARC Software* (<http://www.oracle.com/technetwork/server-storage/vm/documentation/sparc-whatsnew-330281.html>)을 참조하십시오.

Oracle VM Server for SPARC 및 Oracle Solaris OS 버전 정보

Oracle VM Server for SPARC 소프트웨어는 특정 Oracle Solaris OS 버전, 필요한 소프트웨어 패치 및 특정 버전의 시스템 펌웨어에 따라 다릅니다. 자세한 내용은 *Oracle VM Server for SPARC 3.4 설치 설명서*의 “정규화된 Oracle Solaris OS 버전”을 참조하십시오.

게스트 도메인에서 실행되는 Oracle Solaris OS 버전은 `primary` 도메인에서 실행되는 Oracle Solaris OS 버전과 독립적입니다. 따라서 `primary` 도메인에서 Oracle Solaris 11 OS를 실행할 경우 게스트 도메인에서 Oracle Solaris 10 OS를 계속 실행할 수 있습니다.

주 - Oracle Solaris 10 OS는 `primary` 도메인에서 더 이상 지원되지 않습니다. 게스트 도메인에서 Oracle Solaris 10 OS를 계속 실행할 수 있습니다.

하이퍼바이저 및 Logical Domains

이 절에서는 논리적 도메인을 지원하는 SPARC 하이퍼바이저의 개요를 제공합니다.

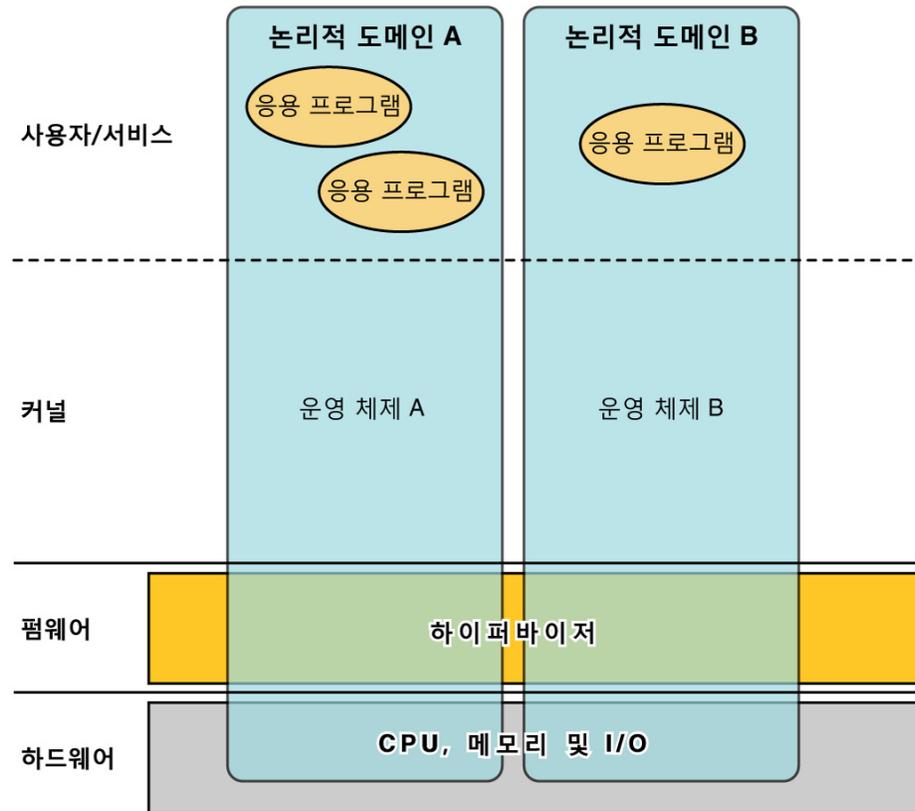
SPARC 하이퍼바이저는 운영체제가 작성될 수 있는 안정적인 가상화 컴퓨터 구조를 제공하는 작은 펌웨어 계층입니다. 하이퍼바이저를 사용하는 SPARC 서버는 하드웨어 기능을 통해 하이퍼바이저가 논리적 운영체제의 작업을 통제할 수 있도록 지원합니다.

논리적 도메인은 별도의 논리적 리소스 그룹으로 구성된 가상 컴퓨터입니다. 논리적 도메인은 단일 컴퓨터 시스템 안에 자체 운영체제와 신원을 가지고 있습니다. 각 논리적 도메인은 서버의 전원을 켜다가 켜 필요 없이 개별적으로 생성, 삭제, 재구성 및 재부트할 수 있습니다. 여러 논리적 도메인에서 다양한 응용 프로그램 소프트웨어를 실행하고 성능 및 보안 목적에 맞게 독립적으로 유지할 수 있습니다.

각 논리적 도메인은 하이퍼바이저가 사용 가능하도록 설정한 서버 리소스만 관찰하고 상호 작용할 수 있습니다. Logical Domains Manager는 하이퍼바이저가 컨트롤 도메인을 통해 수행할 작업을 지정할 수 있습니다. 따라서 하이퍼바이저는 서버 리소스를 분할해서 여러 운영체제 환경에 제한된 일부를 제공합니다. 이러한 분할 및 프로비전이 논리적 도메인을 만들기 위한 기본 방식입니다. 다음 다이어그램은 두 논리적 도메인을 지원하는 하이퍼바이저를 보여줍니다. 또한 Oracle VM Server for SPARC 기능을 구성하는 다음 계층을 보여줍니다.

- 사용자/서비스(응용 프로그램)
- 커널(운영체제)
- 펌웨어(하이퍼바이저)
- 하드웨어, CPU, 메모리, I/O 포함

그림 1 두 도메인을 지원하는 하이퍼바이저



특정 SPARC 하이퍼바이저가 지원하는 각 논리적 도메인의 개수와 기능은 서버에 종속적입니다. 하이퍼바이저는 서버의 전체 CPU, 메모리, I/O 리소스 중 일부를 주어진 논리적 도메인에 할당할 수 있습니다. 이 기능을 통해 각각의 자체 논리적 도메인 내에서 여러 운영체제를 동시에 지원할 수 있습니다. 임의 세분성을 사용하여 별도의 논리적 도메인 간에 리소스를 재배열할 수 있습니다. 예를 들어, CPU 스레드의 세분성으로 논리적 도메인에 CPU를 지정할 수 있습니다.

각 논리적 도메인은 다음과 같은 자체 리소스를 사용하여 완전히 독립적인 컴퓨터로 관리할 수 있습니다.

- 커널, 패치 및 조정 매개변수
- 사용자 계정 및 관리자
- 디스크
- 네트워크 인터페이스, MAC 주소 및 IP 주소

각 논리적 도메인은 서버의 전원을 켜다 켜지 않고도 서로 독립적으로 중지, 시작 및 재부트될 수 있습니다.

하이퍼바이저 소프트웨어는 논리적 도메인 사이에 구분을 유지할 책임이 있습니다. 또한 하이퍼바이저 소프트웨어는 논리적 도메인이 서로 통신할 수 있는 LDC(논리적 도메인 채널)를 제공합니다. LDC를 통해 도메인은 네트워킹 또는 디스크 서비스를 서로 제공할 수 있습니다.

서비스 프로세서(SP)는 시스템 컨트롤러(SC)라고도 하며, 물리적 컴퓨터를 모니터링하고 실행하지만 논리적 도메인은 관리하지 않습니다. Logical Domains Manager가 논리적 도메인을 관리합니다.

`ldm` 명령을 사용하여 Oracle VM Server for SPARC 소프트웨어를 관리할 뿐만 아니라 이제 Oracle VM Manager를 사용할 수 있습니다.

Oracle VM Manager는 Oracle VM 환경을 관리하는 데 사용할 수 있는 웹 기반 사용자 인터페이스입니다. 이 사용자 인터페이스의 이전 버전에서는 Oracle VM Server x86 소프트웨어만 관리했지만 Oracle VM Manager 3.2 및 Oracle VM Server for SPARC 3.0부터는 Oracle VM Server for SPARC 소프트웨어도 관리할 수 있습니다. Oracle VM Manager에 대한 자세한 내용은 [Oracle VM Documentation \(http://www.oracle.com/technetwork/documentation/vm-096300.html\)](http://www.oracle.com/technetwork/documentation/vm-096300.html)를 참조하십시오.

Logical Domains Manager

Logical Domains Manager를 사용하여 논리적 도메인을 만들고 관리할 뿐만 아니라 논리적 도메인을 물리적 리소스에 매핑할 수 있습니다. 하나의 Logical Domains Manager만 서버에 실행할 수 있습니다.

도메인의 역할

모든 논리적 도메인은 동일하며, 사용자가 지정한 역할을 기반으로 서로 구별할 수 있습니다. 논리적 도메인은 다음 역할을 수행할 수 있습니다.

- **컨트롤 도메인.** Logical Domains Manager는 이 도메인에서 실행되며, 다른 논리적 도메인을 만들고 관리하며 다른 도메인에 가상 리소스를 할당할 수 있습니다. 서버당 하나의 컨트롤 도메인만 가질 수 있습니다. 컨트롤 도메인은 Oracle VM Server for SPARC 소프트웨어를 설치할 때 처음 만든 도메인입니다. 컨트롤 도메인을 `primary`라고 합니다.
- **서비스 도메인.** 서비스 도메인은 가상 스위치, 가상 콘솔 집중기, 가상 디스크 서버와 같은 가상 장치 서비스를 다른 도메인에 제공합니다. 서비스 도메인을 여러 개 가질 수 있으며 모든 도메인을 서비스 도메인으로 구성할 수 있습니다.
- **I/O 도메인.** I/O 도메인은 PCI EXPRESS(PCIe) 컨트롤러를 통해 네트워크 카드와 같은 물리적 I/O 장치에 직접 액세스를 제공합니다. I/O 도메인은 다음을 소유할 수 있습니다.

- PCIe 루트 컴플렉스
- DIO(직접 I/O) 기능을 통한 PCIe 슬롯 또는 내장 PCIe 장치. “[PCIe 끝점 장치를 지정하여 I/O 도메인 만들기](#)” [135]를 참조하십시오.
- PCIe SR-IOV 가상 기능. [8장. PCIe SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기](#)를 참조하십시오.

I/O 도메인은 서비스 도메인으로 사용될 경우 물리적 I/O 장치를 가상 장치 형태로 다른 도메인과 공유할 수 있습니다.

- 루트 도메인. 루트 도메인에 PCIe 루트 컴플렉스가 지정됩니다. 이 도메인은 PCIe 패브릭을 소유하며 패브릭 오류 처리와 같은 모든 패브릭 관련 서비스를 제공합니다. 루트 도메인은 물리적 I/O 장치를 소유하고 직접 액세스를 제공하므로 I/O 도메인이기도 합니다.
플랫폼 구조에 따라 지정할 수 있는 루트 도메인 수가 다릅니다. 예를 들어, 8소켓 Oracle SPARC T5-8 서버를 사용하는 경우 최대 16개의 루트 도메인을 지정할 수 있습니다.
기본 루트 도메인은 `primary` 도메인입니다. 비`primary` 도메인을 루트 도메인으로 사용할 수 있습니다.
- 게스트 도메인. 게스트 도메인은 하나 이상의 서비스 도메인에서 제공한 가상 장치 서비스를 이용하는 비I/O 도메인입니다. 게스트 도메인의 경우 물리적 I/O 장치는 없고 가상 I/O 장치(예: 가상 디스크 및 가상 네트워크 인터페이스)만 있습니다.

아직 Oracle VM Server for SPARC로 구성되지 않은 기존 시스템에 Logical Domains Manager를 설치할 수 있습니다. 이 경우 현재 OS 인스턴스가 컨트롤 도메인이 됩니다. 또한 단 하나의 도메인, 즉 컨트롤 도메인으로 시스템이 구성됩니다. 컨트롤 도메인을 구성한 후 도메인을 추가하고 해당 응용 프로그램을 컨트롤 도메인에서 새 도메인으로 이동하여 전체 시스템을 가장 효율적으로 사용할 수 있도록 다른 도메인에서 응용 프로그램의 로드 균형을 조정할 수 있습니다.

명령줄 인터페이스

Logical Domains Manager는 명령줄 인터페이스(CLI)를 사용하여 논리적 도메인을 만들고 구성합니다. CLI는 단일 명령 `ldm`에 여러 하위 명령이 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Logical Domains Manager CLI를 사용하려면 Logical Domains Manager 데몬 `ldmd`가 실행 중이어야 합니다.

가상 입/출력

Oracle VM Server for SPARC 환경에서는 시스템 하나에서 도메인을 최대 128개(Fujitsu M10 서버의 경우 최대 256개) 구성할 수 있습니다. 일부 서버, 특히 단일 프로세서와 일부 듀얼 프로세서 시스템에는 제한된 수의 I/O 버스와 물리적 I/O 슬롯이 허용됩니다. 그 결과, 이러

한 시스템의 모든 도메인에서 물리적 디스크 및 네트워크 장치에 배타적 액세스를 제공하지 못할 수 있습니다. 물리적 장치에 액세스를 제공하려면 PCIe 버스 또는 끝점 장치를 도메인에 지정할 수 있습니다. 이 해결법은 모든 도메인에 배타적 장치 액세스를 제공하기에는 부족합니다. 직접 액세스할 수 있는 물리적 I/O 장치 수에 대한 이와 같은 제한은 가상화된 I/O 모델을 구현하여 해결할 수 있습니다. [6장. I/O 도메인 구성](#)을 참조하십시오.

물리적 I/O 액세스가 없는 논리적 도메인은 서비스 도메인과 통신하는 가상 I/O 장치로 구성됩니다. 서비스 도메인은 가상 장치 서비스를 실행하여 물리적 장치 또는 그 기능에 액세스할 수 있습니다. 이 클라이언트-서버 모델에서 가상 I/O 장치는 LDC(논리적 도메인 채널)라는 도메인 간 통신 채널을 통해 서비스 상대방과 또는 서로 통신합니다. 가상화된 I/O 기능은 가상 네트워크, 저장소, 콘솔에 지원됩니다.

가상 네트워크

Oracle VM Server for SPARC는 가상 네트워크 장치 및 가상 네트워크 스위치 장치를 사용하여 가상 네트워킹을 구현합니다. 가상 네트워크(vnet) 장치는 이더넷 장치를 에뮬레이트하고 지정된 채널을 사용하여 시스템의 다른 vnet 장치와 통신합니다. 가상 스위치(vsw) 장치는 주로 모든 가상 네트워크의 송/수신 패킷의 멀티플렉서로 작동합니다. vsw 장치는 서비스 도메인에서 물리적 네트워크 어댑터와 직접 상호 작용하고 가상 네트워크 대신 패킷을 송/수신합니다. 또한 vsw 장치는 단순 계층-2 스위치로 작동하고 시스템에서 연결된 vnet 장치 간에 패킷을 전환합니다.

가상 저장소

가상 저장소 기반구조는 클라이언트-서버 모델을 사용하여 논리적 도메인에 직접 지정되지 않은 블록 레벨 저장소에 액세스할 수 있습니다. 이 모델은 다음 구성요소를 사용합니다.

- 가상 디스크 클라이언트(vdc) - 블록 장치 인터페이스를 내보냅니다.
- 가상 디스크 서비스(vds) - 가상 디스크 클라이언트 대신 디스크 요청을 처리하여 서비스 도메인에 상주하는 백엔드 저장소에 제출합니다.

가상 디스크는 클라이언트 도메인에 일반 디스크로 나타나지만, 대부분의 디스크 작업이 가상 디스크 서비스로 전달되고 서비스 도메인에서 처리됩니다.

가상 콘솔

Oracle VM Server for SPARC 환경에서 primary 도메인의 콘솔 I/O는 서비스 프로세서로 지정됩니다. 그 밖의 모든 도메인의 콘솔 I/O는 가상 콘솔 집중기(vcc)를 실행 중인 서비스 도메인으로 재지정됩니다. vcc를 실행하는 도메인은 일반적으로 primary 도메인입니다. 가상 콘솔 집중 장치 서비스는 모든 도메인의 콘솔 트래픽 집중 장치로 작동하고 가상 네트워크 터미널 서버 데몬(vntsd) 상호 작용하여 UNIX 소켓을 통해 각 콘솔에 대한 액세스를 제공합니다.

리소스 구성

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템은 가상 CPU, 가상 I/O 장치, 암호화 장치, 메모리와 같은 리소스를 구성할 수 있습니다. 일부 리소스는 실행 중인 도메인에 동적으로 구성할 수 있는 반면, 나머지는 중지된 도메인에 구성해야 합니다. 컨트롤 도메인에서 리소스를 동적으로 구성할 수 없는 경우 먼저 지연된 재구성을 시작해야 합니다. 컨트롤 도메인이 재부트될 때까지 지연된 재구성이 구성 작업을 연기합니다. 자세한 내용은 “[리소스 재구성](#)” [299]을 참조하십시오.

지속적 구성

`ldm` 명령을 사용하여 논리적 도메인의 현재 구성을 서비스 프로세서에 저장할 수 있습니다. 구성을 추가하고, 사용할 구성을 지정하고, 구성을 제거하고, 구성을 나열할 수 있습니다. 자세한 내용은 `ldm(1M)` 매뉴얼 페이지를 참조하십시오. “[서비스 프로세서와 함께 Oracle VM Server for SPARC 사용](#)” [355]에 설명된 대로 SP에서 부트하도록 구성을 지정할 수도 있습니다.

구성 관리에 대한 내용은 “[도메인 구성 관리](#)” [337]를 참조하십시오.

Oracle VM Server for SPARC Management Information Base

Oracle VM Server for SPARC MIB(Management Information Base)를 사용하면 타사 시스템 관리 응용 프로그램이 SNMP(Simple Network Management Protocol)를 사용하여 도메인을 원격 모니터링하고 논리적 도메인을 시작/중지할 수 있습니다. 자세한 내용은 [Oracle VM Server for SPARC Management Information Base 사용자 설명서](#)를 참조하십시오.

Oracle VM Server for SPARC 문제 해결

다음 발행물에서 Oracle VM Server for SPARC 소프트웨어의 특정 문제에 대한 정보를 얻을 수 있습니다.

- [Oracle VM Server for SPARC 3.4 릴리스 노트](#)의 “[알려진 문제](#)”
- 정보 센터: [Overview of Oracle VM Server for SPARC \(LDoms\) \(Doc ID 1589473.2\)](#) (https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay?_afzLoop=227880986952919&id=1589473.2&_afzWindowMode=0&_adf.ctrl-state=wu09805r6_96)

Oracle VM Server for SPARC 보안

이 장에서는 Oracle VM Server for SPARC 시스템에서 사용으로 설정할 수 있는 몇 가지 보안 기능에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- “권한을 사용하여 논리적 도메인의 관리 위임” [27]
- “확인된 부트 사용” [31]

주 - 이 책에서는 superuser가 수행하는 예를 보여줍니다. 하지만 프로파일을 사용하여 사용자에게 관리 작업을 수행할 수 있는 더 미세 조정된 권한 설정을 부여할 수도 있습니다.

권한을 사용하여 논리적 도메인의 관리 위임

Logical Domains Manager 패키지는 로컬 권한 구성에 두 가지 사전 정의된 권한 프로파일을 추가합니다. 해당 권한 프로파일은 권한 없는 사용자에게 관리 권한을 위임합니다.

- LDoms Management 프로파일은 사용자가 모든 ldm 하위 명령을 사용할 수 있도록 허용합니다.
- LDoms Review 프로파일은 사용자가 모든 목록 관련 ldm 하위 명령을 사용할 수 있도록 허용합니다.

사용자 또는 역할에 이러한 권한 프로파일을 직접 지정할 수 있습니다. 역할에 지정할 경우 해당 역할이 사용자에게 지정됩니다. 이러한 프로파일 중 하나를 사용자에게 직접 지정할 경우 ldm 명령을 사용하여 도메인을 관리하려면 pfexec 명령 또는 프로파일 셸(예: pfbash 또는 pfksh)을 사용해야 합니다. 권한 구성을 기반으로 역할 또는 권한 프로파일을 사용할지 여부를 결정하십시오. [System Administration Guide: Security Services](#) 또는 [Securing Users and Processes in Oracle Solaris 11.3](#)을 참조하십시오.

다음 방법으로 사용자, 권한 부여, 권한 프로파일 및 역할을 구성할 수 있습니다.

- 파일을 사용하여 시스템에서 로컬로
- 이름 지정 서비스(예: LDAP)에서 중앙 집중식으로

Logical Domains Manager를 설치하면 필요한 권한 프로파일이 로컬 파일에 추가됩니다. 이름 지정 서비스에서 프로파일 및 역할을 구성하려면 [System Administration Guide: Naming](#)

*and Directory Services (DNS, NIS, and LDAP)*를 참조하십시오. Logical Domains Manager 패키지를 통해 전달된 권한 부여 및 실행 속성에 대한 개요는 “[Logical Domains Manager 프로파일 콘텐츠](#)” [30]를 참조하십시오. 이 장의 모든 예에서는 권한 구성에 로컬 파일이 사용된다고 가정합니다.

권한 프로파일 및 역할 사용



주의 - `usermod` 및 `rolemod` 명령을 사용하여 권한 부여, 권한 프로파일 또는 역할을 추가할 때는 신중해야 합니다.

- Oracle Solaris 11 OS의 경우 추가하는 각 권한 부여에 대해 플러스 기호(+)를 사용하여 값을 추가하십시오.
예를 들어, `usermod -A +auth username` 명령은 `rolemod` 명령과 유사하게 사용자 `username`에게 `auth` 권한 부여를 허가합니다.
- Oracle Solaris 10 OS의 경우 `usermod` 또는 `rolemod` 명령이 기존 값을 바꿉니다.
값을 바꾸지 않고 추가하려면 기존 값과 새 값 목록을 쉼표로 구분하여 지정하십시오.

사용자 권한 프로파일 관리

다음 절차에서는 로컬 파일을 사용하여 시스템에서 사용자 권한 프로파일을 관리하는 방법을 보여줍니다. 이름 지정 서비스에서 사용자 프로파일을 관리하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오.

▼ 사용자에게 권한 프로파일을 지정하는 방법

LDoms Management 프로파일이 직접 지정된 사용자는 프로파일 셀을 호출하여 `ldm` 명령을 보안 속성과 함께 실행해야 합니다. 자세한 내용은 [System Administration Guide: Security Services](#) 또는 [Securing Users and Processes in Oracle Solaris 11.3](#)을 참조하십시오.

1. 관리자로 전환합니다.

Oracle Solaris 11.3의 경우, [Securing Users and Processes in Oracle Solaris 11.3](#)의 1장, “About Using Rights to Control Users and Processes”를 참조하십시오.

2. 로컬 사용자 계정에 관리 프로파일을 지정합니다.

사용자 계정에 LDoms Review 프로파일 또는 LDoms Management 프로파일을 지정할 수 있습니다.

```
# usermod -P "profile-name" username
```

다음 명령은 사용자 `sam`에게 LDoms Management 프로파일을 지정합니다.

```
# usermod -P "LDoms Management" sam
```

사용자에게 역할 지정

다음 절차에서는 로컬 파일을 사용하여 역할을 만든 후 사용자에게 해당 역할을 지정하는 방법을 보여줍니다. 이름 지정 서비스에서 역할을 관리하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오.

이 절차를 수행할 경우 특정 역할이 지정된 사용자만 해당 역할을 맡을 수 있다는 장점이 있습니다. 역할에 암호가 지정된 경우 역할을 맡을 때 암호가 필요합니다. 이러한 두 가지 보안 계층은 암호가 있는 경우에도 역할이 지정되지 않은 사용자는 해당 역할을 맡을 수 없도록 합니다.

▼ 역할을 만들어 사용자에게 지정하는 방법

1. 관리자로 전환합니다.

Oracle Solaris 11.3의 경우, [Securing Users and Processes in Oracle Solaris 11.3](#)의 1장, "About Using Rights to Control Users and Processes"를 참조하십시오.

2. 역할을 만듭니다.

```
# roleadd -P "profile-name" role-name
```

3. 역할에 암호를 지정합니다.

새 암호를 지정한 다음 확인하라는 메시지가 표시됩니다.

```
# passwd role-name
```

4. 사용자에게 역할을 지정합니다.

```
# useradd -R role-name username
```

5. 사용자에게 암호를 지정합니다.

새 암호를 지정한 다음 확인하라는 메시지가 표시됩니다.

```
# passwd username
```

6. 해당 사용자로 로그인하고 필요한 경우 암호를 제공합니다.

```
# su username
```

7. 사용자에게 지정된 역할에 대한 액세스 권한이 있는지 확인합니다.

```
$ id
uid=nn(username) gid=nn(group-name)
$ roles
role-name
```

8. 역할을 맡고 필요한 경우 암호를 제공합니다.

```
$ su role-name
```

9. 사용자가 역할을 맡았는지 확인합니다.

```
$ id
uid=nn(role-name) gid=nn(group-name)
```

예 1 역할을 만들어 사용자에게 지정

이 예에서는 ldm_read 역할을 만들어 사용자 user_1에게 지정하고 사용자 user_1로 로그인한 다음 ldm_read 역할을 맡는 방법을 보여줍니다.

```
# roleadd -P "LDoms Review" ldm_read
# passwd ldm_read
New Password:
Re-enter new Password:
passwd: password successfully changed for ldm_read
# useradd -R ldm_read user_1
# passwd user_1
New Password:
Re-enter new Password:
passwd: password successfully changed for user_1
# su user_1
Password:
$ id
uid=95555(user_1) gid=10(staff)
$ roles
ldm_read
$ su ldm_read
Password:
$ id
uid=99667(ldm_read) gid=14(sysadmin)
```

Logical Domains Manager 프로파일 콘텐츠

Logical Domains Manager 패키지는 다음 권한 프로파일을 로컬 권한 프로파일 설명 데이터베이스에 추가합니다.

```
LDoms Power Mgmt Observability::View LDoms Power Consumption:auths=solaris.ldoms.ldmpower
LDoms Review::Review LDoms configuration:profiles=LDoms Power Mgmt Observability;
auths=solaris.ldoms.read
LDoms Management::Manage LDoms domains:profiles=LDoms Power Mgmt Observability;
auths=solaris.ldoms.*
```

또한 Logical Domains Manager 패키지는 LDoms Management 프로파일 및 LDoms Power Mgmt Observability 프로파일과 연관된 다음 실행 속성을 로컬 실행 프로파일 데이터베이스에 추가합니다.

```
LDoms Management:suser:cmd:::/usr/sbin/ldm:privs=file_dac_read,file_dac_search
LDoms Power Mgmt Observability:suser:cmd:::/usr/sbin/ldmpower:privs=file_dac_search
```

다음 표에서는 명령 실행에 필요한 해당 사용자 권한 부여와 함께 ldm 하위 명령을 나열합니다.

표 1 ldm 하위 명령 및 사용자 권한 부여

ldm 하위 명령 [†]	사용자 권한 부여
add-*	solaris.ldoms.write
bind-domain	solaris.ldoms.write
list	solaris.ldoms.read
list-*	solaris.ldoms.read
panic-domain	solaris.ldoms.write
remove-*	solaris.ldoms.write
set-*	solaris.ldoms.write
start-domain	solaris.ldoms.write
stop-domain	solaris.ldoms.write
unbind-domain	solaris.ldoms.write

[†]추가, 나열, 제거 또는 설정할 수 있는 모든 리소스를 나타냅니다.

확인된 부트 사용

Logical Domains Manager는 Oracle Solaris OS 확인된 부트 기술을 사용하여 부트 시 커널 모듈의 디지털 서명을 확인합니다. 서명 확인은 확인된 부트 정책이 사용으로 설정되어 있지 않은 경우 자동으로 실행됩니다. `boot-policy` 값에 따라 커널 모듈이 Oracle Solaris 릴리스 인증서 파일로 서명되지 않았거나 손상된 경우 게스트 도메인이 부트되지 않을 수 있습니다.

`ldm add-domain` 또는 `ldm set-domain` 명령을 사용하여 `boot-policy` 및 `module-policy` 등록 정보에 대한 값을 지정합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

이 기능을 사용하려면 시스템에 다음 시스템 펌웨어 및 운영체제 버전 이상이 실행되어야 합니다.

- 시스템 펌웨어 – 버전 9.5.0(Oracle SPARC 서버의 경우), 릴리스된 모든 버전(SPARC S7 시리즈 서버의 경우) 및 XCP 2280(Fujitsu M10 서버의 경우)
- 운영체제 – Oracle Solaris 11.2 OS

주 - 기본적으로 Oracle VM Server for SPARC 3.4 이전 버전을 사용하여 만든 모든 도메인은 `boot-policy=warning`을 설정합니다. 이 설정으로 커널 모듈이 서명되지 않았거나 손상된 경우 Oracle VM Server for SPARC 업데이트 후 도메인 부트 중 경고 메시지가 실행됩니다.

◆◆◆ 3 장

서비스 및 컨트롤 도메인 설정

이 장에서는 기본 서비스 및 컨트롤 도메인을 설정하는 데 필요한 절차를 설명합니다.
이 장에서는 다음 주제를 다룹니다.

- “출력 메시지” [33]
- “기본 서비스 만들기” [34]
- “컨트롤 도메인의 초기 구성” [35]
- “도메인을 사용하도록 재부트” [37]
- “Oracle Solaris 10 서비스 도메인과 다른 도메인 사이의 네트워킹 사용으로 설정” [38]
- “가상 네트워크 터미널 서버 데몬 사용으로 설정” [39]
- “ILOM 상호 연결이 사용으로 설정되었는지 확인” [39]

출력 메시지

컨트롤 도메인에서 리소스를 동적으로 구성할 수 없는 경우 먼저 지연된 재구성을 시작하는 것이 좋습니다. 컨트롤 도메인이 재부트될 때까지 지연된 재구성이 구성 작업을 연기합니다.

primary 도메인에서 지연된 재구성을 시작할 때 다음 메시지가 수신됩니다.

```
Initiating a delayed reconfiguration operation on the primary domain.  
All configuration changes for other domains are disabled until the  
primary domain reboots, at which time the new configuration for the  
primary domain also takes effect.
```

재부트할 때까지 primary 도메인에 매 후속 작업마다 다음 공지가 수신됩니다.

```
Notice: The primary domain is in the process of a delayed reconfiguration.  
Any changes made to the primary domain will only take effect after it reboots.
```

기본 서비스 만들기

컨트롤 도메인을 서비스 도메인으로 사용하고 다른 도메인에 대한 가상 장치를 만들려면 다음 가상 장치 서비스를 만들어야 합니다.

- vcc – 가상 콘솔 집중기 서비스
- vds – 가상 디스크 서버
- vsw – 가상 스위치 서비스

▼ 기본 서비스를 만드는 방법

1. 가상 콘솔 집중기(vcc) 서비스를 만듭니다. 가상 네트워크 터미널 서버 데몬(vntsd)에서 모든 논리적 도메인 콘솔의 집중기로 사용할 수 있습니다.

예를 들어, 다음 명령은 포트 범위 5000 - 5100의 가상 콘솔 집중기 서비스(primary-vcc0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

2. 가상 디스크 서버(vds)를 만듭니다. 가상 디스크를 논리적 도메인으로 가져올 수 있습니다.

예를 들어, 다음 명령은 가상 디스크 서버(primary-vds0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vds primary-vds0 primary
```

3. 가상 스위치 서비스(vsw)를 만듭니다. 논리적 도메인에서 가상 네트워크(vnet) 장치 사이의 네트워킹을 사용으로 설정할 수 있습니다.

각 논리적 도메인이 가상 스위치를 통해 범위 밖에서 통신해야 하는 경우 GLDV3 호환 네트워크 어댑터를 가상 스위치에 지정합니다.

게스트 도메인 네트워킹에 사용할 네트워크 장치의 가상 스위치 서비스(primary-vsw0)를 추가합니다.

```
primary# ldm add-vsw net-dev=network-device vsw-service primary
```

예를 들어, 다음 명령은 네트워크 장치 net0의 가상 스위치 서비스(primary-vsw0)를 컨트롤 도메인(primary)에 추가합니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

ldm list-netdev -b 명령을 사용하여 가상 스위치에 사용 가능한 백엔드 네트워크 장치를 확인할 수 있습니다. “[가상 스위치](#)” [223]를 참조하십시오.

ldm set-vsw 명령을 사용하여 net-dev 등록 정보 값을 동적으로 업데이트할 수 있습니다.

4. list-services 하위 명령을 사용하여 서비스가 만들어졌는지 확인합니다.

다음과 비슷한 출력이 나타나야 합니다.

```
primary# ldm list-services primary
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0

VCC
  NAME          PORT-RANGE
  primary-vcc0  5000-5100

VSW
  NAME          MAC              NET-DEV          DEVICE          MODE
  primary-vsw0  02:04:4f:fb:9f:0d net0             switch@0        prog,promisc
```

컨트롤 도메인의 초기 구성

처음에 모든 시스템 리소스가 컨트롤 도메인에 할당됩니다. 다른 논리적 도메인을 만들려면 이러한 리소스의 일부를 해제해야 합니다.

컨트롤 도메인 구성

▼ 컨트롤 도메인을 구성하는 방법

이 절차에는 컨트롤 도메인에 설정할 리소스의 예가 포함됩니다. 이러한 숫자는 예시일 뿐이며, 사용된 값이 사용자의 컨트롤 도메인에 부적합할 수 있습니다.

도메인 크기 조정 권장 사항은 [Oracle VM Server for SPARC Best Practices](http://www.oracle.com/technetwork/server-storage/vm/ovmsparc-best-practices-2334546.pdf) (<http://www.oracle.com/technetwork/server-storage/vm/ovmsparc-best-practices-2334546.pdf>)를 참조하십시오.

1. 컨트롤 도메인에 가상 CPU를 지정합니다.

컨트롤 도메인을 비롯한 서비스 도메인에서 게스트 도메인에 가상 디스크 및 가상 네트워크 I/O 작업을 수행하려면 CPU 및 메모리 리소스가 필요합니다. CPU 및 메모리 리소스 할당량은 게스트 도메인의 작업 부하에 따라 다릅니다.

예를 들어, 다음 명령은 2개의 CPU 코어(16개 가상 CPU 스레드)를 컨트롤 도메인 `primary`에 지정합니다. 나머지 가상 CPU 스레드는 게스트 도메인에 사용할 수 있습니다.

```
primary# ldm set-core 2 primary
```

응용 프로그램 요구사항에 따라 동적으로 실제 CPU 할당을 변경할 수 있습니다. `ldm list` 명령을 사용하여 컨트롤 도메인의 CPU 활용률을 확인합니다. 컨트롤 도메인의 CPU 활용률이 높으면 `ldm add-core` 및 `ldm set-core` 명령을 사용하여 서비스 도메인에 CPU 리소스를 추가합니다.

2. 컨트롤 도메인에 암호화 장치가 필요한지 여부를 확인합니다.

UltraSPARC T2, UltraSPARC T2 Plus, SPARC T3 플랫폼만 암호화 장치(MAU)를 사용할 수 있습니다. SPARC T4 시스템 및 Fujitsu M10 서버와 같은 최신 플랫폼에는 이미 암호화 가속기가 있으므로 해당 플랫폼에는 암호화 가속기를 지정할 필요가 없습니다.

구형 프로세서 중 하나를 사용하는 경우 컨트롤 도메인에서 각 CPU 전체 코어마다 하나의 암호화 장치를 지정합니다.

다음 예제는 2개의 암호화 리소스를 컨트롤 도메인 `primary`에 지정합니다.

```
primary# ldm set-crypto 2 primary
```

3. 컨트롤 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf primary
```

4. 컨트롤 도메인에 메모리를 지정합니다.

예를 들어, 다음 명령은 16GB의 메모리를 컨트롤 도메인 `primary`에 지정합니다. 이 설정에서는 게스트 도메인이 사용할 수 있도록 나머지 메모리를 남겨 둡니다.

```
primary# ldm set-memory 16G primary
```

5. 도메인 구성을 서비스 프로세서(SP)에 저장합니다.

예를 들어, 다음 명령은 `initial`이라는 구성을 추가합니다.

```
primary# ldm add-config initial
```

6. 다음 재부트 시 구성이 사용될 준비가 되었는지 확인합니다.

```
primary# ldm list-config
factory-default
initial [current]
```

이 `ldm list-config` 명령은 전원을 껐다 켜 후 `initial` 구성 세트가 사용될 것임을 보여줍니다.

7. 컨트롤 도메인을 재부트하여 재구성 변경사항을 적용합니다.

컨트롤 도메인의 초기 `factory-default` 구성에서 CPU 및 메모리 리소스 줄이기

CPU DR을 사용하여 초기 `factory-default` 구성에서 컨트롤 도메인의 코어 수를 줄일 수 있습니다. 그러나 컨트롤 도메인의 메모리를 줄이려면 메모리 DR 대신 지연된 재구성을 사용해야 합니다.

`factory-default` 구성의 경우 컨트롤 도메인은 호스트 시스템의 모든 메모리를 소유하고 있습니다. 활성 도메인은 요청된 모든 메모리를 추가하거나 더 일반적으로는 해제하지 않으므로 메모리 DR 기능은 이 용도로 적합하지 않습니다. 한편 해당 도메인에서 실행 중인 OS는 요청을 처

리하기 위해 최선을 다합니다. 또한 메모리 제거는 오래 실행되는 작업이 될 수 있습니다. 초기 컨트롤 도메인의 메모리 줄이기와 마찬가지로 대량 메모리 작업이 수행되는 경우 이러한 문제는 더 커집니다.

주 - Oracle Solaris OS를 ZFS 파일 시스템에 설치할 때 물리적 메모리 양에 따라 자동으로 스왑 및 덤프 영역의 크기를 조정하여 ZFS 루트 풀에 ZFS 볼륨으로 만듭니다. 도메인의 메모리 할당을 변경하려면 이 볼륨의 권장 크기를 변경할 수 있습니다. 컨트롤 도메인 메모리를 줄인 후에 필요한 것보다 할당량이 클 수도 있습니다. 스왑 공간 권장 사항은 [Managing File Systems in Oracle Solaris 11.3](#)의 “Planning for Swap Space”을 참조하십시오. 디스크 공간을 확보하기 전에 선택적으로 스왑 및 덤프 공간을 변경할 수 있습니다. [Managing ZFS File Systems in Oracle Solaris 11.3](#)의 “Managing ZFS Swap and Dump Devices”를 참조하십시오.

▼ 컨트롤 도메인의 초기 `factory-default` 구성에서 CPU 및 메모리 리소스를 줄이는 방법

이 절차에서는 컨트롤 도메인의 초기 `factory-default` 구성에서 CPU 및 메모리 리소스를 줄이는 방법을 보여줍니다. 먼저 CPU DR을 사용하여 코어 수를 줄인 다음, 메모리 양을 줄이기 전에 지연된 재구성을 시작합니다.

예제 값은 `ldmd` 데몬을 실행하고 마이그레이션을 수행하기에 충분한 리소스가 있는 소형 컨트롤 도메인의 CPU 및 메모리 크기에 해당합니다. 그러나 추가 목적으로 컨트롤 도메인을 사용하려면 필요에 따라 컨트롤 도메인에 더 많은 코어와 메모리를 지정할 수 있습니다.

1. `factory-default` 구성을 부트합니다.
2. 컨트롤 도메인을 구성합니다.
[컨트롤 도메인을 구성하는 방법 \[35\]](#)을 참조하십시오.

도메인을 사용하도록 재부트

구성 변경사항을 적용하고 다른 논리적 도메인에 사용할 리소스를 해제하려면 컨트롤 도메인을 재부트해야 합니다.

▼ 재부트하는 방법

- 컨트롤 도메인을 종료하고 재부트합니다.

```
primary# shutdown -y -g0 -i6
```

주 - 재부트하거나 전원을 껐다 켜면 새 구성이 인스턴스화됩니다. 전원을 껐다 켜야만 실제로 SP(서비스 프로세서)에 저장된 구성이 부트되고, 그런 다음 `list-config` 출력에 반영됩니다.

Oracle Solaris 10 서비스 도메인과 다른 도메인 사이의 네트워킹 사용으로 설정

기본적으로 Oracle Solaris 10 서비스 도메인과 시스템의 다른 도메인 사이의 네트워킹은 사용 안함으로 설정됩니다. Oracle Solaris 10 OS에서 기본적으로 네트워킹은 사용으로 설정되지 않기 때문에 가상 스위치 장치를 네트워크 장치로 구성하여 네트워킹을 사용으로 설정해야 합니다. 가상 스위치는 기본 물리적 장치(이 예제에서 `nxge0`)를 주 인터페이스로 대체하거나 도메인에 추가 네트워크 인터페이스로 구성할 수 있습니다.

해당하는 네트워크 백엔드 장치가 동일한 가상 LAN 또는 가상 네트워크에 구성된 경우, 게스트 도메인은 Oracle Solaris 10 서비스 도메인과 자동으로 통신할 수 있습니다.

▼ 가상 스위치를 주 인터페이스로 구성하는 방법

주 - 절차로 인해 도메인에 대한 네트워크 연결이 일시적으로 중단될 수 있으므로 Oracle Solaris 10 서비스 도메인의 콘솔에서 다음 절차를 수행하십시오.

필요한 경우, 가상 스위치와 물리적 네트워크 장치를 구성할 수 있습니다. 이 경우 2단계에서 가상 스위치를 만들고 물리적 장치를 삭제하지 마십시오(3단계 생략). 그 다음 가상 스위치를 정적 IP 주소 또는 동적 IP 주소로 구성해야 합니다. DHCP 서버에서 동적 IP 주소를 얻을 수 있습니다. 이 사례의 추가 정보와 예제는 [“NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성” \[240\]](#)을 참조하십시오.

1. 모든 인터페이스에 대한 주소 지정 정보를 인쇄합니다.

```
# ifconfig -a
```

2. 가상 스위치 네트워크 인터페이스를 구성합니다.

```
# ifconfig vsw0 plumb
```

3. 가상 스위치에 지정된 장치에 대한 물리적 인터페이스를 제거합니다(`net-dev`).

```
# ifconfig nxge0 down unplumb
```

4. 물리적 네트워크 장치(`nxge0`)의 등록 정보를 가상 스위치 장치(`vsw0`)로 마이그레이션하려면 다음 중 하나를 수행합니다.

- 네트워킹이 정적 IP 주소로 구성된 경우 가상 스위치에 대해 `nxge0`의 IP 주소와 넷마스크를 재사용합니다.

```
# ifconfig vsw0 IP-of-nxge0 netmask netmask-of-nxge0 broadcast + up
```

- 네트워킹이 DHCP로 구성된 경우 가상 스위치에 대해 DHCP를 사용으로 설정합니다.

```
# ifconfig vsw0 dhcp start
```

5. 필요한 구성 파일을 수정하여 이 변경사항을 영구 저장합니다.

```
# mv /etc/hostname.nxge0 /etc/hostname.vsw0
# mv /etc/dhcp.nxge0 /etc/dhcp.vsw0
```

가상 네트워크 터미널 서버 데몬 사용으로 설정

각 논리적 도메인의 가상 콘솔에 액세스하려면 가상 네트워크 터미널 서버 데몬(`vntsd`)을 사용으로 설정해야 합니다. 이 도메인의 사용 방법은 `vntsd(1M)` 매뉴얼 페이지를 참조하십시오.

▼ 가상 네트워크 터미널 서버 데몬을 사용으로 설정하는 방법

주 - `vntsd`를 사용으로 설정하기 전에 컨트롤 도메인에서 기본 서비스 `vconscn(vcc)`을 만들었는지 확인하십시오. 자세한 내용은 “기본 서비스 만들기” [34]를 참조하십시오.

1. 가상 네트워크 터미널 서버 데몬 `vntsd`를 사용으로 설정합니다.

```
primary# svcadm enable vntsd
```

2. `vntsd` 데몬이 사용으로 설정되었는지 확인합니다.

```
primary# svcs vntsd
STATE          STIME      FMRI
online         Oct_08    svc:/ldoms/vntsd:default
```

ILOM 상호 연결이 사용으로 설정되었는지 확인

ILOM 상호 연결은 SPARC T7 시리즈 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버의 SP(서비스 프로세서)와 `ldmd` 데몬 간의 통신에 필요하므로 사용 안함으로 설정하지 않아야 합니다. 자세한 내용은 `ilomconfig(1M)` 매뉴얼 페이지를 참조하십시오.

주 - 다른 SPARC T-Series 서버, SPARC M5, SPARC M6 서버에서는 ILOM 상호 연결을 사용 안함으로 설정하지 않는 것이 좋습니다. 하지만 사용 안함으로 설정하는 경우에도 `ldmd` 데몬은 SP와 통신을 계속할 수 있습니다.

SP와의 통신 오류로 인해 `ldm` 명령을 사용하여 SPARC T7 시리즈 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에서 도메인 구성을 관리하려는 시도가 실패할 경우 ILOM 상호 연결 상태를 확인하고 필요한 경우 `ilomconfig-interconnect` 서비스를 다시 사용으로 설정하십시오. [ILOM 상호 연결 구성을 확인하는 방법 \[40\]](#) 및 [ILOM 상호 연결 서비스를 다시 사용으로 설정하는 방법 \[40\]](#)을 참조하십시오.

▼ ILOM 상호 연결 구성을 확인하는 방법

1. `ilomconfig-interconnect` 서비스가 사용으로 설정되었는지 확인합니다.

```
primary# svcs ilomconfig-interconnect
STATE          STIME      FMRI
online         9:53:28   svc:/network/ilomconfig-interconnect:default
```

2. ILOM 상호 연결이 제대로 구성되었는지 확인합니다.

제대로 구성된 경우 State 값이 `enabled`로 표시되고 Host Interconnect IP Address 값이 `none`이 아닌 IP 주소로 표시됩니다.

```
primary# ilomconfig list interconnect
Interconnect
=====
State: enabled
Type: USB Ethernet
SP Interconnect IP Address: 169.254.182.76
Host Interconnect IP Address: 169.254.182.77
Interconnect Netmask: 255.255.255.0
SP Interconnect MAC Address: 02:21:28:57:47:16
Host Interconnect MAC Address: 02:21:28:57:47:17
```

3. `ldmd` 데몬이 SP와 통신할 수 있는지 확인합니다.

```
primary# ldm list-spconfig
```

▼ ILOM 상호 연결 서비스를 다시 사용으로 설정하는 방법

기본적으로 `ilomconfig-interconnect` 서비스는 사용으로 설정되어 있습니다. 이 서비스를 수동으로 다시 사용으로 설정해야 하는 경우 이 절차를 사용하십시오.

1. ILOM 상호 연결 서비스를 사용으로 설정합니다.

```
primary# svcadm enable ilomconfig-interconnect
```

2. `ilomconfig-interconnect` 서비스가 사용으로 설정되었는지 확인합니다.

```
primary# svcs ilomconfig-interconnect
STATE          STIME      FMRI
online         9:53:28   svc:/network/ilomconfig-interconnect:default
```

3. ILOM 상호 연결이 제대로 구성되었는지 확인합니다.

제대로 구성된 경우 State 값이 `enabled`로 표시되고 Host Interconnect IP Address 값이 `none`이 아닌 IP 주소로 표시됩니다.

```
primary# ilomconfig list interconnect
Interconnect
=====
State: enabled
Type: USB Ethernet
SP Interconnect IP Address: 169.254.182.76
Host Interconnect IP Address: 169.254.182.77
Interconnect Netmask: 255.255.255.0
SP Interconnect MAC Address: 02:21:28:57:47:16
Host Interconnect MAC Address: 02:21:28:57:47:17
```

4. `ldmd` 데몬이 SP와 통신할 수 있는지 확인합니다.

```
primary# ldm list-sponfig
```


◆◆◆ 4 장

게스트 도메인 설정

이 장에서는 게스트 도메인을 설정하는 데 필요한 절차를 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- “게스트 도메인 만들기 및 시작” [43]
- “게스트 도메인에 Oracle Solaris OS 설치” [46]

게스트 도메인 만들기 및 시작

게스트 도메인은 sun4v 플랫폼 및 하이퍼바이저가 제공한 가상 장치와 호환되는 운영체제를 실행해야 합니다. 즉, 현재 최소한 Oracle Solaris 10 11/06 OS를 실행하고 있어야 합니다. Oracle Solaris 10 1/13 OS를 실행하면 모든 Oracle VM Server for SPARC 3.4 기능이 제공됩니다. 필요한 관련 패치는 [Oracle VM Server for SPARC 3.4 설치 설명서](#)를 참조하십시오. 컨트롤 도메인에서 기본 서비스를 만들고 리소스를 재할당한 후에 게스트 도메인을 만들고 시작할 수 있습니다.

주 - CPU가 1024개보다 많이 지정되었거나 물리적 CPU ID가 1024개 이상인 게스트 도메인은 Oracle Solaris 10 OS를 실행할 수 없습니다. Oracle Solaris 10 OS를 실행하기 위해 CPU 또는 CPU ID 수를 1024 미만으로 줄이는 데 CPU DR을 사용할 수 없습니다.

▼ 게스트 도메인을 만들고 시작하는 방법

1. 논리적 도메인을 만듭니다.

다음 명령은 `ldg1`이라는 게스트 도메인을 만듭니다.

```
primary# ldm add-domain ldg1
```

2. 게스트 도메인에 CPU를 추가합니다.

다음 중 하나를 수행합니다.

- 가상 CPU를 추가합니다.

다음 명령은 8개의 가상 CPU를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-vcpu 8 ldg1
```

- 전체 코어를 추가합니다.

다음 명령은 2개의 전체 코어를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-core 2 ldg1
```

3. 게스트 도메인에 메모리를 추가합니다.

다음 명령은 2GB의 메모리를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-memory 2G ldg1
```

4. 게스트 도메인에 가상 네트워크 장치를 추가합니다.

다음 명령은 아래와 같은 가상 네트워크 장치를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-vnet vnet1 primary-vsw0 ldg1
```

설명:

- `vnet1`은 이 가상 네트워크 장치 인스턴스에 지정된 논리적 도메인의 고유한 인터페이스 이름으로, 나중에 `set-vnet` 또는 `remove-vnet` 하위 명령에서 참조할 수 있습니다.
- `primary-vsw0`은 연결할 기존 네트워크 서비스(가상 스위치)의 이름입니다.

주 - 단계 5-6은 `primary` 도메인에 가상 디스크 서버 장치(`vdsdev`)를 추가하고 게스트 도메인에 가상 디스크(`vdisk`)를 추가하기 위한 간단한 지침입니다. ZFS 볼륨 및 파일 시스템을 가상 디스크로 사용하는 방법을 알아보려면 [ZFS 볼륨을 단일 슬라이스 디스크로 내보내는 방법 \[174\]](#) 및 ["가상 디스크에서 ZFS 사용" \[186\]](#)을 참조하십시오.

5. 가상 디스크 서버가 게스트 도메인에 가상 디스크로 내보낼 장치를 지정합니다.

물리적 디스크, 디스크 슬라이스, 볼륨 또는 파일을 블록 장치로 내보낼 수 있습니다. 다음 예제는 물리적 디스크 및 파일을 보여줍니다.

- 물리적 디스크 예. 이 예에서는 아래와 같은 물리적 디스크를 추가합니다.

```
primary# ldm add-vdsdev /dev/dsk/c2t1d0s2 vol1@primary-vds0
```

설명:

- `/dev/dsk/c2t1d0s2`는 실제 물리적 장치의 경로 이름입니다. 장치를 추가할 때 경로 이름이 장치 이름과 쌍을 이루어야 합니다.
- `vol1`은 가상 디스크 서버에 추가하려는 장치에 지정할 고유 이름입니다. 볼륨 이름은 이 가상 디스크 서버 인스턴스에 고유해야 합니다. 이 가상 디스크 서버가 추가를 위해 해당 이름을 클라이언트로 내보내기 때문입니다. 장치를 추가할 때 볼륨 이름이 실제 장치의 경로 이름과 쌍을 이루어야 합니다.

- `primary-vds0`은 이 장치를 추가할 가상 디스크 서버의 이름입니다.
- **파일 예.** 이 예에서는 파일을 블록 장치로 내보냅니다.

```
primary# ldm add-vdsdev backend vol1@primary-vds0
```

설명:

- `backend`는 블록 장치로 내보낸 실제 파일의 경로 이름입니다. 장치를 추가할 때 백엔드 장치 이름과 쌍을 이루어야 합니다.
 - `vol1`은 가상 디스크 서버에 추가하려는 장치에 지정할 고유 이름입니다. 볼륨 이름은 이 가상 디스크 서버 인스턴스에 고유해야 합니다. 이 가상 디스크 서버가 추가를 위해 해당 이름을 클라이언트로 내보내기 때문입니다. 장치를 추가할 때 볼륨 이름이 실제 장치의 경로 이름과 쌍을 이루어야 합니다.
 - `primary-vds0`은 이 장치를 추가할 가상 디스크 서버의 이름입니다.
6. 게스트 도메인에 가상 디스크를 추가합니다.

다음 예제는 가상 디스크를 게스트 도메인 `ldg1`에 추가합니다.

```
primary# ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1
```

설명:

- `vdisk1`은 가상 디스크의 이름입니다.
- `vol1`은 연결할 기존 볼륨의 이름입니다.
- `primary-vds0`은 연결할 기존 가상 디스크 서버의 이름입니다.

주 - 가상 디스크는 여러 유형의 물리적 장치, 볼륨, 파일과 연관된 일반 블록 장치입니다. 가상 디스크는 SCSI 디스크와 동의어가 아니므로 디스크 레이블에서 대상 ID가 제외됩니다. 논리적 도메인의 가상 디스크는 `cNdNsN` 형식을 사용합니다. 여기서 `cN`은 가상 컨트롤러이고 `dN`은 가상 디스크 번호이고 `sN`은 슬라이스입니다.

7. 게스트 도메인에 대해 `auto-boot?` 및 `boot-device` 변수를 설정합니다.

주 - `boot-device` 등록 정보 값을 설정할 때는 가상 디스크 이름에 대문자가 포함된 경우에도 소문자만 사용하십시오.

다음 명령 예에서는 게스트 도메인 `ldg1`에 대해 `auto-boot?`를 `true`로 설정합니다.

```
primary# ldm set-var auto-boot\?=true ldg1
```

다음 명령 예에서는 게스트 도메인 `ldg1`에 대해 `boot-device`를 `vdisk1`로 설정합니다.

```
primary# ldm set-var boot-device=vdisk1 ldg1
```

8. 게스트 도메인 `ldg1`에 리소스를 바인드하고 도메인 목록을 나열하여 바인드되었는지 확인합니다.

```
primary# ldm bind-domain ldg1
```

```
primary# idm list-domain ldg1
NAME          STATE   FLAGS  CONS  VCPU MEMORY  UTIL  UPTIME
ldg1          bound  ----- 5000   8    2G
```

9. 게스트 도메인의 콘솔 포트를 찾으려면 앞의 `list-domain` 하위 명령의 출력을 살펴볼 수 있습니다.

머리글 `CONS` 아래를 보면 논리적 도메인 게스트 1(`ldg1`)의 콘솔 출력이 포트 `5000`에 바인드된 것을 알 수 있습니다.

10. 컨트롤 도메인에 로그인하고 로컬 호스트의 콘솔 포트에 직접 연결하여 다른 터미널에서 게스트 도메인의 콘솔에 연결합니다.

```
$ ssh hostname.domain-name
$ telnet localhost 5000
```

11. 게스트 도메인 `ldg1`을 시작합니다.

```
primary# idm start-domain ldg1
```

게스트 도메인에 Oracle Solaris OS 설치

이 절에서는 게스트 도메인에 Oracle Solaris OS를 설치할 수 있는 여러 가지 방법의 지침을 제공합니다.



주의 - Oracle Solaris OS 설치 중 가상 콘솔에서 연결을 끊지 마십시오.

Oracle Solaris 11 도메인의 경우 `DefaultFixed` 네트워크 구성 프로파일(NCP)을 사용합니다. 설치 중이나 설치 후에 이 프로파일을 사용으로 설정할 수 있습니다.

Oracle Solaris 11 설치 중에는 수동 네트워킹 구성을 선택합니다. Oracle Solaris 11 설치 후에는 `netadm list` 명령을 사용하여 `DefaultFixed` NCP가 사용으로 설정되었는지 확인합니다. [Configuring and Managing Network Components in Oracle Solaris 11.3](#)의 2, 3, 5, 6장을 참조하십시오.

메모리 크기 요구사항

Oracle VM Server for SPARC 소프트웨어는 도메인을 만들 때 메모리 크기 제한을 적용하지 않습니다. 메모리 크기 요구사항은 게스트 운영체제의 특성입니다. 제공된 메모리 양이 권

장 크기보다 작을 경우 일부 Oracle VM Server for SPARC 기능이 작동하지 않을 수 있습니다. Oracle Solaris 10 OS의 권장 및 최소 메모리 요구사항을 보려면 [Oracle Solaris 10 1/13 Installation Guide: Planning for Installation and Upgrade](#)의 “System Requirements and Recommendations”을 참조하십시오. Oracle Solaris 11 OS의 권장 및 최소 메모리 요구사항을 보려면 [Oracle Solaris 11 Release Notes](#), [Oracle Solaris 11.1 Release Notes](#), [Oracle Solaris 11.2 Release Notes](#), [Oracle Solaris 11.3 Release Notes](#)를 참조하십시오.

OpenBoot PROM은 도메인에 대한 최소 크기 제한을 갖고 있습니다. 현재까지 이 제한은 12MB입니다. 이 크기보다 작은 도메인을 설정하면 Logical Domains Manager가 도메인 크기를 자동으로 12MB로 늘립니다. Fujitsu M10 서버에 대한 최소 크기 제한은 256MB입니다. 메모리 크기 요구사항에 대한 자세한 내용은 시스템 펌웨어의 릴리스 노트를 참조하십시오.

메모리 DR(동적 재구성) 기능은 지정된 작업에 관련된 메모리의 주소 및 크기에 256MB를 강제로 할당합니다. “메모리 정렬” [318]을 참조하십시오.

▼ DVD에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법

1. Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS DVD를 DVD 드라이브에 넣습니다.

2. primary 도메인에서 볼륨 관리 데몬 vold(1M)를 중지합니다.

```
primary# svcadm disable volfs
```

3. 게스트 도메인(ldg1)을 중지하고 바인드를 해제합니다.

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
```

4. DVD-ROM 매체가 있는 DVD를 보조 볼륨 및 가상 디스크로 추가합니다.

다음 예에서는 c0t0d0s2를 Oracle Solaris 매체가 상주하는 DVD 드라이브로, dvd_vol@primary-vds0을 보조 볼륨으로, vdisk_cd_media를 가상 디스크로 사용합니다.

```
primary# ldm add-vdsdev options=ro /dev/dsk/c0t0d0s2 dvd_vol@primary-vds0
primary# ldm add-vdisk vdisk_cd_media dvd_vol@primary-vds0 ldg1
```

5. DVD가 보조 볼륨 및 가상 디스크로 추가되었는지 확인합니다.

```
primary# ldm list-bindings
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    8     8G     0.2%  22h 45m
...
VDS
NAME          VOLUME  OPTIONS  DEVICE
primary-vds0  vol1             /dev/dsk/c2t1d0s2
dvd_vol               /dev/dsk/c0t0d0s2
....
```

```

-----
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1          inactive -----          60    6G
...
DISK
  NAME          VOLUME          TOUT DEVICE  SERVER
  vdisk1        vol1@primary-vds0
  vdisk_cd_media dvd_vol@primary-vds0
.....

```

6. 게스트 도메인(ldg1)을 바인드하고 시작합니다.

```

primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..

```

7. 클라이언트 OpenBoot PROM에 장치 별명을 표시합니다.

이 예제에서 vdisk_cd_media의 장치 별명은 Oracle Solaris DVD이고, vdisk1은 Oracle Solaris OS를 설치할 수 있는 가상 디스크입니다.

```

ok devalias
vdisk_cd_media /virtual-devices@100/channel-devices@200/disk@1
vdisk1         /virtual-devices@100/channel-devices@200/disk@0
vnet1         /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name          aliases

```

8. 게스트 도메인의 콘솔에서 vdisk_cd_media(disk@1)를 슬라이스 f에 부트합니다.

```

ok boot vdisk_cd_media:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c), 1983-2010, Oracle and/or its affiliates. All rights reserved.

```

9. Oracle Solaris OS 설치를 계속합니다.

▼ Oracle Solaris ISO 파일에서 게스트 도메인에 Oracle Solaris OS를 설치하는 방법

1. 게스트 도메인(ldg1)을 중지하고 바인드를 해제합니다.

```

primary# ldm stop ldg1
primary# ldm unbind ldg1

```

2. Oracle Solaris ISO 파일을 보조 볼륨 및 가상 디스크로 추가합니다.

다음 예에서는 `solarisdvd.iso`를 Oracle Solaris ISO 파일로, `iso_vol@primary-vds0`을 보조 볼륨으로, `vdisk_iso`를 가상 디스크로 사용합니다.

```
primary# ldm add-vdsdev /export/solarisdvd.iso iso_vol@primary-vds0
primary# ldm add-vdisk vdisk_iso iso_vol@primary-vds0 ldg1
```

3. Oracle Solaris ISO 파일이 보조 볼륨 및 가상 디스크로 추가되었는지 확인합니다.

```
primary# ldm list-bindings
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    8     8G      0.2%  22h 45m
...
VDS
NAME          VOLUME      OPTIONS      DEVICE
primary-vds0  vol1                /dev/dsk/c2t1d0s2
iso_vol              /export/solarisdvd.iso
....
-----
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1         inactive -----          60    6G
...
DISK
NAME          VOLUME      TOUT ID DEVICE  SERVER  MPGROUP
vdisk1        vol1@primary-vds0
vdisk_iso     iso_vol@primary-vds0
....
```

4. 게스트 도메인(ldg1)을 바인드하고 시작합니다.

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

5. 클라이언트 OpenBoot PROM에 장치 별명을 표시합니다.

이 예제에서 `vdisk_iso`의 장치 별명은 Oracle Solaris ISO 이미지이고, `vdisk_install`은 디스크 공간입니다.

```
ok devalias
vdisk_iso      /virtual-devices@100/channel-devices@200/disk@1
vdisk1        /virtual-devices@100/channel-devices@200/disk@0
vnet1         /virtual-devices@100/channel-devices@200/network@0
virtual-console /virtual-devices/console@1
name          aliases
```

6. 게스트 도메인의 콘솔에서 vdisk_iso(disk@1)를 슬라이스 f에 부트합니다.

```
ok boot vdisk_iso:f
Boot device: /virtual-devices@100/channel-devices@200/disk@1:f File and args: -s
SunOS Release 5.10 Version Generic_139555-08 64-bit
Copyright (c) 1983-2010, Oracle and/or its affiliates. All rights reserved.
```

7. Oracle Solaris OS 설치를 계속합니다.

▼ Oracle Solaris 10 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법

주 - Oracle Solaris JumpStart 기능은 Oracle Solaris 10 OS에만 사용할 수 있습니다. [Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#)를 참조하십시오.

Oracle Solaris 11 OS의 자동 설치를 수행하려면 자동 설치 프로그램 기능을 사용하십시오. [Transitioning From Oracle Solaris 10 to Oracle Solaris 11.3](#)을 참조하십시오.

- 게스트 도메인에 대한 다른 디스크 장치 이름 형식이 반영되도록 **JumpStart** 프로파일을 수정합니다.

논리적 도메인의 가상 디스크 장치 이름은 물리적 디스크 장치 이름과 다릅니다. 가상 디스크 장치 이름에는 대상 ID(tN)를 넣지 않습니다. 일반적인 $cNtNdNsN$ 형식 대신, 가상 디스크 장치 이름은 $cNdNsN$ 형식을 사용합니다. 여기서 cN 은 가상 컨트롤러이고 dN 은 가상 디스크 번호이고 sN 은 슬라이스 번호입니다.

주 - 가상 디스크는 전체 디스크 또는 단일 슬라이스 디스크로 나타낼 수 있습니다. Oracle Solaris OS는 다중 분할 영역을 지정하는 일반 JumpStart 프로파일을 사용하여 전체 디스크에 설치할 수 있습니다. 단일 슬라이스 디스크는 단일 분할 영역 $s0$ 이 디스크 전체를 사용합니다. 단일 디스크에 Oracle Solaris OS를 설치하려면 단일 분할 영역(\wedge)이 디스크 전체를 사용하는 프로파일을 사용해야 합니다. 교체 분할 영역과 같은 다른 분할 영역은 정의할 수 없습니다. 전체 디스크 및 단일 슬라이스 디스크에 대한 자세한 내용은 "[가상 디스크 표시](#)" [166]를 참조하십시오.

- **UFS 루트 파일 시스템을 설치하기 위한 JumpStart 프로파일**

[Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#)를 참조하십시오.

일반 UFS 프로파일

```
filesys c1t1d0s0 free /
filesys c1t1d0s1 2048 swap
filesys c1t1d0s5 120 /spare1
filesys c1t1d0s6 120 /spare2
```

전체 디스크에 도메인을 설치하기 위한 실제 UFS 프로파일

```
filesys c0d0s0 free /
filesys c0d0s1 2048 swap
filesys c0d0s5 120 /spare1
filesys c0d0s6 120 /spare2
```

단일 슬라이스 디스크에 도메인을 설치하기 위한 실제 UFS 프로파일

```
filesys c0d0s0 free /
```

- **ZFS 루트 파일 시스템을 설치하기 위한 JumpStart 프로파일**

[Oracle Solaris 10 1/13 Installation Guide: JumpStart Installations](#) 의 9 장, “Installing a ZFS Root Pool With JumpStart”를 참조하십시오.

일반 ZFS 프로파일

```
pool rpool auto 2G 2G c1t1d0s0
```

도메인을 설치하기 위한 실제 ZFS 프로파일

```
pool rpool auto 2G 2G c0d0s0
```


도메인 콘솔 사용

이 장에서는 Oracle VM Server for SPARC 시스템에서 사용으로 설정할 수 있는 도메인 콘솔 기능에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- “권한을 사용하여 도메인 콘솔에 대한 액세스 제어” [53]
- “도메인 콘솔 로깅 사용” [58]
- “네트워크를 통해 게스트 콘솔에 연결” [59]
- “콘솔 그룹 사용” [60]

주 - 이 책에서는 `superuser`가 수행하는 예를 보여줍니다. 하지만 프로파일을 사용하여 사용자에게 관리 작업을 수행할 수 있는 더 미세 조정된 권한 설정을 부여할 수도 있습니다.

권한을 사용하여 도메인 콘솔에 대한 액세스 제어

기본적으로 모든 사용자가 모든 도메인 콘솔에 액세스할 수 있습니다. 도메인 콘솔에 대한 액세스를 제어하려면 권한 부여 확인이 수행되도록 `vntsd` 데몬을 구성하십시오. `vntsd` 데몬은 `vntsd/authorization`이라는 SMF(서비스 관리 기능) 등록 정보를 제공합니다. 도메인 콘솔 또는 콘솔 그룹에 대해 사용자 및 역할의 권한 부여 확인이 사용으로 설정되도록 이 등록 정보를 구성할 수 있습니다. 권한 부여 확인을 사용으로 설정하려면 `svccfg` 명령을 사용하여 이 등록 정보의 값을 `true`로 설정하십시오. 이 옵션이 사용으로 설정된 상태에서 `vntsd`는 `localhost`에서만 연결을 수신하고 수락합니다. `vntsd/authorization`이 사용으로 설정된 상태에서는 `listen_addr` 등록 정보가 대체 IP 주소를 지정하는 경우에도 `vntsd`가 대체 IP 주소를 무시하고 계속 `localhost`에서만 수신합니다.



주의 - `localhost` 이외의 다른 호스트를 사용하려면 `vntsd` 서비스를 구성하지 마십시오.

`localhost` 이외의 다른 호스트를 지정할 경우 더 이상 컨트롤 도메인에서 게스트 도메인 콘솔로의 연결이 제한되지 않습니다. `telnet` 명령을 사용하여 게스트 도메인에 원격으로 연결할 경우 네트워크를 통해 일반 텍스트로 로그인 자격 증명이 전달됩니다.

기본적으로 모든 게스트 콘솔에 액세스할 수 있는 권한 부여는 로컬 권한 부여 설명 데이터베이스에서 제공됩니다.

```
solaris.vntsd.consoles:::Access All LDoms Guest Consoles:::
```

로컬 파일에서 사용자 또는 역할에 필요한 권한 부여를 지정하려면 `usermod` 명령을 사용하십시오. 이 명령은 필요한 권한 부여를 가진 사용자 또는 역할만 지정된 도메인 콘솔 또는 콘솔 그룹에 액세스할 수 있도록 허용합니다. 이름 지정 서비스에서 사용자 또는 역할에 권한 부여를 지정하려면 [System Administration Guide: Naming and Directory Services \(DNS, NIS, and LDAP\)](#)를 참조하십시오.

모든 도메인 콘솔에 대한 액세스를 제어할 수도 있고, 단일 도메인 콘솔에 대한 액세스를 제어할 수도 있습니다.

- 모든 도메인 콘솔에 대한 액세스를 제어하려면 [역할을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법 \[54\]](#) 및 [권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법 \[55\]](#)을 참조하십시오.
- 단일 도메인 콘솔에 대한 액세스를 제어하려면 [역할을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법 \[56\]](#) 및 [권한 프로파일을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법 \[58\]](#)을 참조하십시오.

▼ 역할을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법

1. 콘솔 권한 부여 확인을 사용으로 설정하여 도메인 콘솔에 대한 액세스를 제한합니다.

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. 모든 도메인 콘솔에 대한 액세스를 허용하는 `solaris.vntsd.consoles` 권한 부여를 가진 역할을 만듭니다.

```
primary# roleadd -A solaris.vntsd.consoles role-name
primary# passwd role-name
```

3. 사용자에게 새 역할을 지정합니다.

```
primary# usermod -R role-name username
```

예 2 역할을 사용하여 모든 도메인 콘솔에 대한 액세스 제어

먼저 도메인 콘솔에 대한 액세스가 제한되도록 콘솔 권한 부여 확인을 사용으로 설정합니다.

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
primary# ldm ls
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary      active    -n-cv-  UART     8       16G      0.2%   47m
```

```
ldg1          active  -n--v- 5000   2    1G      0.1% 17h 50m
ldg2          active  -t---- 5001   4    2G      25%  11s
```

다음 예에서는 모든 도메인 콘솔에 대한 액세스를 허용하는 `solaris.vntsd.consoles` 권한 부여를 가진 `all_cons` 역할을 만드는 방법을 보여줍니다.

```
primary# roleadd -A solaris.vntsd.consoles all_cons
primary# passwd all_cons
New Password:
Re-enter new Password:
passwd: password successfully changed for all_cons
```

이 명령은 사용자 `sam`에게 `all_cons` 역할을 지정합니다.

```
primary# usermod -R all_cons sam
```

그러면 사용자 `sam`이 `all_cons` 역할을 맡고 모든 콘솔에 액세스할 수 있습니다. 예를 들어, 다음과 같습니다.

```
$ id
uid=700299(sam) gid=1(other)
$ su all_cons
Password:
$ telnet localhost 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..

$ telnet localhost 5001
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.

Connecting to console "ldg2" in group "ldg2" ....
Press ~? for control options ..
```

이 예에서는 권한이 부여되지 않은 사용자 `dana`가 도메인 콘솔에 액세스하려고 시도할 때 발생하는 오류를 보여줍니다.

```
$ id
uid=702048(dana) gid=1(other)
$ telnet localhost 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
Connection to 0 closed by foreign host.
```

▼ 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법

1. 콘솔 권한 부여 확인을 사용으로 설정하여 도메인 콘솔에 대한 액세스를 제한합니다.

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
```

```
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. **solaris.vntsd.consoles** 권한 부여를 가진 권한 프로파일을 만듭니다.

profiles 명령을 사용하여 새 프로파일을 만듭니다.

```
primary# profiles -p "LDoms Consoles" \
'set desc="Access LDoms Consoles"; set auths=solaris.vntsd.consoles'
```

3. **사용자에게 권한 프로파일을 지정합니다.**

```
primary# usermod -P +"LDoms Consoles" username
```

4. **해당 사용자로 도메인 콘솔에 연결합니다.**

```
$ telnet localhost 5000
```

예 3 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스 제어

다음 예에서는 권한 프로파일을 사용하여 모든 도메인 콘솔에 대한 액세스를 제어하는 방법을 보여줍니다. profiles 명령을 사용하여 권한 프로파일 설명 데이터베이스에 solaris.vntsd.consoles 권한 부여를 가진 권한 프로파일을 만듭니다.

```
primary# profiles -p "LDoms Consoles" \
'set desc="Access LDoms Consoles"; set auths=solaris.vntsd.consoles'
```

사용자에게 권한 프로파일을 지정합니다.

```
primary# usermod -P +"LDoms Consoles" sam
```

다음 명령은 사용자가 sam이며 All, Basic Solaris User 및 LDoms Consoles 권한 프로파일이 적용되는지 확인하는 방법을 보여줍니다. telnet 명령은 ldg1 도메인 콘솔에 액세스하는 방법을 보여줍니다.

```
$ id
uid=702048(sam) gid=1(other)
```

```
$ profiles
All
Basic Solaris User
LDoms Consoles
```

```
$ telnet localhost 5000
Trying 0.0.0.0...
Connected to 0.
Escape character is '^['.
```

```
Connecting to console "ldg1" in group "ldg1" ....
Press ~? for control options ..
```

▼ 역할을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법

1. **콘솔 권한 부여 확인을 사용으로 설정하여 도메인 콘솔에 대한 액세스를 제한합니다.**

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
```

```
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. 권한 부여 설명 데이터베이스에 단일 도메인에 대한 권한 부여를 추가합니다.
권한 부여 이름은 도메인 이름에서 파생되며 `solaris.vntsd.console-domain` 형식입니다.

```
solaris.vntsd.console-domain::Access domain Console::
```

3. 도메인 콘솔에 대해서만 액세스를 허용하는 새 권한 부여를 가진 역할을 만듭니다.

```
primary# roleadd -A solaris.vntsd.console-domain role-name
primary# passwd role-name
New Password:
Re-enter new Password:
passwd: password successfully changed for role-name
```

4. 사용자에게 `role-name` 역할을 지정합니다.

```
primary# usermod -R role-name username
```

예 4 단일 도메인 콘솔에 액세스

이 예에서는 사용자 `terry`가 `ldg1cons` 역할을 맡고 `ldg1` 도메인 콘솔에 액세스하는 방법을 보여줍니다.

먼저 권한 부여 설명 데이터베이스에 단일 도메인 `ldg1`에 대한 권한 부여를 추가합니다.

```
solaris.vntsd.console-ldg1::Access ldg1 Console::
```

그런 다음 도메인 콘솔에 대해서만 액세스를 허용하는 새 권한 부여를 가진 역할을 만듭니다.

```
primary# roleadd -A solaris.vntsd.console-ldg1 ldg1cons
primary# passwd ldg1cons
New Password:
Re-enter new Password:
passwd: password successfully changed for ldg1cons
```

사용자 `terry`에게 `ldg1cons` 역할을 지정하고 `ldg1cons` 역할을 맡은 다음 도메인 콘솔에 액세스합니다.

```
primary# usermod -R ldg1cons terry
primary# su terry
Password:
$ id
uid=700300(terry) gid=1(other)
$ su ldg1cons
Password:
$ id
uid=700303(ldg1cons) gid=1(other)
$ telnet localhost 5000
Trying 0.0.0.0...
Escape character is '^]'.
```

```
Connecting to console "ldg1" in group "ldg1" ...
Press ~? for control options ..
```

다음 예에서는 사용자 `terry`가 `ldg2` 도메인 콘솔에 액세스할 수 없음을 보여줍니다.

```
$ telnet localhost 5001
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
Connection to 0 closed by foreign host.
```

▼ 권한 프로파일을 사용하여 단일 콘솔에 대한 액세스를 제어하는 방법

1. 콘솔 권한 부여 확인을 사용으로 설정하여 도메인 콘솔에 대한 액세스를 제한합니다.

```
primary# svccfg -s vntsd setprop vntsd/authorization = true
primary# svcadm refresh vntsd
primary# svcadm restart vntsd
```

2. 권한 부여 설명 데이터베이스에 단일 도메인에 대한 권한 부여를 추가합니다. 다음 항목 예에서는 단일 도메인 콘솔에 대한 권한 부여를 추가합니다.

```
solaris.vntsd.console-domain:::Access domain Console:::
```

3. 특정 도메인 콘솔에 액세스하는 권한 부여를 가진 권한 프로파일을 만듭니다.

profiles 명령을 사용하여 새 프로파일을 만듭니다.

```
primary# profiles -p "domain Console" \
'set desc="Access domain Console";
set auths=solaris.vntsd.console-domain'
```

4. 권한 프로파일을 지정합니다.

```
primary# usermod -P +"domain Console" username
```

도메인 콘솔 로깅 사용

Oracle VM Server for SPARC 환경에서 primary 도메인의 콘솔 I/O는 SP(서비스 프로세서)로 지정됩니다. 기타 모든 도메인의 콘솔 I/O는 가상 콘솔 집중 장치 vcc를 실행하는 서비스 도메인으로 재지정됩니다. 서비스 도메인이 Oracle Solaris 11 OS를 실행하는 경우 게스트 도메인 콘솔 출력을 파일에 기록할 수 있습니다.

서비스 도메인은 논리적 도메인에 대한 콘솔 로깅을 지원합니다. 서비스 도메인은 Oracle Solaris 11 OS를 실행해야 하지만, 기록 중인 게스트 도메인은 Oracle Solaris 10 OS 또는 Oracle Solaris 11 OS를 실행할 수 있습니다.

도메인 콘솔 로그는 vcc 서비스를 제공하는 /var/log/vntsd/domain/console-log라는 서비스 도메인의 파일에 저장됩니다. logadm 명령을 사용하여 콘솔 로그 파일을 교체할 수 있습니다. logadm(1M) 및 logadm.conf(4) 매뉴얼 페이지를 참조하십시오.

Oracle VM Server for SPARC 소프트웨어를 통해 선별적으로 각 논리적 도메인에 대한 콘솔 로깅을 사용 및 사용 안함으로 설정할 수 있습니다. 기본적으로 콘솔 로깅은 사용으로 설정되어 있습니다.

▼ 콘솔 로깅을 사용 또는 사용 안함으로 설정하는 방법

도메인이 동일한 콘솔 그룹에 속해 있는 경우에도 각 개별 논리적 도메인에 대한 콘솔 로깅을 사용 또는 사용 안함으로 설정해야 합니다.

1. 도메인에 대한 현재 콘솔 설정을 나열합니다.

```
primary# ldm list -o console domain
```

2. 도메인을 중지하고 바인드 해제합니다.

콘솔 설정을 수정하기 전에 도메인은 비활성 및 바인드 해제 상태여야 합니다.

```
primary# ldm stop domain
primary# ldm unbind domain
```

3. 콘솔 로깅을 사용 또는 사용 안함으로 설정합니다.

- 콘솔 로깅을 사용으로 설정하려면 다음 명령을 사용합니다.

```
primary# ldm set-vcons log=on domain
```

- 콘솔 로깅을 사용 안함으로 설정하려면 다음 명령을 사용합니다.

```
primary# ldm set-vcons log=off domain
```

도메인 콘솔 로깅에 대한 서비스 도메인 요구사항

Oracle Solaris 11.1 이전의 OS 버전을 실행하는 서비스 도메인에 연결된 도메인은 기록할 수 없습니다.

주 - 도메인에 대한 콘솔 로깅을 사용으로 설정한 경우에도 서비스 도메인에서 필요한 지원이 제공되지 않으면 도메인의 가상 콘솔이 기록되지 않습니다.

네트워크를 통해 게스트 콘솔에 연결

`listen_addr` 등록 정보가 `vntsd(1M)` SMF 매니페스트에서 컨트롤 도메인의 IP 주소로 설정된 경우 네트워크를 통해 게스트 콘솔에 연결할 수 있습니다. 예를 들어, 다음과 같습니다.

```
$ telnet hostname 5001
```

주 - 콘솔에 대한 네트워크 액세스를 사용으로 설정하면 보안이 영향을 받을 수 있습니다. 모든 사용자가 콘솔에 연결할 수 있으므로 기본적으로 사용 안함으로 설정되어 있습니다.

서비스 관리 기능 매니페스트는 서비스를 기술하는 XML 파일입니다. SMF 매니페스트 만들기
 기에 대한 자세한 내용은 [Oracle Solaris 10 System Administrator Documentation \(http://download.oracle.com/docs/cd/E18752_01/index.html\)](http://download.oracle.com/docs/cd/E18752_01/index.html)를 참조하십시오.

주 - 콘솔을 통해 게스트 도메인에서 영어 이외의 OS에 액세스하려면 콘솔 터미널이 OS에 필요한 로케일이어야 합니다.

콘솔 그룹 사용

가상 네트워크 터미널 서버 데몬 `vntsd`를 사용하면 단일 TCP 포트를 통해 다중 도메인 콘솔에 대한 액세스를 제공할 수 있습니다. 도메인을 만들 때 Logical Domains Manager는 해당 도메인의 콘솔에 대해 새 기본 그룹을 만들어 각 콘솔에 고유한 TCP 포트를 지정합니다. 그러면 콘솔 자체와 반대로 콘솔 그룹에 TCP 포트가 지정됩니다. `set-vcons` 하위 명령을 사용하여 기존 그룹에 콘솔을 바인드할 수 있습니다.

▼ 하나의 그룹에 다중 콘솔을 결합하는 방법

1. 하나의 그룹에 다중 도메인에 대한 콘솔을 바인드합니다.

다음 예에서는 동일한 콘솔 그룹(`group1`)에 3개의 다른 도메인(`ldg1`, `ldg2` 및 `ldg3`)에 대한 콘솔을 바인드하는 방법을 보여줍니다.

```
primary# ldm set-vcons group=group1 service=primary-vc0 ldg1
primary# ldm set-vcons group=group1 service=primary-vc0 ldg2
primary# ldm set-vcons group=group1 service=primary-vc0 ldg3
```

2. 연관된 TCP 포트(이 예의 경우 포트 5000의 `localhost`)에 연결합니다.

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
```

도메인 콘솔 중 하나를 선택하라는 메시지가 표시됩니다.

3. `l(list)`을 선택하여 그룹 내 도메인을 나열합니다.

```
primary-vnts-group1: h, l, c{id}, n{name}, q: l
DOMAIN ID          DOMAIN NAME          DOMAIN STATE
0                   ldg1                 online
1                   ldg2                 online
```

2

ldg3

online

주 - 다른 그룹 또는 vcc 인스턴스에 콘솔을 재지정하려면 도메인의 바인드를 해제해야 합니다. 즉, 도메인이 비활성 상태여야 합니다. SMF를 구성 및 사용하여 vntsd를 관리하고 콘솔 그룹을 사용하는 것과 관련된 자세한 내용은 vntsd(1M) 매뉴얼 페이지를 참조하십시오.

I/O 도메인 구성

이 장에서는 I/O 도메인과 Oracle VM Server for SPARC 환경에서 I/O 도메인을 구성하는 방법에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- “I/O 도메인 개요” [63]
- “일반적인 I/O 도메인 만들기 지침” [64]

I/O 도메인 개요

I/O 도메인은 물리적 I/O 장치를 직접 소유하며 물리적 I/O 장치에 대한 직접 액세스 권한을 가집니다. 도메인에 PCIe(PCI EXPRESS) 버스, PCIe 끝점 장치 또는 PCIe SR-IOV 가상 기능을 지정하여 만들 수 있습니다. `ldm add-io` 명령을 사용하여 도메인에 버스, 장치 또는 가상 기능을 지정하십시오.

다음과 같은 이유로 I/O 도메인을 구성할 수 있습니다.

- I/O 도메인은 물리적 I/O 장치에 대한 직접 액세스 권한을 가지므로 가상 I/O와 연관된 성능 오버헤드가 발생하지 않습니다. 따라서 I/O 도메인의 I/O 성능이 베어 메탈 시스템의 I/O 성능과 거의 동일합니다.
- I/O 도메인은 게스트 도메인에 사용될 가상 I/O 서비스를 호스트할 수 있습니다.

I/O 도메인 구성에 대한 자세한 내용은 다음 장의 정보를 참조하십시오.

- 7장. PCIe 버스를 지정하여 루트 도메인 만들기
- 9장. 직접 I/O를 사용하여 I/O 도메인 만들기
- 8장. PCIe SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기
- 10장. 비_{primary} 루트 도메인 사용

주 - PCIe 버스, PCIe 끝점 장치 또는 SR-IOV 가상 기능이 있는 도메인은 마이그레이션할 수 없습니다. 기타 마이그레이션 제한 사항에 대한 자세한 내용은 14장. 도메인 마이그레이션을 참조하십시오.

일반적인 I/O 도메인 만들기 지침

I/O 도메인은 PCIe 버스, NIU(네트워크 인터페이스 장치), PCIe 끝점 장치, PCIe SR-IOV(단일 루트 I/O 가상화) 가상 기능 등 하나 이상의 I/O 장치에 대한 직접 액세스 권한을 가질 수 있습니다.

I/O 장치에 대한 이 유형의 직접 액세스 권한은 보다 넓은 I/O 대역폭을 사용하여 다음을 가능하게 합니다.

- I/O 도메인의 응용 프로그램에 서비스 제공
- 게스트 도메인에 가상 I/O 서비스 제공

다음과 같은 기본적인 지침을 통해 I/O 대역폭을 효율적으로 사용할 수 있습니다.

- CPU 코어 단위에서 CPU 리소스를 지정하십시오. I/O 도메인 내 I/O 장치의 유형 및 수에 따라 하나 이상의 CPU 코어를 지정하십시오.
예를 들어, 1Gbps 이더넷 장치의 경우 10Gbps 이더넷 장치에 비해 보다 적은 CPU 코어로도 대역폭을 완전히 사용할 수 있습니다.
- 메모리 요구사항을 준수하십시오. 메모리 요구사항은 도메인에 지정된 I/O 장치의 유형에 따라 다릅니다. I/O 장치당 최소 4GB가 권장됩니다. 지정한 I/O 장치 수가 많을수록 메모리를 보다 많이 할당해야 합니다.
- PCIe SR-IOV 기능을 사용할 때는 다른 I/O 장치에 적용되는 것과 동일한 지침에 따라 각 SR-IOV 가상 기능을 사용하십시오. 따라서 가상 기능에서 사용 가능한 대역폭을 완전히 사용하려면 하나 이상의 CPU 코어 및 메모리(GB)를 지정하십시오.

CPU 및 메모리 리소스가 충분하지 않은 많은 수의 가상 기능을 만들어 도메인에 지정하면 구성이 최적화되지 않을 수 있습니다.

SPARC T5 및 SPARC M6 플랫폼을 비롯한 SPARC 시스템은 한정된 수의 인터럽트를 제공하므로 Oracle Solaris는 각 장치가 사용할 수 있는 인터럽트 수를 제한합니다. 기본 제한은 표준 시스템 구성 요구에 맞아야 하지만, 특정 시스템 구성에 따라 이 값을 조정해야 할 수도 있습니다. 자세한 내용은 ["인터럽트 제한 조정" \[369\]](#)을 참조하십시오.

PCIe 버스를 지정하여 루트 도메인 만들기

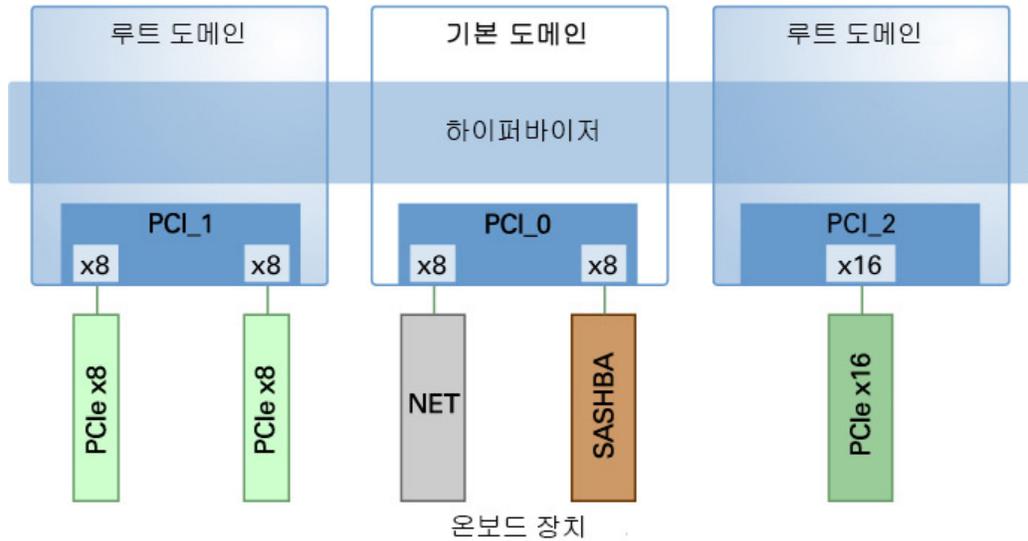
이 장에서는 PCIe 버스를 지정하여 루트 도메인을 만드는 방법에 대해 설명합니다.

PCIe 버스를 지정하여 루트 도메인 만들기

Oracle VM Server for SPARC 소프트웨어를 사용하여 도메인에 전체 PCIe 버스(루트 컴플렉스라고도 함)를 지정할 수 있습니다. 전체 PCIe 버스는 PCIe 버스 자체 및 관련된 모든 PCI 스위치와 장치로 구성됩니다. 서버에 있는 PCIe 버스는 `pci@400(pci_0)`과 같은 이름으로 식별됩니다. 전체 PCIe 버스로 구성된 I/O 도메인을 루트 도메인이라고도 합니다.

다음 다이어그램은 루트 컴플렉스가 3개(`pci_0`, `pci_1`, `pci_2`)인 시스템을 보여줍니다.

그림 2 루트 도메인에 PCIe 버스 지정



PCIe 버스로 만들 수 있는 최대 루트 도메인 수는 서버에서 사용 가능한 PCIe 버스 수에 따라 다릅니다. `ldm list-io`를 사용하여 시스템에서 사용 가능한 PCIe 버스 수를 확인합니다.

루트 도메인에 PCIe 버스를 지정하면 이 루트가 해당 버스의 모든 장치를 소유합니다. 해당 버스의 PCIe 끝점 장치를 다른 도메인에 지정할 수 있습니다.

서버가 처음에 Oracle VM Server for SPARC 환경에서 구성되거나 `factory-default` 구성을 사용 중인 경우 `primary` 도메인은 모든 물리적 장치 리소스에 대한 액세스 권한을 가집니다. 따라서 `primary` 도메인이 시스템에서 구성된 유일한 루트 도메인이며 모든 PCIe 버스를 소유합니다.

정적 PCIe 버스 지정

루트 도메인에 정적 PCIe 버스 지정 방법을 사용하려면 PCIe 버스를 지정/제거할 때 루트 도메인에서 지연된 재구성을 시작해야 합니다. 아직 PCIe 버스가 없는 도메인에 이 방법을 사용하려면 PCIe 버스를 지정하기 전에 도메인을 중지해야 합니다. 루트 도메인에서 구성 단계를 완료한 후에는 재부트해야 합니다. Oracle VM Server for SPARC 3.2 펌웨어가 시스템에 설치되지 않은 경우 또는 개별 도메인에 설치된 OS 버전이 동적 PCIe 버스 지정을 지원하지 않는 경우 정적 방법을 사용해야 합니다.

루트 도메인이 중지되거나 지연된 재구성 상태인 동안, 루트 도메인을 재부트하기 전에 `ldm add-io` 및 `ldm remove-io` 명령 중 하나 이상을 실행할 수 있습니다. 도메인 작동 중지 시간을 최소화하려면 PCIe 버스를 지정/제거하기 전에 사전 계획을 세우십시오.

- 루트 도메인의 경우 `primary`와 `비primary`에서 모두 지연된 재구성을 사용합니다. PCIe 버스를 추가/제거한 후에 루트 도메인을 재부트하여 변경사항을 적용합니다.

```
primary# ldm start-reconf root-domain
Add or remove the PCIe bus by using the ldm add-io or ldm remove-io command
primary# ldm stop -r domain-name
```

도메인이 이미 PCIe 버스를 소유한 경우에만 지연된 재구성을 사용할 수 있습니다.

- 비루트 도메인의 경우 도메인을 중지하고 PCIe 버스를 추가/제거합니다.

```
primary# ldm stop domain-name
Add or remove the PCIe bus by using the ldm add-io or ldm remove-io command
primary# ldm start domain-name
```

동적 PCIe 버스 지정

동적 PCIe 버스 지정 기능으로 루트 도메인에서 PCIe 버스를 동적으로 지정/제거할 수 있습니다.

시스템에 필요한 펌웨어 및 소프트웨어가 실행될 때 동적 PCIe 버스 지정 기능이 사용으로 설정됩니다. “[동적 PCIe 버스 지정 요구사항](#)” [67]을 참조하십시오. 시스템에 필요한 펌웨어 및 소프트웨어가 실행되지 않으면 `ldm add-io` 및 `ldm remove-io` 명령을 적절하게 실패합니다.

사용으로 설정된 경우, 루트 도메인을 중지하거나 루트 도메인을 지연된 재구성 모드에 넣지 않고도 `ldm add-io` 및 `ldm remove-io` 명령을 실행할 수 있습니다.

동적 PCIe 버스 지정 요구사항

동적 PCIe 버스 지정 기능은 루트 도메인에서 Oracle Solaris 11 OS를 실행하는 SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버, SPARC T7 시리즈 서버, SPARC S7 시리즈 서버 및 Fujitsu M10 서버에서 지원됩니다. SPARC M5 서버 및 SPARC M6 서버에서는 최소 9.4.2 버전의 시스템 펌웨어를 실행해야 하고, SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버에서는 최소 9.4.3을, SPARC S7 시리즈 서버에서는 릴리스된 모든 버전의 시스템 펌웨어를, Fujitsu M10 서버는 최소 XCP2240을 실행해야 합니다.

▼ PCIe 버스를 지정하여 루트 도메인을 만드는 방법

이 절차 예에서는 `primary` 도메인이 여러 버스를 소유한 초기 구성에서 새 루트 도메인을 만드는 방법을 보여줍니다. 기본적으로 `primary` 도메인은 시스템에 있는 모든 버스를 소유합니다. 이 예는 SPARC T4-2 서버용입니다. 다른 서버에서도 이 절차를 사용할 수 있습니다. 다른 서버에 대한 지침은 이 예의 지침과 약간 다를 수 있지만 이 예를 통해 기본 원칙을 확인할 수 있습니다.

부트 디스크와 기본 네트워크 인터페이스를 호스트하는 PCIe 버스를 `primary` 도메인에서 제거하지 않도록 하십시오.



주의 - 지원되는 서버의 모든 내부 디스크를 단일 PCIe 버스에 연결할 수 있습니다. 도메인이 내부 디스크에서 부트되는 경우 도메인에서 해당 버스를 제거하지 마십시오.

네트워크 포트나 `usbem` 장치와 같이 도메인이 사용하는 장치가 있는 버스를 제거하지 않도록 하십시오. 잘못된 버스를 제거하면 도메인이 필요한 장치에 액세스하지 못할 수 있으며 도메인을 사용하지 못할 수 있습니다. 도메인이 사용하는 장치가 있는 버스를 제거하려면 다른 버스에서 장치를 사용하도록 해당 도메인을 재구성하십시오. 예를 들어, 다른 PCIe 슬롯에서 다른 내장 네트워크 포트 또는 PCIe 카드를 사용하도록 도메인을 재구성해야 할 수 있습니다.

특정 SPARC 서버의 경우 USB, 그래픽 컨트롤러와 기타 장치가 있는 PCIe 버스를 제거할 수 있습니다. 그러나 해당 PCIe 버스를 다른 도메인에 추가할 수는 없습니다. 해당 PCIe 버스는 `primary` 도메인에만 추가할 수 있습니다.

이 예에서 `primary` 도메인은 ZFS 풀(`rpool`) 및 네트워크 인터페이스(`igb0`)만 사용합니다. `primary` 도메인이 보다 많은 장치를 사용하는 경우 각 장치에 대해 2-4단계를 반복하여 제거할 버스에 장치가 남아 있지 않도록 하십시오.

장치 경로(`pci@nnn`) 또는 익명(`pci_n`)을 사용하여 버스를 도메인에 추가하거나 도메인에서 제거할 수 있습니다. `ldm list-bindings primary` 또는 `ldm list -l -o physio primary` 명령은 다음을 보여줍니다.

- `pci@400`은 `pci_0`에 해당합니다.
- `pci@500`은 `pci_1`에 해당합니다.
- `pci@600`은 `pci_2`에 해당합니다.
- `pci@700`은 `pci_3`에 해당합니다.

1. `primary` 도메인이 2개 이상의 PCIe 버스를 소유하는지 확인합니다.

```
primary# ldm list-io
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
pci_0                               BUS   pci_0    primary
pci_1                               BUS   pci_1    primary
pci_2                               BUS   pci_2    primary
pci_3                               BUS   pci_3    primary
/SYS/MB/PCIE1                       PCIE  pci_0    primary EMP
```

```

/SYS/MB/SASHBA0      PCIE  pci_0  primary OCC
/SYS/MB/NET0         PCIE  pci_0  primary OCC
/SYS/MB/PCIE5        PCIE  pci_1  primary EMP
/SYS/MB/PCIE6        PCIE  pci_1  primary EMP
/SYS/MB/PCIE7        PCIE  pci_1  primary EMP
/SYS/MB/PCIE2        PCIE  pci_2  primary EMP
/SYS/MB/PCIE3        PCIE  pci_2  primary EMP
/SYS/MB/PCIE4        PCIE  pci_2  primary EMP
/SYS/MB/PCIE8        PCIE  pci_3  primary EMP
/SYS/MB/SASHBA1      PCIE  pci_3  primary OCC
/SYS/MB/NET2         PCIE  pci_3  primary OCC
/SYS/MB/NET0/IOVNET.PF0 PF    pci_0  primary
/SYS/MB/NET0/IOVNET.PF1 PF    pci_0  primary
/SYS/MB/NET2/IOVNET.PF0 PF    pci_3  primary
/SYS/MB/NET2/IOVNET.PF1 PF    pci_3  primary

```

2. 보존해야 할 부트 디스크의 장치 경로를 확인합니다.

- UFS 파일 시스템의 경우 `df /` 명령을 실행하여 부트 디스크의 장치 경로를 확인합니다.

```

primary# df /
/                               (/dev/dsk/c0t5000CCA03C138904d0s0):22755742 blocks 2225374 files

```

- ZFS 파일 시스템의 경우 먼저 `df /` 명령을 실행하여 풀 이름을 확인합니다. 그런 다음 `zpool status` 명령을 실행하여 부트 디스크의 장치 경로를 확인합니다.

```

primary# zpool status rpool
pool: rpool
state: ONLINE
scan: none requested
config:

      NAME                                STATE      READ WRITE CKSUM
      rpool                                ONLINE    0     0     0
      c0t5000CCA03C138904d0s0             ONLINE    0     0     0

```

3. 시스템의 부트 디스크에 대한 정보를 얻습니다.

- Solaris I/O 다중 경로로 관리되는 디스크의 경우 `mpathadm` 명령을 사용하여 연결된 부트 디스크에서 PCIe 버스를 확인합니다.

a. 디스크가 연결된 개시자 포트를 찾습니다.

```

primary# mpathadm show lu /dev/rdisk/c0t5000CCA03C138904d0s0
Logical Unit: /dev/rdisk/c0t5000CCA03C138904d0s2
mpath-support: libmpscsi_vhci.so
Vendor: HITACHI
Product: H106030SDSUN300G
Revision: A2B0
Name Type: unknown type
Name: 5000cca03c138904
Asymmetric: no
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:

```

```
Initiator Port Name: w50800200014100c8
Target Port Name: w5000cca03c138905
Override Path: NA
Path State: OK
Disabled: no
```

```
Target Ports:
Name: w5000cca03c138905
Relative ID: 0
```

b. 개시자 포트가 있는 PCIe 버스를 확인합니다.

```
primary# mpathadm show initiator-port w50800200014100c8
Initiator Port: w50800200014100c8
Transport Type: unknown
OS Device File: /devices/pci@400/pci@2/pci@0/pci@e/scsi@0/ipro@1
```

- Solaris I/O 다중 경로로 관리되지 않는 디스크의 경우 `ls -l` 명령을 사용하여 블록 장치가 연결된 물리적 장치를 확인합니다.

다음 예에서는 블록 장치 `c1t0d0s0`을 사용합니다.

```
primary# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root 49 Oct 1 10:39 /dev/dsk/c0t1d0s0 ->
../devices/pci@400/pci@0/pci@1/scsi@0/sd@1,0:a
```

이 예에서 `primary` 도메인의 부트 디스크에 대한 물리적 장치가 `pci@400` 버스에 연결됩니다.

4. 시스템이 사용하는 네트워크 인터페이스를 확인합니다.

`ifconfig` 명령을 사용하여 “플럼된” 기본 네트워크 인터페이스를 식별합니다. IP 프로토콜에서 장치를 사용할 수 있도록 연결된 인터페이스의 스트림이 설정됩니다.

```
primary# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
inet 127.0.0.1 netmask ff000000
net0: flags=1004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4> mtu 1500 index 3
inet 10.129.241.135 netmask ffffffff broadcast 10.129.241.255
ether 0:10:e0:e:f1:78
```

```
primary# dladm show-phys net0
LINK          MEDIA          STATE          SPEED          DUPLEX         DEVICE
net0          Ethernet      up            1000          full          igb0
```

5. 네트워크 인터페이스가 연결된 물리적 장치를 확인합니다.

다음 명령은 `igb0` 네트워크 인터페이스를 사용합니다.

```
primary# ls -l /dev/igb0
lrwxrwxrwx 1 root root 46 Oct 1 10:39 /dev/igb0 ->
../devices/pci@500/pci@0/pci@c/network@0:igb0
```

`ls -l /dev/usbecm` 명령도 수행합니다.

이 예에서 `primary` 도메인이 사용하는 네트워크 인터페이스에 대한 물리적 장치는 이전 `pci_1` 목록에 해당하는 버스 `pci@500`에 있습니다. 따라서 다른 2개의 버스 `pci_2(pci@600)` 및 `pci_3(pci@700)`은 `primary` 도메인이 사용하지 않으므로 다른 도메인에 지정해도 됩니다.

primary 도메인이 사용하는 네트워크 인터페이스가 다른 도메인에 지정할 버스에 있을 경우 다른 네트워크 인터페이스를 사용하도록 primary 도메인을 재구성합니다.

6. 부트 디스크 또는 네트워크 인터페이스를 포함하지 않는 버스를 primary 도메인에서 제거합니다.

이 예에서는 pci_2 버스가 primary 도메인에서 제거되고 있습니다.

■ 동적 방법:

pci_2 버스의 장치를 primary 도메인 OS에서 사용하지 않는지 확인합니다. 만일 사용 중이면 이 명령이 버스를 제거하지 못할 수 있습니다. 정적 방법을 사용하여 pci_2 버스를 강제로 제거합니다.

```
primary# ldm remove-io pci_2 primary
```

■ 정적 방법:

버스를 제거하기 전에 지연된 재구성을 시작해야 합니다.

```
primary# ldm start-reconf primary
primary# ldm remove-io pci_2 primary
primary# shutdown -y -g0 -i6
```

primary 도메인이 부트 디스크 및 네트워크 장치에 사용하는 버스는 다른 도메인에 지정할 수 없습니다. 기타 모든 버스는 다른 도메인에 지정할 수 있습니다. 이 예에서 pci@600은 primary 도메인에 사용되지 않으므로 다른 도메인에 재지정할 수 있습니다.

7. 도메인에 버스를 추가합니다.

이 예에서는 pci_2 버스를 ldg1 도메인에 추가합니다.

■ 동적 방법:

```
primary# ldm add-io pci_2 ldg1
```

■ 정적 방법:

버스를 추가하기 전에 도메인을 중지해야 합니다.

```
primary# ldm stop-domain ldg1
primary# ldm add-io pci_2 ldg1
primary# ldm start-domain ldg1
```

8. 서비스 프로세서에 이 구성을 저장합니다.

이 예에서 구성은 io-domain입니다.

```
primary# ldm add-config io-domain
```

이 구성 io-domain은 재부트 후 사용할 다음 구성으로도 설정됩니다.

9. primary 도메인에 여전히 올바른 버스가 지정되어 있으며 도메인 ldg1에 올바른 버스가 지정되었는지 확인합니다.

```

primary# ldm list-io
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
pci_0                               BUS   pci_0    primary
pci_1                               BUS   pci_1    primary
pci_2                               BUS   pci_2    ldg1
pci_3                               BUS   pci_3    primary
/SYS/MB/PCIE1                       PCIE  pci_0    primary  EMP
/SYS/MB/SASHBA0                     PCIE  pci_0    primary  OCC
/SYS/MB/NET0                         PCIE  pci_0    primary  OCC
/SYS/MB/PCIE5                       PCIE  pci_1    primary  EMP
/SYS/MB/PCIE6                       PCIE  pci_1    primary  EMP
/SYS/MB/PCIE7                       PCIE  pci_1    primary  EMP
/SYS/MB/PCIE2                       PCIE  pci_2    ldg1     EMP
/SYS/MB/PCIE3                       PCIE  pci_2    ldg1     EMP
/SYS/MB/PCIE4                       PCIE  pci_2    ldg1     EMP
/SYS/MB/PCIE8                       PCIE  pci_3    primary  EMP
/SYS/MB/SASHBA1                     PCIE  pci_3    primary  OCC
/SYS/MB/NET2                         PCIE  pci_3    primary  OCC
/SYS/MB/NET0/IOVNET.PF0             PF    pci_0    primary
/SYS/MB/NET0/IOVNET.PF1             PF    pci_0    primary
/SYS/MB/NET2/IOVNET.PF0             PF    pci_3    primary
/SYS/MB/NET2/IOVNET.PF1             PF    pci_3    primary

```

이 출력은 PCIe 버스 pci_0, pci_1 및 pci_3과 해당 장치가 primary 도메인에 지정되었음을 확인합니다. PCIe 버스 pci_2 및 해당 장치가 ldg1 도메인에 지정된 것도 확인할 수 있습니다.

PCIe SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기

이 장에서는 다음과 같은 PCIe SR-IOV 항목을 다룹니다.

- “SR-IOV 개요” [73]
- “SR-IOV 하드웨어 및 소프트웨어 요구사항” [76]
- “현재 SR-IOV 기능 제한 사항” [79]
- “정적 SR-IOV” [80]
- “동적 SR-IOV” [81]
- “I/O 가상화 사용으로 설정” [83]
- “PCIe SR-IOV 가상 기능 사용 계획” [84]
- “이더넷 SR-IOV 가상 기능 사용” [85]
- “InfiniBand SR-IOV 가상 기능 사용” [103]
- “광 섬유 채널 SR-IOV 가상 기능 사용” [115]
- “I/O 도메인 복원성” [128]
- “비복원형 I/O 도메인이 구성된 상태로 루트 도메인 재부트” [134]

SR-IOV 개요

주 - 루트 도메인은 다른 루트 도메인에 대한 종속성을 가질 수 없으므로 PCIe 버스를 소유하는 루트 도메인은 다른 루트 도메인에 지정된 SR-IOV 가상 기능 또는 PCIe 끝점 장치를 가질 수 없습니다. 하지만 PCIe 버스의 PCIe 끝점 장치 또는 가상 기능을 해당 버스를 소유하는 루트 도메인에 지정할 수는 있습니다.

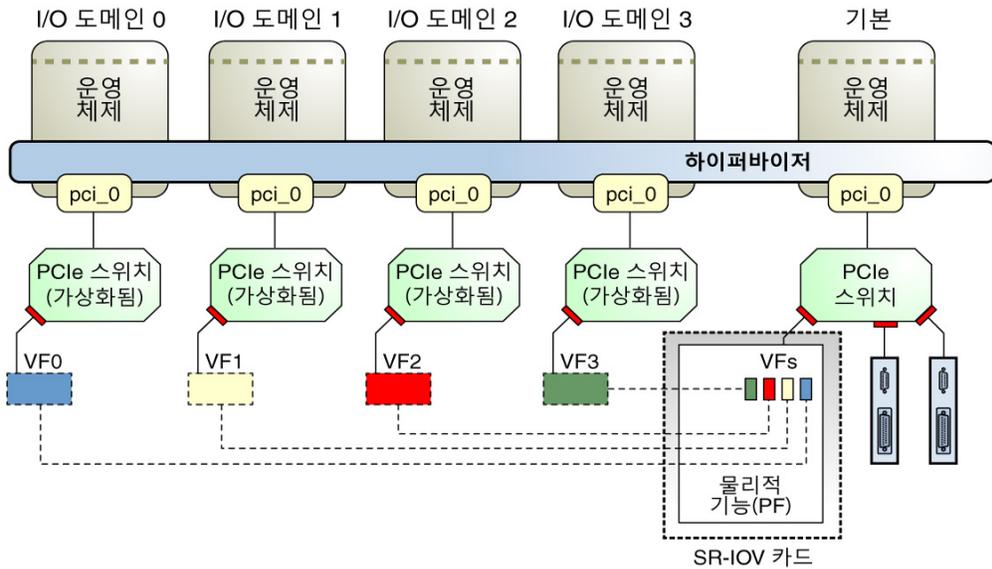
PCIe(Peripheral Component Interconnect Express) SR-IOV(단일 루트 I/O 가상화) 구현은 PCI-SIG가 정의한 표준의 버전 1.1을 기반으로 합니다. SR-IOV 표준은 가상 시스템에서 효율적으로 PCIe 장치를 공유할 수 있도록 하며 고유 성능과 유사한 I/O 성능이 발휘되도록 하드웨어에서 구현됩니다. SR-IOV 사양은 만들어진 새 장치를 통해 가상 시스템을 I/O 장치에 직접 연결할 수 있다는 점에서 새로운 표준을 정의합니다.

물리적 기능이라는 단일 I/O 리소스는 여러 가상 시스템에서 공유될 수 있습니다. 공유된 장치는 전용 리소스를 제공하며 공유된 공통 리소스를 사용하기도 합니다. 따라서 각 가상 시스템은 고유한 리소스에 대한 액세스 권한을 가집니다. 이로 인해 적절한 하드웨어 및 OS 지원을 통해 SR-IOV가 사용으로 설정된 이더넷 포트 등의 PCIe 장치는 각각 고유의 PCIe 구성 공간을 가진 별도의 다중 물리적 장치로 표시될 수 있습니다.

SR-IOV에 대한 자세한 내용은 [PCI-SIG 웹 사이트 \(http://www.pcisig.com/\)](http://www.pcisig.com/)를 참조하십시오.

다음 그림에서는 I/O 도메인에서 가상 기능과 물리적 기능 간의 관계를 보여줍니다.

그림 3 I/O 도메인에서 가상 기능과 물리적 기능 사용



SR-IOV의 기능 유형은 다음과 같습니다.

- 물리적 기능 - SR-IOV 사양에 정의된 SR-IOV 기능을 지원하는 PCI 기능입니다. 물리적 기능은 SR-IOV 기능 구조를 포함하며 SR-IOV 기능을 관리합니다. 물리 함수는 다른 PCIe 장치처럼 검색, 관리 및 조작할 수 있는 완전형 PCIe 함수입니다. 물리적 기능을 사용하여 PCIe 장치를 구성 및 제어할 수 있습니다.
- 가상 기능 - 물리적 기능과 연관된 PCI 기능입니다. 가상 기능은 하나 이상의 물리적 리소스를 물리적 기능 및 해당 물리적 기능과 연관된 가상 기능과 공유하는 경량형 PCIe 기능입니다. 물리적 기능과 달리 가상 기능은 고유의 동작만 구성할 수 있습니다.

각 SR-IOV 장치는 물리적 기능을 사용할 수 있으며 각 물리적 기능에는 최대 256개의 가상 기능이 연관될 수 있습니다. 이 수치는 SR-IOV 장치마다 다릅니다. 가상 기능은 물리적 기능에 의해 만들어집니다.

SR-IOV가 물리적 기능에서 사용으로 설정되면 물리적 기능의 버스, 장치 및 기능 번호가 각 가상 기능의 PCI 구성 공간에 액세스할 수 있습니다. 각 가상 기능에는 레지스터 세트 매핑에 사용되는 PCI 메모리 공간이 있습니다. 가상 기능 장치 드라이버는 기능이 사용으로 설정되도록 레지스터 세트에서 작동하며 가상 기능은 실제 PCI 장치로 표시됩니다. 만들어진 가상 기능은 I/O 도메인에 직접 지정할 수 있습니다. 이를 통해 가상 기능이 물리적 장치를 공유하고 CPU 및 하이퍼바이저 소프트웨어 오버헤드 없이 I/O를 수행할 수 있습니다.

환경에서 SR-IOV 기능을 사용하면 다음 이점을 얻을 수 있습니다.

- 성능 향상 및 대기 시간 감소 – 가상 시스템 환경에서 하드웨어에 직접 액세스합니다.
- 비용 절감 – 다음과 같이 자본 및 운영비가 절감됩니다.
 - 절전
 - 어댑터 수 감소
 - 케이블 연결 감소
 - 스위치 포트 감소

Oracle VM Server for SPARC SR-IOV 구현에는 정적 구성 방법과 동적 구성 방법이 포함되어 있습니다. 자세한 내용은 “정적 SR-IOV” [80] 및 “동적 SR-IOV” [81]를 참조하십시오.

Oracle VM Server for SPARC SR-IOV 기능을 통해 다음 작업을 수행할 수 있습니다.

- 지정된 물리적 기능에 가상 기능 만들기
- 물리적 기능에서 지정된 가상 기능 삭제
- 도메인에 가상 기능 지정
- 도메인에서 가상 기능 제거

SR-IOV 물리적 기능 장치에서 가상 기능을 만들거나 삭제하려면 먼저 해당 PCIe 버스에서 I/O 가상화를 사용으로 설정해야 합니다. `ldm set-io` 또는 `ldm add-io` 명령을 사용하여 `iov` 등록 정보를 `on`으로 설정할 수 있습니다. 또한 `ldm add-domain` 또는 `ldm set-domain` 명령을 사용하여 `rc-add-policy` 등록 정보를 `iov`로 설정할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

주 - SPARC M7 시리즈 서버, SPARC T7 시리즈 서버 및 Fujitsu M10 서버에서는 기본적으로 PCIe 버스에서 I/O 가상화가 사용으로 설정됩니다.

SR-IOV 가상 기능을 도메인에 지정하면 SR-IOV 물리적 기능 서비스를 제공하는 도메인에 암시적 종속성이 생깁니다. `ldm list-dependencies` 명령을 사용하여 해당 종속성을 보거나 이 SR-IOV 물리적 기능에 종속된 도메인을 볼 수 있습니다. “도메인 I/O 종속성 나열” [371]을 참조하십시오.

SR-IOV 하드웨어 및 소프트웨어 요구사항

동적 및 정적 PCIe SR-IOV 기능은 SPARC T4 서버, SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버 및 SPARC M7 시리즈 서버에서 지원됩니다. Fujitsu M10 플랫폼에서는 이더넷 장치에만 동적 기능이 지원되고, 다른 장치 유형은 정적 방법을 사용해야 합니다. SPARC T3 플랫폼은 정적 PCIe SR-IOV 기능만 지원됩니다.

■ 하드웨어 요구사항.

플랫폼에서 사용할 수 있는 카드를 확인하려면 해당 플랫폼의 하드웨어 설명서를 참조하십시오. 지원되는 PCIe 카드에 대한 최신 목록을 보려면 <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1>을 참조하십시오.

- **이더넷 SR-IOV.** SR-IOV 기능을 사용하기 위해서는 PCIe SR-IOV 플러그인 카드뿐만 아니라 온보드 PCIe SR-IOV 장치를 사용할 수 있습니다. 플랫폼 설명서에 명시적으로 언급되지 않는 한, 제공된 플랫폼의 모든 온보드 SR-IOV 장치가 지원됩니다.
- **InfiniBand SR-IOV.** InfiniBand 장치는 SPARC T4 서버, SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버 및 Fujitsu M10 서버에서 지원됩니다.
- **광 섬유 채널 SR-IOV.** 광 섬유 채널 장치는 SPARC T4 서버, SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버 및 Fujitsu M10 서버에서 지원됩니다.

Fujitsu M10 플랫폼에서 지원되는 장치의 최신 목록은 <http://www.fujitsu.com/global/services/computing/server/sparc/downloads/manual/>에서 해당 모델에 대한 제품 안내서에 나와 있는 *Fujitsu M10/SPARC M10 Systems PCI Card Installation Guide*를 참조하십시오.

■ 펌웨어 요구사항.

- **이더넷 SR-IOV.** 동적 SR-IOV 기능을 사용하려면 SPARC T4 서버에서 최소 8.4.0.a 버전의 시스템 펌웨어를 실행해야 합니다. SPARC T5 서버, SPARC M5 서버 및 SPARC M6 서버에서는 최소 9.1.0 버전의 시스템 펌웨어를 실행해야 합니다. SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버에서는 최소 9.4.3 버전의 시스템 펌웨어를 실행해야 하고, Fujitsu M10 서버에서는 최소 XCP2210 버전의 시스템 펌웨어를 실행해야 합니다. SPARC T3 서버는 정적 SR-IOV 기능만 지원됩니다.

SR-IOV 기능을 사용하려면 PCIe SR-IOV 장치에서 최소 3.01 버전의 펌웨어 장치를 실행해야 합니다. 다음 단계에 따라 Sun Dual 10-Gigabit Ethernet SFP+ PCIe 2.0 네트워크 어댑터의 펌웨어를 업데이트하십시오.

1. 장치에서 FCode 버전을 업그레이드해야 하는지 여부를 확인합니다.

ok 프롬프트에서 다음 명령을 수행합니다.

```
{0} ok cd path-to-device
{0} ok .properties
```

출력에서 `version` 값은 다음 중 하나여야 합니다.

LP	Sun Dual 10GbE SFP+ PCIe 2.0 LP FCode 3.01 4/2/2012
PEM	Sun Dual 10GbE SFP+ PCIe 2.0 EM FCode 3.01 4/2/2012
FEM	Sun Dual 10GbE SFP+ PCIe 2.0 FEM FCode 3.01 4/2/2012

2. [My Oracle Support \(https://support.oracle.com/CSP/ui/flash.html#tab=PatchHomePage\(page=PatchHomePage&id=h0wvdx6\(\)\)\)](https://support.oracle.com/CSP/ui/flash.html#tab=PatchHomePage(page=PatchHomePage&id=h0wvdx6()))에서 패치 ID 13932765를 다운로드합니다.

3. 패치를 설치합니다.

패치 패키지에는 도구를 사용하여 업그레이드를 수행하는 방법을 설명하는 설명서가 포함됩니다.

- **InfiniBand SR-IOV.** 이 기능을 사용하려면 시스템에서 최소한 다음 버전의 시스템 펌웨어를 실행해야 합니다.
 - **SPARC T4** 서버 – 8.4
 - **SPARC T5** 서버 – 9.1.0.x
 - **SPARC T7** 시리즈 서버 – 9.4.3
 - **SPARC M5** 및 **SPARC M6** 서버 – 9.1.0.x
 - **SPARC M7** 시리즈 서버 – 9.4.3
 - **Fujitsu M10** 서버 – XCP2210

InfiniBand SR-IOV 장치로 이중 40Gbit(4x) InfiniBand 호스트 채널 어댑터 M2를 지원하려면 카드나 익스프레스 모듈에서 최소 2.11.2010 버전의 펌웨어를 실행해야 합니다. 다음 패치를 설치하면 이 버전의 펌웨어를 얻을 수 있습니다.

- **로우 프로파일(X4242A)** – 패치 ID 16340059
- **익스프레스 모듈(X4243A)** – 패치 ID 16340042

Oracle Solaris 11.1 `fwflash` 명령을 사용하여 `primary` 도메인에서 펌웨어를 나열하고 업데이트할 수 있습니다. 현재 펌웨어 버전을 나열하려면 `fwflash -lc IB` 명령을 사용합니다. 펌웨어를 업데이트하려면 `fwflash -f firmware-file -d device` 명령을 사용합니다. [fwflash\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

InfiniBand SR-IOV를 사용하려면 InfiniBand 스위치에 최소 펌웨어 버전 2.1.2가 실행되는지 확인하십시오. 다음 패치를 설치하면 이 버전의 펌웨어를 얻을 수 있습니다.

- **Sun Datacenter InfiniBand Switch 36(X2821A-Z)** – 패치 ID 16221424
- **Sun Network QDR InfiniBand GatewaySwitch(X2826A-Z)** – 패치 ID 16221538

펌웨어 업데이트 방법에 대한 자세한 내용은 InfiniBand 스위치 설명서를 참조하십시오.

- 광 섬유 채널 **SR-IOV**. 이 기능을 사용하려면 시스템에서 최소한 다음 버전의 시스템 펌웨어를 실행해야 합니다.
 - **SPARC T4** 서버 – 8.4.2.c
 - **SPARC T5** 서버 – 9.1.2.d
 - **SPARC T7** 시리즈 서버 – 9.4.3
 - **SPARC M5** 서버 – 9.1.2.d
 - **SPARC M6** 서버 – 9.1.2.d
 - **SPARC M7** 시리즈 서버 – 9.4.3
 - **Fujitsu M10** 서버 – XCP2210

광 섬유 채널 SR-IOV 기능을 사용으로 설정하려면 Sun Storage 16Gb 광 섬유 채널 범용 HBA(Emulex)의 펌웨어는 최소 1.1.60.1 개정이어야 합니다. 설치 지침은 펌웨어와 함께 제공됩니다.

주 - SR-IOV 기능을 사용하려는 경우 필요한 최소 레벨을 충족하도록 펌웨어를 업데이트해야 합니다.

- **소프트웨어 요구사항.**
 - 이더넷 **SR-IOV**. SR-IOV 기능을 사용하려면 모든 도메인에서 최소한 Oracle Solaris 11.1 SRU 10 OS를 실행해야 합니다.
 - **InfiniBand SR-IOV**. 다음 도메인에서 지원되는 Oracle Solaris OS를 실행해야 합니다.
 - primary 도메인 또는 비primary 루트 도메인에서 최소 Oracle Solaris 11.1 SRU 10 OS를 실행해야 합니다.
 - I/O 도메인은 최소 Oracle Solaris 11.1 SRU 10 OS를 실행해야 합니다.
 - InfiniBand SR-IOV 물리적 기능을 기반으로 가상 기능을 구성하려는 루트 도메인에서 /etc/system 파일을 업데이트합니다.

```
set ldc:ldc_mactable_entries = 0x20000
```

/etc/system 등록 정보 값을 정확히 만들거나 업데이트하는 방법은 “/etc/system 파일에서 등록 정보 값 업데이트” [352]를 참조하십시오.

가상 기능을 추가하려는 I/O 도메인에서 /etc/system 파일을 업데이트합니다.

```
set rds3:rds3_fmr_pool_size = 16384
```

- 광 섬유 채널 **SR-IOV**. SR-IOV 기능을 사용하려면 모든 도메인에서 최소한 Oracle Solaris 11.1 SRU 17 OS를 실행해야 합니다.

정적 및 동적 SR-IOV 소프트웨어 요구사항에 대한 자세한 내용은 다음을 참조하십시오.

- “정적 SR-IOV 소프트웨어 요구사항” [80]
- “동적 SR-IOV 소프트웨어 요구사항” [81]

클래스별 SR-IOV 하드웨어 요구사항에 대한 자세한 내용은 다음을 참조하십시오.

- “이더넷 SR-IOV 하드웨어 요구사항” [85]
- “InfiniBand SR-IOV 하드웨어 요구사항” [103]
- “광 섬유 채널 SR-IOV 하드웨어 요구사항” [115]

현재 SR-IOV 기능 제한 사항

SR-IOV 기능에는 다음 제한 사항이 있습니다.

- 연관된 루트 도메인이 실행 중이 아니면 I/O 도메인을 시작할 수 없습니다.
- 마이그레이션은 SR-IOV 물리적 기능 또는 SR-IOV 가상 기능이 하나 이상 지정된 도메인에 대해 사용 안함으로 설정됩니다.
- 물리적 기능에 대해 마지막으로 만들어진 가상 기능만 삭제할 수 있습니다. 따라서 3개의 가상 기능을 만들 경우 삭제할 수 있는 첫번째 가상 기능은 세번째 가상 기능입니다.
- DIO(직접 I/O) 기능을 사용하여 도메인에 SR-IOV 카드를 지정할 경우 해당 카드에 대해 SR-IOV 기능이 사용으로 설정되지 않습니다.
- 지원되는 SPARC T-Series 서버 및 SPARC M-Series 서버에서 특정 PCIe 버스의 PCIe 끝점 장치 및 SR-IOV 가상 기능은 최대 15개의 도메인에 지정할 수 있습니다. On SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버에서는 특정 PCIe 버스의 PCIe 끝점 장치 및 SR-IOV 가상 기능을 최대 31개의 도메인에 지정할 수 있습니다. Fujitsu M10 서버에서는 특정 PCIe 버스의 PCIe 끝점 장치 및 SR-IOV 가상 기능을 최대 24개의 도메인에 지정할 수 있습니다. PCIe 리소스(예: 각 PCIe 버스에 대한 인터럽트 벡터)는 루트 도메인 및 I/O 도메인에서 배분됩니다. 이에 따라 특정 I/O 도메인에 지정할 수 있는 장치 수도 제한됩니다. 동일한 I/O 도메인에 많은 수의 가상 기능을 지정하지 않아야 합니다. SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버에는 인터럽트 제한이 없습니다. SR-IOV 관련 문제에 대한 설명은 [Oracle VM Server for SPARC 3.4 릴리스 노트](#)를 참조하십시오.
- 루트 도메인은 PCIe 버스의 소유자이며 버스 초기화 및 관리를 담당합니다. 루트 도메인은 활성 상태여야 하며 SR-IOV 기능을 지원하는 Oracle Solaris OS 버전을 실행 중이어야 합니다. 루트 도메인을 종료, 정지 또는 재부트하면 PCIe 버스에 대한 액세스가 중단됩니다. PCIe 버스를 사용할 수 없는 경우 해당 버스의 PCIe 장치가 영향을 받아 이러한 장치를 사용하지 못할 수 있습니다.

PCIe SR-IOV 가상 기능이 있는 I/O 도메인이 실행 중인 동안 루트 도메인이 재부트되면 해당 I/O 도메인의 동작을 예측할 수 없습니다. 예를 들어, 재부트 도중 또는 재부트 후 PCIe 끝점 장치가 있는 I/O 도메인에 패닉이 발생할 수 있습니다. 루트 도메인을 재부트할 때 수동으로 각 도메인을 중지하고 시작해야 합니다.

I/O 도메인이 복원형인 경우 PCIe 버스 소유자인 루트 도메인을 사용할 수 없게 되더라도 작동을 계속할 수 있습니다. “[I/O 도메인 복원성](#)” [128]을 참조하십시오.

- SPARC T5 및 SPARC M6 플랫폼을 비롯한 SPARC 시스템은 한정된 수의 인터럽트를 제공하므로 Oracle Solaris는 각 장치가 사용할 수 있는 인터럽트 수를 제한합니다. 기본 제한은 표준 시스템 구성 요구에 맞아야 하지만, 특정 시스템 구성에 따라 이 값을 조정해야 할 수도 있습니다. 자세한 내용은 “[인터럽트 제한 조정](#)” [369]을 참조하십시오.

정적 SR-IOV

정적 SR-IOV 방법을 사용하려면 SR-IOV 작업을 수행하는 동안 루트 도메인이 지연된 재구성 상태이거나 I/O 도메인을 중지해야 합니다. 루트 도메인에서 구성 단계를 완료한 후에는 재부트해야 합니다. Oracle VM Server for SPARC 3.1 펌웨어가 시스템에 설치되지 않은 경우 또는 개별 도메인에 설치된 OS 버전이 동적 SR-IOV를 지원하지 않는 경우 이 방법을 사용해야 합니다.

SR-IOV 가상 기능을 만들거나 삭제하려면 먼저 루트 도메인에서 지연된 재구성을 시작해야 합니다. 그런 다음 `ldm create-vf` 및 `ldm destroy-vf` 명령 중 하나 이상을 실행하여 가상 기능을 구성할 수 있습니다. 마지막으로 루트 도메인을 재부트하십시오. 다음 명령은 비`primary` 루트 도메인에 가상 기능을 만드는 방법을 보여줍니다.

```
primary# ldm start-reconf root-domain-name
primary# ldm create-vf pf-name
primary# ldm stop-domain -r root-domain-name

primary# shutdown -i6 -g0 -y
```

정적으로 가상 기능을 게스트 도메인에 추가하거나 게스트 도메인에서 제거하려면 먼저 게스트 도메인을 중지해야 합니다. 그런 다음 `ldm add-io` 및 `ldm remove-io` 명령을 수행하여 가상 기능을 구성하십시오. 변경이 완료되면 도메인을 시작하십시오. 다음 명령은 이 방법으로 가상 기능을 지정하는 방법을 보여줍니다.

```
primary# ldm stop guest-domain
primary# ldm add-io vf-name guest-domain
primary# ldm start guest-domain
```

가상 기능을 게스트 도메인 대신 루트 도메인에 추가하거나 루트 도메인에서 제거할 수도 있습니다. SR-IOV 가상 기능을 루트 도메인에 추가하거나 루트 도메인에서 제거하려면 먼저 루트 도메인에서 지연된 재구성을 시작하십시오. 그런 다음 `ldm add-io` 및 `ldm remove-io` 명령 중 하나 이상을 실행할 수 있습니다. 마지막으로 루트 도메인을 재부트하십시오.

도메인 작동 중지 시간을 최소화하려면 가상 기능을 구성하기 전에 사전 계획을 세우십시오.

주 - InfiniBand SR-IOV 장치는 정적 SR-IOV에서만 지원됩니다.

정적 SR-IOV 소프트웨어 요구사항

SR-IOV 하드웨어 및 소프트웨어 요구사항에 대한 자세한 내용은 <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1>을 참조하십시오.

`ldm set-io` 또는 `ldm add-io` 명령을 사용하여 `iov` 등록 정보를 `on`으로 설정할 수 있습니다. 또한 `ldm add-domain` 또는 `ldm set-domain` 명령을 사용하여 `rc-add-policy` 등록 정보를 `iov`로 설정할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

루트 도메인을 재부트하면 SR-IOV가 영향을 받으므로 루트 도메인에 대한 SR-IOV 관련 변경사항 적용을 최대화하고 루트 도메인 재부트를 최소화하도록 신중하게 직접 I/O 구성 변경을 계획해야 합니다.

동적 SR-IOV

동적 SR-IOV 기능을 사용하면 다음과 같은 정적 SR-IOV 요구사항을 충족할 필요가 없습니다.

- **루트 도메인.** 루트 도메인에서 지연된 재구성을 시작하고, 가상 기능을 만들거나 삭제하고, 루트 도메인을 재부트합니다.
- **I/O 도메인.** I/O 도메인을 중지하고, 가상 기능을 추가하거나 제거하고, I/O 도메인을 시작합니다.

동적 SR-IOV를 사용하면 루트 도메인에서 지연된 재구성을 시작하지 않고도 동적으로 가상 기능을 만들거나 삭제할 수 있습니다. 또한 도메인을 중지하지 않고도 동적으로 가상 기능을 I/O 도메인에 추가하거나 I/O 도메인에서 제거할 수 있습니다. Logical Domains Manager는 Logical Domains 에이전트 및 Oracle Solaris I/O 가상화 프레임워크와 통신하여 이러한 변경사항을 동적으로 적용합니다.

동적 SR-IOV 소프트웨어 요구사항

필요한 PCIe SR-IOV 소프트웨어 및 펌웨어 버전에 대한 자세한 내용은 [“SR-IOV 하드웨어 및 소프트웨어 요구사항” \[76\]](#)을 참조하십시오.

주 - 시스템이 동적 SR-IOV 소프트웨어 및 펌웨어 요구사항을 충족하지 않을 경우 정적 SR-IOV 방법을 사용하여 SR-IOV 관련 작업을 수행해야 합니다. [“정적 SR-IOV” \[80\]](#)를 참조하십시오.

동적 SR-IOV 구성 요구사항

동적으로 가상 기능을 만들거나 삭제하려면 다음 조건을 충족해야 합니다.

- 가상 기능 구성을 시작하기 전에 PCIe 버스에 대한 I/O 가상화가 사용으로 설정되어 있어야 합니다.
- 루트 도메인 및 I/O 도메인에서 실행되는 OS가 최소한 Oracle Solaris 11.1 SRU 10 OS여야 합니다.

- 물리적 기능 장치가 OS에서 구성되어 있지 않거나 다중 경로 구성 상태여야 합니다. 예를 들어, 이더넷 SR-IOV 장치를 연결 해제하거나 IPMP 또는 통합에 포함시켜 가상 기능을 성공적으로 만들거나 삭제할 수 있습니다.
가상 기능을 만들거나 삭제하는 작업을 수행하려면 물리적 기능 장치 드라이버를 오프라인 상태와 온라인 상태 간에 토글해야 합니다. 다중 경로 구성에서는 해당 상태 간에 장치 드라이버를 토글할 수 있습니다.
- I/O 도메인에서 가상 기능을 제거하기 전에 가상 기능이 사용되고 있지 않거나 다중 경로 구성 상태가 아니어야 합니다. 예를 들어, SR-IOV 가상 기능을 연결 해제하거나 IPMP 구성에서 사용하지 않을 수 있습니다.

주 - 현재 다중 경로 구현은 가상 기능을 지원하지 않으므로 이더넷 SR-IOV 가상 기능에 대한 통합을 사용할 수 없습니다.

모든 가상 기능을 삭제하고 슬롯을 루트 도메인으로 반환해도 루트 컴플렉스 리소스가 복원되지 않음

주 - 이 절은 SPARC M6 시리즈 서버를 비롯한 SPARC T5 시리즈 서버와 같은 서버에 적용됩니다.

모든 가상 기능을 삭제하고 슬롯을 루트 도메인에 반환한 후에도 루트 컴플렉스의 리소스가 복원되지 않습니다.

복구: 루트 컴플렉스와 연관된 모든 가상 I/O 리소스를 해당 루트 도메인에 반환합니다.

먼저 컨트롤 도메인을 지연된 재구성에 배치합니다.

```
primary# ldm start-reconf primary
```

모든 하위 PCIe 슬롯을 pci_0 버스를 소유한 루트 도메인에 반환합니다. 그런 다음 pci_0 버스에서 모든 하위 가상 기능을 제거하고 삭제합니다.

마지막으로 pci_0 버스에 iov=off를 설정하고 루트 도메인을 재부트합니다.

```
primary# ldm set-io iov=off pci_0
primary# shutdown -y -g 10
```

임시해결책: 특정 PCIe 버스에 대해 iov 옵션을 off로 설정합니다.

```
primary# ldm start-reconf primary
primary# ldm set-io iov=off pci_0
```

I/O 가상화 사용으로 설정

SR-IOV 가상 기능을 구성하려면 루트 도메인이 지연된 재구성 상태인 동안 PCIe 버스에 대한 I/O 가상화를 사용으로 설정해야 합니다. 이 변경사항이 적용되도록 도메인을 재부트하십시오.

주 - SPARC M7 시리즈 서버, SPARC T7 시리즈 서버 및 Fujitsu M10 서버에서는 기본적으로 PCIe 버스에서 I/O 가상화가 사용으로 설정됩니다.

▼ PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법

이 절차는 루트 컴플렉스당 한 번만 수행해야 합니다. 루트 컴플렉스는 동일한 SP 구성의 일부로 실행 중이어야 합니다.

1. 루트 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

2. PCIe 버스에 대한 I/O 가상화 작업을 사용으로 설정합니다.

물리적 기능을 가진 버스에 대한 I/O 가상화가 사용으로 설정되지 않은 경우에만 이 단계를 수행합니다.

다음 명령 중 하나를 실행합니다.

- 지정된 PCIe 버스가 루트 도메인에 이미 지정된 경우 I/O 가상화를 사용으로 설정합니다.

```
primary# ldm set-io iov=on bus
```

- 루트 도메인에 PCIe 버스를 추가하는 동안 I/O 가상화를 사용으로 설정합니다.

```
primary# ldm add-io iov=on bus
```

3. 루트 도메인을 재부트합니다.

다음 명령 중 하나를 실행합니다.

- 비primary 루트 도메인을 재부트합니다.

```
primary# ldm stop-domain -r root-domain
```

- primary 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

PCIe SR-IOV 가상 기능 사용 계획

사전 계획을 통해 구성에서 가상 기능을 사용할 방법을 결정하십시오. 현재 및 향후 구성 요구 사항을 충족할 SR-IOV 장치의 가상 기능을 결정하십시오.

정적 방법 사용 시 필요한 I/O 가상화를 사용으로 설정하지 않은 경우 이 단계를 가상 기능 만들기 단계와 결합하십시오. 해당 단계를 결합하면 루트 도메인을 한 번만 재부트해야 합니다.

동적 SR-IOV를 사용할 수 있는 경우에도 모든 가상 기능을 동시에 만드는 것이 좋습니다. I/O 도메인에 지정된 후에는 동적으로 만들지 못할 수도 있기 때문입니다.

정적 SR-IOV의 경우 계획을 통해 각각 I/O 도메인에 부정적인 영향을 끼칠 수 있는 다중 루트 도메인 재부트를 방지할 수 있습니다.

I/O 도메인에 대한 자세한 내용은 [“일반적인 I/O 도메인 만들기 지침” \[64\]](#)을 참조하십시오.

SR-IOV 가상 기능 구성 및 지정을 계획하고 수행하려면 다음과 같은 일반 단계를 사용하십시오.

1. 시스템에서 사용 가능하며 요구 사항에 가장 적합한 PCIe SR-IOV 물리적 기능을 확인합니다.

다음 명령을 사용하여 필요한 정보를 식별합니다.

<code>ldm list-io</code>	사용 가능한 SR-IOV 물리적 기능 장치를 식별합니다.
<code>prtdiag -v</code>	사용 가능한 PCIe SR-IOV 카드 및 내장 장치를 식별합니다.
<code>ldm list-io -l pf-name</code>	지정된 물리적 기능에 대한 추가 정보(예: 장치에서 지원하는 최대 가상 기능 수)를 식별합니다.
<code>ldm list-io -d pf-name</code>	장치에서 지원하는 장치 관련 등록 정보를 식별합니다. “고급 SR-IOV 항목: 이더넷 SR-IOV” [97] 를 참조하십시오.

2. PCIe 버스에 대한 I/O 가상화 작업을 사용으로 설정합니다.

[PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법 \[83\]](#)을 참조하십시오.

3. 지정된 SR-IOV 물리적 기능에 필요한 수의 가상 기능을 만듭니다.

다음 명령을 사용하여 물리적 기능에 대한 가상 기능을 만듭니다.

```
primary# ldm create-vf -n max pf-name
```

자세한 내용은 [이더넷 SR-IOV 가상 기능을 만드는 방법 \[86\]](#), [InfiniBand 가상 기능을 만드는 방법 \[103\]](#) 및 [광 섬유 채널 SR-IOV 가상 기능을 만드는 방법 \[118\]](#)을 참조하십시오.

4. `ldm add-config` 명령을 사용하여 SP에 구성을 저장합니다.

자세한 내용은 [I/O 도메인에 이더넷 SR-IOV 가상 기능을 추가하는 방법 \[95\]](#), [I/O 도메인에 InfiniBand 가상 기능을 추가하는 방법 \[107\]](#) 및 [I/O 도메인에 광 섬유 채널 SR-IOV 가상 기능을 추가하는 방법 \[125\]](#)을 참조하십시오.

이더넷 SR-IOV 가상 기능 사용

정적 및 동적 SR-IOV 방법을 모두 사용하여 이더넷 SR-IOV 장치를 관리할 수 있습니다.



주의 - SR-IOV를 지원하는 일부 Intel 네트워크 어댑터를 사용할 경우 가상 기능이 악의적인 동작의 대상이 될 수 있습니다. 소프트웨어에서 생성된 예상치 않은 프레임으로 인해 호스트와 가상 스위치 간 트래픽이 감소할 수 있는데, 이 동작이 성능에 부정적인 영향을 미칠 수 있습니다.

VLAN 태그 지정을 사용하여 예상치 않은 악의적인 프레임을 삭제하도록 모든 SR-IOV 사용 포트를 구성하십시오.

- 물리적 기능 및 연관된 가상 기능에서 VLAN 태그 지정을 구성하려면 다음 명령을 사용하십시오.

```
ldm create-vf [pvid=pvid] [vid=vid1,vid2,...>] net-pf-name
```

- 기존 가상 기능에서 VLAN 태그 지정을 구성하려면 다음 명령을 사용하십시오.

```
ldm set-io [pvid=[pvid]] [vid=[vid1,vid2,...]] net-vf-name
```

I/O 도메인에서 VLAN 인터페이스를 만드는 방법은 [Oracle VM Server for SPARC 3.4 관리 설명서](#)의 “[VLAN 태그 지정 사용](#)”을 참조하십시오.

이더넷 SR-IOV 하드웨어 요구사항

필요한 PCIe 이더넷 SR-IOV 하드웨어에 대한 자세한 내용은 “[SR-IOV 하드웨어 및 소프트웨어 요구사항](#)” [76]을 참조하십시오.

이더넷 SR-IOV 제한 사항

이 릴리스에서 이더넷 SR-IOV 기능의 제한 사항은 다음과 같습니다.

- `pvid` 또는 `vid` 등록 정보를 설정하여 가상 기능의 VLAN 구성을 사용으로 설정할 수 있습니다. 이러한 가상 기능 등록 정보는 동시에 설정할 수 없습니다.

- SR-IOV 가상 기능을 가상 스위치용 백엔드 장치로 사용할 수 없습니다.

이더넷 SR-IOV 가상 기능 사용 계획

동적으로 가상 기능을 만드는 경우 물리적 기능이 다중 경로를 사용하거나 연결되어 있지 않은지 확인하십시오.

다중 경로를 사용할 수 없거나 물리적 기능을 연결해야 하는 경우 정적 방법을 사용하여 가상 기능을 만드십시오. [“정적 SR-IOV” \[80\]](#)를 참조하십시오.

이더넷 장치 관련 등록 정보 및 네트워크 관련 등록 정보

`ldm create-vf` 명령을 사용하여 가상 기능의 장치 관련 및 네트워크 관련 등록 정보를 설정합니다. `unicast-slots`는 장치 관련 등록 정보입니다. `mac-addr`, `alt-mac-addr`s, `mtu`, `pvid` 및 `vid`는 네트워크 관련 등록 정보입니다.

`mac-addr`, `alt-mac-addr`s 및 `mtu` 네트워크 관련 등록 정보는 지연된 재구성 상태에서 가상 기능이 `primary` 도메인에 지정된 경우에만 변경할 수 있습니다.

다음과 같이 가상 기능이 지정되면 해당 등록 정보를 변경하려는 시도가 실패합니다.

- 활성 I/O 도메인에 가상 기능이 지정된 경우: 소유 도메인이 비활성 또는 바인드 상태일 때 변경을 수행해야 하므로 등록 정보 변경 요청이 거부됩니다.
- 비`primary` 도메인에 가상 기능이 지정되고 지연된 재구성이 이미 적용된 경우: 오류 메시지와 함께 등록 정보 변경 요청이 실패합니다.

`pvid` 및 `vid` 네트워크 관련 등록 정보는 제한 없이 변경할 수 있습니다.

이더넷 가상 기능 만들기

이 절에서는 동적으로 가상 기능을 만들고 삭제하는 방법에 대해 설명합니다. 동적 방법을 사용하여 해당 작업을 수행할 수 없을 경우 가상 기능을 만들거나 삭제하기 전에 루트 도메인에서 지연된 재구성을 시작하십시오.

▼ 이더넷 SR-IOV 가상 기능을 만드는 방법

이 동적 방법을 사용할 수 없는 경우 정적 방법을 대신 사용하십시오. [“정적 SR-IOV” \[80\]](#)를 참조하십시오.

1. 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

물리적 기능의 이름에는 PCIe SR-IOV 카드 또는 내장 장치에 대한 위치 정보가 포함되어 있습니다.

2. 물리적 기능을 가진 버스에 대한 I/O 가상화가 아직 사용으로 설정되지 않은 경우 사용으로 설정합니다.

물리적 기능을 가진 버스에 대한 I/O 가상화가 사용으로 설정되지 않은 경우에만 이 단계를 수행합니다.

PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법 [83]을 참조하십시오.

3. 이더넷 물리적 기능에서 동적이거나 정적으로 단일 가상 기능 또는 여러 가상 기능을 만듭니다. 하나 이상의 가상 기능을 만든 후 게스트 도메인에 지정할 수 있습니다.

■ 동적 방법:

- 물리적 기능에서 모두 동시에 여러 가상 기능을 만들려면 다음 명령을 사용합니다.

```
primary# ldm create-vf -n number | max pf-name
```

`ldm create-vf -n max` 명령을 사용하여 한 번에 해당 물리적 기능에 대한 모든 가상 기능을 만듭니다.



주의 - 시스템에서 Intel 10Gbit 이더넷 카드를 사용하는 경우 각 물리적 기능에서 31개 이하의 가상 기능을 만들어 성능을 최대화합니다.

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

- 물리적 기능에서 하나의 가상 기능을 만들려면 다음 명령을 사용합니다.

```
ldm create-vf [mac-addr=num] [alt-mac-addr=[auto|num1, [auto|num2, ...]]
[pvid=pvid] [vid=vid1, vid2, ...] [mtu=size] [name=value...] pf-name
```

주 - 명시적으로 지정되지 않은 경우 네트워크 장치에 대한 MAC 주소가 자동으로 할당됩니다.

이 명령을 사용하여 해당 물리적 기능에 대해 하나의 가상 기능을 만듭니다. 이더넷 클래스 관련 등록 정보 값을 수동으로 지정할 수도 있습니다.

주 - 때때로 OS가 IOV 장치를 프로브하는 동안 새로 만든 가상 기능을 즉시 사용하지 못할 수도 있습니다. `ldm list-io` 명령을 사용하여 상위 물리적 기능과 그 하위 가상 기능에서 Status 열에 INV 값이 있는지 여부를 확인합니다. 이 값이 보이면 `ldm list-io` 출력에서 Status 열에 INV 값이 더 이상 표시되지 않을 때까지 기다렸다가(약 45초 정도) 물리적 기능이나 하위 가상 기능을 사용하십시오. 이 상태가 지속되면 장치에 문제가 있는 것입니다.

`primary`를 포함한 루트 도메인 재부트 직후나 `ldm create-vf` 또는 `ldm destroy-vf` 명령을 사용한 직후에 장치 상태가 INV로 표시될 수 있습니다.

■ 정적 방법:

a. 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

b. 이더넷 물리적 기능에서 단일 가상 기능 또는 여러 가상 기능을 만듭니다.

앞에 나온 동일한 명령을 사용하여 가상 기능을 동적으로 만듭니다.

c. 루트 도메인을 재부트합니다.

■ 비primary 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# ldm stop-domain -r root-domain
```

■ primary 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# shutdown -i6 -g0 -y
```

예 5 이더넷 물리적 기능에 대한 정보 표시

이 예는 `/SYS/MB/NET0/IOVNET.PF0` 물리적 기능에 대한 정보를 보여줍니다.

- 이 물리적 기능은 내장 `NET0` 네트워크 장치에서 온 것입니다.
- `IOVNET` 문자열은 물리적 기능이 네트워크 SR-IOV 장치임을 나타냅니다.

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN  STATUS
----                                     -
niu_0                                   NIU   niu_0    primary
niu_1                                   NIU   niu_1    primary
pci_0                                   BUS   pci_0    primary
pci_1                                   BUS   pci_1    primary
/SYS/MB/PCIE0                           PCIE  pci_0    primary  OCC
/SYS/MB/PCIE2                           PCIE  pci_0    primary  OCC
/SYS/MB/PCIE4                           PCIE  pci_0    primary  OCC
/SYS/MB/PCIE6                           PCIE  pci_0    primary  EMP
/SYS/MB/PCIE8                           PCIE  pci_0    primary  EMP
/SYS/MB/SASHBA                          PCIE  pci_0    primary  OCC
/SYS/MB/NET0                             PCIE  pci_0    primary  OCC
```

```

/SYS/MB/PCIE1          PCIE  pci_1  primary OCC
/SYS/MB/PCIE3          PCIE  pci_1  primary OCC
/SYS/MB/PCIE5          PCIE  pci_1  primary OCC
/SYS/MB/PCIE7          PCIE  pci_1  primary EMP
/SYS/MB/PCIE9          PCIE  pci_1  primary EMP
/SYS/MB/NET2           PCIE  pci_1  primary OCC
/SYS/MB/NET0/IOVNET.PF0 PF    pci_0  primary
/SYS/MB/NET0/IOVNET.PF1 PF    pci_0  primary
/SYS/MB/PCIE5/IOVNET.PF0 PF    pci_1  primary
/SYS/MB/PCIE5/IOVNET.PF1 PF    pci_1  primary
/SYS/MB/NET2/IOVNET.PF0 PF    pci_1  primary
/SYS/MB/NET2/IOVNET.PF1 PF    pci_1  primary

```

다음 명령은 지정된 물리적 기능에 대한 세부정보를 보여줍니다. `maxvfs` 값은 장치에서 지원하는 최대 가상 기능 수를 나타냅니다.

```

primary# ldm list-io -l /SYS/MB/NET0/IOVNET.PF0
NAME                                     TYPE  BUS    DOMAIN  STATUS
----                                     -
/SYS/MB/NET0/IOVNET.PF0                 PF    pci_0  primary
[pci@400/pci@1/pci@0/pci@4/network@0]
maxvfs = 7

```

예 6 선택적 등록 정보를 설정하지 않고 동적으로 이더넷 가상 기능 만들기

이 예에서는 선택적 등록 정보를 설정하지 않고 동적으로 가상 기능을 만듭니다. 이 경우 네트워크 클래스 가상 기능에 대한 MAC 주소가 자동으로 할당됩니다.

I/O 가상화가 `pci_0` PCIe 버스에서 사용으로 설정되어 있는지 확인합니다. [PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법 \[83\]](#)을 참조하십시오.

이제 `ldm create-vf` 명령을 사용하여 `/SYS/MB/NET0/IOVNET.PF0` 물리적 기능을 기반으로 가상 기능을 만들 수 있습니다.

```

primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0

```

예 7 동적으로 이더넷 가상 기능 만들기 및 등록 정보 설정

이 예에서는 `mac-addr` 등록 정보를 `00:14:2f:f9:14:c0`으로, `vid` 등록 정보를 VLAN ID 2 및 3으로 설정한 상태에서 동적으로 가상 기능을 만듭니다.

```

primary# ldm create-vf mac-addr=00:14:2f:f9:14:c0 vid=2,3 /SYS/MB/NET0/IOVNET.PF0

```

예 8 동적으로 2개의 대체 MAC 주소를 가진 이더넷 가상 기능 만들기

이 예에서는 동적으로 2개의 대체 MAC 주소를 가진 가상 기능을 만듭니다. 하나의 MAC 주소는 자동으로 할당되며 다른 MAC 주소는 명시적으로 `00:14:2f:f9:14:c2`로 지정됩니다.

```

primary# ldm create-vf alt-mac-addr=auto,00:14:2f:f9:14:c2 /SYS/MB/NET0/IOVNET.PF0

```

예 9 선택적 등록 정보를 설정하지 않고 정적으로 가상 기능 만들기

이 예에서는 선택적 등록 정보를 설정하지 않고 정적으로 가상 기능을 만듭니다. 이 경우 네트워크 클래스 가상 기능에 대한 MAC 주소가 자동으로 할당됩니다.

먼저 `primary` 도메인에서 지연된 재구성을 시작한 다음 `pci_0` PCIe 버스에서 I/O 가상화를 사용으로 설정합니다. `pci_0` 버스가 이미 `primary` 루트 도메인에 지정되었으므로 `ldm set-io` 명령을 사용하여 I/O 가상화를 사용으로 설정합니다.

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

```
primary# ldm set-io iov=on pci_0
```

이제 `ldm create-vf` 명령을 사용하여 `/SYS/MB/NET0/IOVNET.PF0` 물리적 기능을 기반으로 가상 기능을 만들 수 있습니다.

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0
```

마지막으로 변경사항이 적용되도록 `primary` 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

예 10 여러 SR-IOV 이더넷 가상 기능 만들기

다음 명령은 `/SYS/MB/NET2/IOVNET.PF1` 물리적 기능에서 4개의 가상 기능을 만들 수 있는 방법을 보여줍니다.

```
primary# ldm create-vf -n 31 /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF2
...
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF30
```

`ldm create-vf -n` 명령은 기본 등록 정보 값(해당하는 경우)으로 설정된 여러 가상 기능을 만듭니다. 나중에 `ldm set-io` 명령을 사용하여 기본값이 아닌 등록 정보 값을 지정할 수 있습니다.

이더넷 가상 기능 삭제

가상 기능은 현재 도메인에 지정되지 않은 경우 삭제할 수 있습니다. 가상 기능은 만든 순서와 역순으로만 삭제할 수 있으므로 마지막으로 만들어진 가상 기능만 삭제할 수 있습니다. 이로 인한 구성은 물리적 기능 드라이버를 통해 검증됩니다.

▼ 이더넷 SR-IOV 가상 기능을 삭제하는 방법

이 동적 방법을 사용할 수 없는 경우 정적 방법을 대신 사용하십시오. “정적 SR-IOV” [80]를 참조하십시오.

1. 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

2. 동적이나 정적으로 단일 가상 기능 또는 여러 가상 기능을 삭제합니다.

■ 동적 방법:

- 일부 또는 전체 가상 기능을 물리적 기능에서 한 번에 삭제하려면 다음 명령을 사용합니다.

```
primary# ldm destroy-vf -n number | max pf-name
```

`ldm destroy-vf -n max` 명령을 사용하여 한 번에 해당 물리적 기능에 대한 모든 가상 기능을 삭제합니다.

`number`를 `-n` 옵션에 대한 인수로 지정할 경우 가상 기능의 마지막 `number`가 삭제됩니다. 하나의 물리적 기능 장치 드라이버 상태 전환에서만 이 작업을 수행하므로 이 방법을 사용하십시오.

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

- 지정된 가상 기능을 삭제하려면 다음과 같이 하십시오.

```
primary# ldm destroy-vf vf-name
```

영향을 받는 하드웨어 장치와 OS의 지연 때문에, 영향을 받는 물리적 기능과 나머지 하위 가상 기능을 즉시 사용하지 못할 수도 있습니다. `ldm list-io` 명령을 사용하여 상위 물리적 기능과 그 하위 가상 기능에서 `Status` 열에 `INV` 값이 있는지 여부를 확인합니다. 이 값이 보이면 `ldm list-io` 출력에서 `Status` 열에 `INV` 값이 더 이상 표시되지 않을 때까지 기다리십시오(약 45초 정도). 그 후에 물리적 기능이나 하위 가상 기능을 안전하게 사용할 수 있습니다. 이 상태가 지속되면 장치에 문제가 있는 것입니다.

`primary`를 포함한 루트 도메인 재부트 직후나 `ldm create-vf` 또는 `ldm destroy-vf` 명령을 사용한 직후에 장치 상태가 `INV`로 표시될 수 있습니다.

■ 정적 방법:

a. 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

b. 단일 가상 기능 또는 여러 가상 기능을 삭제합니다.

- 지정된 물리적 기능에서 모든 가상 기능을 동시에 삭제하려면 다음 명령을 사용합니다.

```
primary# ldm destroy-vf -n number | max pf-name
```

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

- 지정된 가상 기능을 삭제하려면 다음과 같이 하십시오.

```
primary# ldm destroy-vf vf-name
```

c. 루트 도메인을 재부트합니다.

- 비primary 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# ldm stop-domain -r root-domain
```

- primary 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# shutdown -i6 -g0 -y
```

예 11 이더넷 가상 기능 삭제

이 예는 동적으로 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 삭제하는 방법을 보여줍니다.

```
primary# ldm destroy-vf /SYS/MB/NET0/IOVNET.PF0.VF0
```

다음 예에서는 정적으로 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 삭제하는 방법을 보여줍니다.

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

```
primary# ldm destroy-vf /SYS/MB/NET0/IOVNET.PF0.VF0
primary# shutdown -i6 -g0 -y
```

예 12 여러 이더넷 SR-IOV 가상 기능 삭제

이 예는 /SYS/MB/NET2/IOVNET.PF1 물리적 기능에서 모든 가상 기능을 삭제하는 결과를 보여줍니다. ldm list-io 출력은 물리적 기능에 7개의 가상 기능이 있음을 보여줍니다. ldm destroy-vf 명령은 모든 가상 기능을 삭제하고, 최종 ldm list-io 출력은 가상 기능이 하나도 남아 있지 않음을 보여줍니다.

```
primary# ldm list-io
...
/SYS/MB/NET2/IOVNET.PF1          PF      pci_1
/SYS/MB/NET2/IOVNET.PF1.VF0     VF      pci_1
/SYS/MB/NET2/IOVNET.PF1.VF1     VF      pci_1
```

```

/SYS/MB/NET2/IOVNET.PF1.VF2          VF      pci_1
/SYS/MB/NET2/IOVNET.PF1.VF3          VF      pci_1
/SYS/MB/NET2/IOVNET.PF1.VF4          VF      pci_1
/SYS/MB/NET2/IOVNET.PF1.VF5          VF      pci_1
/SYS/MB/NET2/IOVNET.PF1.VF6          VF      pci_1
primary# ldm destroy-vf -n max /SYS/MB/NET2/IOVNET.PF1
primary# ldm list-io
...
/SYS/MB/NET2/IOVNET.PF1              PF      pci_1    ldg1

```

이더넷 SR-IOV 가상 기능 수정

`ldm set-io vf-name` 명령은 등록 정보 값을 변경하거나 새 등록 정보를 설정하여 가상 기능의 현재 구성을 수정합니다. 이 명령은 네트워크 관련 등록 정보와 장치 관련 등록 정보를 모두 수정할 수 있습니다. 장치 관련 등록 정보에 대한 자세한 내용은 [“고급 SR-IOV 항목: 이더넷 SR-IOV” \[97\]](#)를 참조하십시오.

이 동적 방법을 사용할 수 없을 경우 정적 방법을 사용하십시오. [“정적 SR-IOV” \[80\]](#)를 참조하십시오.

`ldm set-io` 명령을 사용하여 다음 등록 정보를 수정할 수 있습니다.

- `mac-addr`, `alt-mac-addr`s 및 `mtu`

해당 가상 기능 등록 정보를 변경하려면 가상 기능을 소유한 도메인을 중지하고 `ldm set-io` 명령을 사용하여 등록 정보 값을 변경한 다음 도메인을 시작합니다.

- `pvid` 및 `vid`

도메인에 가상 기능이 지정되는 동안 이러한 등록 정보를 동적으로 변경할 수 있습니다. 이 작업을 수행하면 활성 가상 기능의 네트워크 트래픽이 변경될 수 있습니다. `pvid` 등록 정보를 설정하면 투명한 VLAN이 가능합니다. `vid` 등록 정보를 설정하여 VLAN ID를 지정하면 지정된 해당 VLAN에 대한 VLAN 트래픽이 허용됩니다.

- 장치 관련 등록 정보

`ldm list-io -d pf-name` 명령을 사용하여 유효한 장치 관련 등록 정보 목록을 확인합니다. 물리적 기능과 가상 기능 모두에 대해 이러한 등록 정보를 수정할 수 있습니다. 장치 관련 등록 정보를 수정하려면 정적 방법을 사용해야 합니다. [“정적 SR-IOV” \[80\]](#)를 참조하십시오. 장치 관련 등록 정보에 대한 자세한 내용은 [“고급 SR-IOV 항목: 이더넷 SR-IOV” \[97\]](#)를 참조하십시오.

▼ 이더넷 SR-IOV 가상 기능 등록 정보를 수정하는 방법

1. 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

물리적 기능의 이름에는 PCIe SR-IOV 카드 또는 내장 장치에 대한 위치 정보가 포함되어 있습니다.

2. 가상 기능 등록 정보를 수정합니다.

```
ldm set-io name=value [name=value...] vf-name
```

예 13 이더넷 가상 기능 등록 정보 수정

이러한 예에서는 `ldm set-io` 명령을 사용하여 이더넷 가상 기능의 등록 정보를 설정하는 방법에 대해 설명합니다.

- 다음 예에서는 VLAN ID 2, 3, 4에 속하도록 지정된 가상 기능 `/SYS/MB/NET0/IOVNET.PF0.VF0`의 등록 정보를 수정합니다.

```
primary# ldm set-io vid=2,3,4 /SYS/MB/NET0/IOVNET.PF0.VF0
```

이 명령은 가상 기능에 대한 VLAN 연관을 동적으로 변경합니다. 이러한 VLAN을 사용하려면 적절한 Oracle Solaris OS 네트워킹 명령을 사용하여 I/O 도메인의 VLAN 인터페이스를 구성해야 합니다.

- 다음 예에서는 가상 기능이 투명하게 VLAN 2에 속하도록 `/SYS/MB/NET0/IOVNET.PF0.VF0` 가상 기능에 대한 `pvid` 등록 정보 값을 2로 설정합니다. 즉, 가상 기능에서 태그가 지정된 VLAN 트래픽을 확인하지 않습니다.

```
primary# ldm set-io pvid=2 /SYS/MB/NET0/IOVNET.PF0.VF0
```

- 다음 예에서는 자동으로 할당된 3개의 대체 MAC 주소를 가상 기능에 지정합니다. 대체 주소를 통해 가상 기능에 Oracle Solaris 11 VNIC(가상 네트워크 인터페이스 카드)를 만들 수 있습니다. VNIC를 사용하려면 도메인에서 Oracle Solaris 11 OS를 실행해야 합니다.

주 - 이 명령을 실행하기 전에 가상 기능을 소유한 도메인을 중지하십시오.

```
primary# ldm set-io alt-mac-addr=auto,auto,auto /SYS/MB/NET0/IOVNET.PF0.VF0
```

- 다음 예에서는 지정된 가상 기능에 대한 장치 관련 `unicast-slots` 등록 정보를 12로 설정합니다. 물리적 기능에 유효한 장치 관련 등록 정보를 찾으려면 `ldm list-io -d pf-name` 명령을 사용합니다.

```
primary# ldm set-io unicast-slots=12 /SYS/MB/NET0/IOVNET.PF0.VF0
```

All configuration changes for other domains are disabled until the primary domain reboots, at which time the new configuration for the primary domain will also take effect.

I/O 도메인에서 이더넷 SR-IOV 가상 기능 추가 및 제거

▼ I/O 도메인에 이더넷 SR-IOV 가상 기능을 추가하는 방법

가상 기능을 동적으로 제거할 수 없는 경우 정적 방법을 사용하십시오. “[정적 SR-IOV](#)” [80]를 참조하십시오.

1. I/O 도메인에 추가할 가상 기능을 식별합니다.

```
primary# ldm list-io
```

2. 동적이나 정적으로 가상 기능을 추가합니다.

- 동적으로 가상 기능을 추가하려면 다음과 같이 하십시오.

```
primary# ldm add-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 추가할 도메인의 이름을 지정합니다.

도메인의 가상 기능에 대한 장치 경로 이름은 `list-io -l` 출력에 표시된 경로입니다.

- 정적으로 가상 기능을 추가하려면 다음과 같이 하십시오.

- a. 지연된 재구성을 시작한 다음 가상 기능을 추가합니다.

```
primary# ldm start-reconf root-domain-name
primary# ldm add-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 추가할 도메인의 이름을 지정합니다. 지정된 게스트는 비활성 또는 바인드 상태여야 합니다.

도메인의 가상 기능에 대한 장치 경로 이름은 `list-io -l` 출력에 표시된 경로입니다.

- b. 루트 도메인을 재부트합니다.

- `primary` 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# ldm stop-domain -r root-domain
```

- `primary` 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# shutdown -i6 -g0 -y
```

예 14 이더넷 가상 기능 추가

이 예는 동적으로 ldg1 도메인에 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 추가하는 방법을 보여줍니다.

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

동적으로 가상 기능을 추가할 수 없을 경우 정적 방법을 사용하십시오.

```
primary# ldm stop-domain ldg1
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
primary# ldm start-domain ldg1
```

▼ I/O 도메인에서 이더넷 가상 SR-IOV 기능을 제거하는 방법

가상 기능을 동적으로 제거할 수 없는 경우 정적 방법을 사용하십시오. “정적 SR-IOV” [80]를 참조하십시오.



주의 - 도메인에서 가상 기능을 제거하기 전에 해당 도메인 부트에 반드시 필요한 가상 기능이 아닌지 확인하십시오.

- I/O 도메인에서 제거할 가상 기능을 식별합니다.**

```
primary# ldm list-io
```

- 동적이나 정적으로 가상 기능을 제거합니다.**

- 동적으로 가상 기능을 제거하려면 다음과 같이 하십시오.

```
primary# ldm remove-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 장치 익명을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 제거할 도메인의 이름을 지정합니다.

- 정적으로 가상 기능을 제거하려면 다음과 같이 하십시오.

- I/O 도메인을 중지합니다.**

```
primary# ldm stop-domain domain-name
```

- 가상 기능을 제거합니다.**

```
primary# ldm remove-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 장치 익명을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 제거할 도메인의 이름을 지정합니다. 지정된 게스트는 비활성 또는 바인드 상태여야 합니다.

- I/O 도메인을 시작합니다.**

```
primary# ldm start-domain domain-name
```

예 15 동적으로 이더넷 가상 기능 제거

이 예는 동적으로 ldg1 도메인에서 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능을 제거하는 방법을 보여줍니다.

```
primary# ldm remove-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

명령이 성공하면 ldg1 도메인에서 가상 기능이 제거됩니다. ldg1이 다시 시작되면 지정된 가상 기능이 더 이상 해당 도메인에 나타나지 않습니다.

동적으로 가상 기능을 제거할 수 없을 경우 정적 방법을 사용하십시오.

```
primary# ldm stop-domain ldg1
primary# ldm remove-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
primary# ldm start-domain ldg1
```

고급 SR-IOV 항목: 이더넷 SR-IOV

이 절에서는 SR-IOV 가상 기능 사용과 관련된 몇 가지 고급 항목에 대해 설명합니다.

가상 기능에 대한 고급 네트워크 구성

SR-IOV 가상 기능을 사용하는 경우 다음 문제에 유의하십시오.

- SR-IOV 가상 기능은 Logical Domains Manager가 지정한 MAC 주소만 사용할 수 있습니다. 다른 Oracle Solaris OS 네트워킹 명령을 사용하여 I/O 도메인에서 MAC 주소를 변경할 경우 명령이 실패하거나 제대로 작동하지 않을 수 있습니다.
- 이때 I/O 도메인에 있는 SR-IOV 네트워크 가상 기능의 링크 통합은 지원되지 않습니다. 링크 통합을 만들려고 하면 예상대로 작동하지 않을 수 있습니다.
- 가상 I/O 서비스를 만들어 I/O 도메인에 지정할 수 있습니다. 이러한 가상 I/O 서비스는 가상 기능이 만들어진 것과 동일한 물리적 기능에 만들 수 있습니다. 예를 들어, 내장 1Gbps 네트워크 장치(`net0` 또는 `igb0`)를 가상 스위치용 네트워크 백엔드 장치로 사용할 수 있으며 동일한 물리적 기능 장치에서 정적으로 가상 기능을 만들 수도 있습니다.

SR-IOV 가상 기능을 사용하여 I/O 도메인 부트

SR-IOV 가상 기능은 다른 유형의 PCIe 장치와 유사한 기능(예: 가상 기능을 논리적 도메인 부트 장치로 사용할 수 있는 기능)을 제공합니다. 예를 들어, 네트워크 가상 기능을 사용하면 네트워크를 통해 부트하여 I/O 도메인에 Oracle Solaris OS를 설치할 수 있습니다.

주 - 가상 기능 장치에서 Oracle Solaris OS를 부트할 때 로드하려는 Oracle Solaris OS가 가상 기능 장치를 지원하는지 확인하십시오. 지원하는 경우 계획대로 나머지 설치를 계속할 수 있습니다.

SR-IOV 장치 관련 등록 정보

SR-IOV 물리적 기능 장치 드라이버는 장치 관련 등록 정보를 내보낼 수 있습니다. 이러한 등록 정보를 사용하여 물리적 기능과 해당 가상 기능의 리소스 할당을 조정할 수 있습니다. 등록 정보에 대한 자세한 내용은 [igb\(7D\)](#) 및 [ixgbe\(7D\)](#) 매뉴얼 페이지와 같은 물리적 기능 드라이버에 대한 매뉴얼 페이지를 참조하십시오.

`ldm list-io -d` 명령은 지정된 물리적 기능 장치 드라이버가 내보낸 장치 관련 등록 정보를 보여줍니다. 각 등록 정보에 대한 정보에는 이름, 간단한 설명, 기본값, 최대값 및 하나 이상의 다음 플래그가 포함되어 있습니다.

P	물리적 기능에 적용됩니다.
V	가상 기능에 적용됩니다.
R	읽기 전용 또는 단순한 정보 매개변수입니다.

```
primary# ldm list-io -d pf-name
```

물리적 기능 또는 가상 기능에 대해 읽기/쓰기 등록 정보를 설정하려면 `ldm create-vf` 또는 `ldm set-io` 명령을 사용하십시오. 장치 관련 등록 정보를 설정하려면 정적 방법을 사용해야 합니다. “정적 SR-IOV” [80]를 참조하십시오.

다음 예에서는 내장 Intel 1Gbps SR-IOV 장치가 내보낸 장치 관련 등록 정보를 보여줍니다.

```
primary# ldm list-io -d /SYS/MB/NET0/IOVNET.PF0
Device-specific Parameters
-----
max-config-vfs
  Flags = PR
  Default = 7
  Descr = Max number of configurable VFs
max-vf-mtu
  Flags = VR
  Default = 9216
  Descr = Max MTU supported for a VF
max-vlans
  Flags = VR
  Default = 32
  Descr = Max number of VLAN filters supported
pvid-exclusive
  Flags = VR
  Default = 1
  Descr = Exclusive configuration of pvid required
unicast-slots
  Flags = PV
  Default = 0 Min = 0 Max = 24
```

Descr = Number of unicast mac-address slots

다음 예에서는 unicast-slots 등록 정보를 8로 설정합니다.

```
primary# ldm create-vf unicast-slots=8 /SYS/MB/NET0/IOVNET.PF0
```

SR-IOV 가상 기능에 vNIC 만들기

SR-IOV 가상 기능에 Oracle Solaris 11 vNIC를 만들 수 있습니다. 하지만 지원되는 vNIC 수는 가상 기능에 지정된 대체 MAC 주소 수(alt-mac-addr 등록 정보)로 제한됩니다. 가상 기능에서 vNIC를 사용할 때는 대체 MAC 주소 수를 충분히 지정해야 합니다. 대체 MAC 주소와 함께 alt-mac-addr 등록 정보를 설정하려면 ldm create-vf 또는 ldm set-io 명령을 사용하십시오.

다음 예에서는 SR-IOV 가상 기능에 4개의 vNIC를 생성하는 과정을 보여줍니다. 첫번째 명령은 가상 기능 장치에 대체 MAC 주소를 지정합니다. 이 명령에서는 자동 할당 방법을 사용하여 /SYS/MB/NET0/IOVNET.PF0.VF0 가상 기능 장치에 4개의 대체 MAC 주소를 할당합니다.

```
primary# ldm set-io alt-mac-addr=auto,auto,auto,auto /SYS/MB/NET0/IOVNET.PF0.VF0
```

다음 명령은 ldg1 I/O 도메인을 시작합니다. 이 예에서는 auto-boot? 등록 정보가 true로 설정되어 있으므로 I/O 도메인에서 Oracle Solaris 11 OS도 부트됩니다.

```
primary# ldm start ldg1
```

다음 명령은 게스트 도메인에서 Oracle Solaris 11 dladm 명령을 사용하여 대체 MAC 주소가 있는 가상 기능을 표시합니다. 이 출력은 net30 가상 기능에 4개의 대체 MAC 주소가 있음을 보여줍니다.

```
guest# dladm show-phys -m
LINK          SLOT      ADDRESS          INUSE CLIENT
net0          primary  0:14:4f:fa:b4:d1 yes   net0
net25         primary  0:14:4f:fa:c9:eb no    --
net30         primary  0:14:4f:fb:de:4c no    --
              1        0:14:4f:f9:e8:73 no    --
              2        0:14:4f:f8:21:58 no    --
              3        0:14:4f:fa:9d:92 no    --
              4        0:14:4f:f9:8f:1d no    --
```

다음 명령은 4개의 vNIC를 만듭니다. 대체 MAC 주소를 사용하여 지정된 것보다 많은 수의 vNIC를 만들려고 시도하면 실패합니다.

```
guest# dladm create-vnic -l net30 vnic0
guest# dladm create-vnic -l net30 vnic1
guest# dladm create-vnic -l net30 vnic2
guest# dladm create-vnic -l net30 vnic3
guest# dladm show-link
LINK          CLASS    MTU    STATE    OVER
net0          phys    1500   up       --
net25         phys    1500   up       --
net30         phys    1500   up       --
vnic0         vnic    1500   up       net30
vnic1         vnic    1500   up       net30
vnic2         vnic    1500   up       net30
```

```
vnic3          vnic      1500  up      net30
```

SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기

다음 절차에서는 PCIe SR-IOV 가상 기능이 포함된 I/O 도메인을 만드는 방법에 대해 설명합니다.

▼ SR-IOV 가상 기능을 지정하여 I/O 도메인을 만드는 방법

작동 중지 시간이 최소화되도록 사전 계획을 통해 루트 도메인 재부트 횟수를 최소화하십시오.

시작하기 전에 시작하기 전에 가상 기능을 만들 기반이 되는 물리적 기능의 상위인 PCIe 버스에 대한 I/O 가상화를 사용으로 설정했는지 확인하십시오. [PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법 \[83\]](#)을 참조하십시오.

1. SR-IOV 기능을 사용하는 I/O 도메인과 공유할 SR-IOV 물리적 기능을 식별합니다.

```
primary# ldm list-io
```

2. 물리적 기능에 대한 가상 기능을 하나 이상 만듭니다.

```
primary# ldm create-vf pf-name
```

만들려는 각 가상 기능에 대해 이 명령을 실행할 수 있습니다. -n 옵션을 사용하면 단일 명령으로 동일한 물리적 기능에서 둘 이상의 가상 기능을 만들 수도 있습니다. 예 10. ["여러 SR-IOV 이더넷 가상 기능 만들기"](#) 및 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

주 - 연관된 물리적 기능을 기반으로 다른 가상 기능이 이미 만들어진 경우 및 해당 가상 기능이 다른 도메인에 바인드된 경우 이 명령이 실패합니다.

3. 루트 도메인에서 사용 가능한 가상 기능 목록을 확인합니다.

```
primary# ldm list-io
```

4. **2단계**에서 만든 가상 기능을 대상 I/O 도메인에 지정합니다.

```
primary# ldm add-io vf-name domain-name
```

주 - 대상 I/O 도메인의 OS가 동적 SR-IOV를 지원하지 않을 경우 정적 방법을 사용해야 합니다. ["정적 SR-IOV" \[80\]](#)를 참조하십시오.

5. I/O 도메인에서 가상 기능을 사용할 수 있는지 확인합니다.

다음 Oracle Solaris 11 명령은 가상 기능의 가용성을 보여줍니다.

```
guest# dladm show-phys
```

예 16 SR-IOV 가상 기능을 지정하여 동적으로 I/O 도메인 만들기

다음 동적 예에서는 물리적 기능 `/SYS/MB/NET0/IOVNET.PF0`에 대해 가상 기능 `/SYS/MB/NET0/IOVNET.PF0.VF0`을 만들고 `ldg1` I/O 도메인에 가상 기능을 지정하는 방법을 보여줍니다.

이 예에서는 다음과 같이 가정합니다.

- `primary` 도메인의 OS가 동적 SR-IOV 작업을 지원합니다.
- `pci_0` 버스가 `primary` 도메인에 지정되어 있으며 I/O 가상화 작업을 위해 초기화되었습니다.
- `/SYS/MB/NET0/IOVNET.PF0` 물리적 기능이 `pci_0` 버스에 속합니다.
- `/SYS/MB/NET0/IOVNET.PF0` 물리적 기능의 기존 가상 기능이 도메인에 지정되어 있지 않습니다.
- `ldg1` 도메인이 활성 상태이며 부트되었고 해당 OS가 동적 SR-IOV 작업을 지원합니다.

`/SYS/MB/NET0/IOVNET.PF0` 물리적 기능을 기반으로 가상 기능을 만듭니다.

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0
```

`/SYS/MB/NET0/IOVNET.PF0.VF0` 가상 기능을 `ldg1` 도메인에 추가합니다.

```
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
```

다음 명령은 가상 기능이 `ldg1` 도메인에 추가되었음을 보여줍니다.

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN  STATUS
----                                     -
niu_0                                   NIU   niu_0    primary
niu_1                                   NIU   niu_1    primary
pci_0                                   BUS   pci_0    primary  IOV
pci_1                                   BUS   pci_1    primary
/SYS/MB/PCIE0                           PCIE  pci_0    primary  OCC
/SYS/MB/PCIE2                           PCIE  pci_0    primary  OCC
/SYS/MB/PCIE4                           PCIE  pci_0    primary  OCC
/SYS/MB/PCIE6                           PCIE  pci_0    primary  EMP
/SYS/MB/PCIE8                           PCIE  pci_0    primary  EMP
/SYS/MB/SASHBA                          PCIE  pci_0    primary  OCC
/SYS/MB/NET0                             PCIE  pci_0    primary  OCC
/SYS/MB/PCIE1                           PCIE  pci_1    primary  OCC
/SYS/MB/PCIE3                           PCIE  pci_1    primary  OCC
/SYS/MB/PCIE5                           PCIE  pci_1    primary  OCC
/SYS/MB/PCIE7                           PCIE  pci_1    primary  EMP
/SYS/MB/PCIE9                           PCIE  pci_1    primary  EMP
/SYS/MB/NET2                             PCIE  pci_1    primary  OCC
/SYS/MB/NET0/IOVNET.PF0                 PF    pci_0    primary
/SYS/MB/NET0/IOVNET.PF1                 PF    pci_0    primary
/SYS/MB/PCIE5/IOVNET.PF0                PF    pci_1    primary
/SYS/MB/PCIE5/IOVNET.PF1                PF    pci_1    primary
/SYS/MB/NET2/IOVNET.PF0                 PF    pci_1    primary
/SYS/MB/NET2/IOVNET.PF1                 PF    pci_1    primary
/SYS/MB/NET0/IOVNET.PF0.VF0             VF    pci_0    ldg1
```

예 17 SR-IOV 가상 기능을 지정하여 정적으로 I/O 도메인 만들기

다음 정적 예에서는 물리적 기능 `/SYS/MB/NET0/IOVNET.PF0`에 대해 가상 기능 `/SYS/MB/NET0/IOVNET.PF0.VF0`을 만들고 `ldg1` I/O 도메인에 가상 기능을 지정하는 방법을 보여줍니다.

이 예에서는 다음과 같이 가정합니다.

- primary 도메인의 OS가 동적 SR-IOV 작업을 지원하지 않습니다.
- pci_0 버스가 primary 도메인에 지정되어 있으며 I/O 가상화 작업을 위해 초기화되지 않았습니다.
- /SYS/MB/NET0/IOVNET.PF0 물리적 기능이 pci_0 버스에 속합니다.
- /SYS/MB/NET0/IOVNET.PF0 물리적 기능의 기존 가상 기능이 도메인에 지정되어 있지 않습니다.
- ldg1 도메인이 활성 상태이며 부트되었고 해당 OS가 동적 SR-IOV 작업을 지원하지 않습니다.
- 도메인이 시작될 때 자동으로 도메인이 부트되도록 ldg1 도메인의 auto-boot? 등록 정보가 true로 설정되었습니다.

먼저 primary 도메인에서 지연된 재구성을 시작하고, I/O 가상화를 사용으로 설정하고, /SYS/MB/NET0/IOVNET.PF0 물리적 기능을 기반으로 가상 기능을 만듭니다.

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
```

```
primary# ldm set-io iov=on pci_0
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
```

```
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
Created new vf: /SYS/MB/NET0/IOVNET.PF0.VF0
```

다음으로 primary 도메인을 종료합니다.

```
primary# shutdown -i6 -g0 -y
```

ldg1 도메인을 중지하고, 가상 기능을 추가하고, 도메인을 시작합니다.

```
primary# ldm stop ldg1
primary# ldm add-io /SYS/MB/NET0/IOVNET.PF0.VF0 ldg1
primary# ldm start ldg1
```

다음 명령은 가상 기능이 ldg1 도메인에 추가되었음을 보여줍니다.

```
primary# ldm list-io
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
niu_0                               NIU   niu_0   primary
niu_1                               NIU   niu_1   primary
pci_0                               BUS   pci_0   primary  IOV
pci_1                               BUS   pci_1   primary
/SYS/MB/PCIE0                       PCIE  pci_0   primary  OCC
/SYS/MB/PCIE2                       PCIE  pci_0   primary  OCC
/SYS/MB/PCIE4                       PCIE  pci_0   primary  OCC
/SYS/MB/PCIE6                       PCIE  pci_0   primary  EMP
/SYS/MB/PCIE8                       PCIE  pci_0   primary  EMP
/SYS/MB/SASHBA                      PCIE  pci_0   primary  OCC
/SYS/MB/NET0                        PCIE  pci_0   primary  OCC
```

/SYS/MB/PCIE1	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE3	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE5	PCIE	pci_1	primary	OCC
/SYS/MB/PCIE7	PCIE	pci_1	primary	EMP
/SYS/MB/PCIE9	PCIE	pci_1	primary	EMP
/SYS/MB/NET2	PCIE	pci_1	primary	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	primary	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	primary	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	primary	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	primary	
/SYS/MB/NET0/IOVNET.PF0.VF0	VF	pci_0	ldg1	

InfiniBand SR-IOV 가상 기능 사용

주 - InfiniBand SR-IOV 장치에 대한 정적 SR-IOV 메소드만 사용할 수 있습니다.

작동 중지 시간을 최소화하려면 루트 도메인이 지연된 재구성 상태이거나 게스트 도메인이 중지된 상태에서 모든 SR-IOV 명령을 하나의 그룹으로 실행하십시오. 이 방식으로 제한되는 SR-IOV 명령은 `ldm create-vf`, `ldm destroy-vf`, `ldm add-io` 및 `ldm remove-io` 명령입니다.

일반적으로 가상 기능은 둘 이상의 게스트 도메인에 지정되어 있습니다. 루트 도메인을 재부트 하면 루트 도메인의 가상 기능에 지정된 모든 게스트 도메인이 영향을 받습니다.

사용되지 않는 InfiniBand 가상 기능으로는 오버헤드가 거의 발생하지 않으므로 즉시 사용되지 않더라도 필요한 가상 기능을 미리 만들어 작동 중지 시간이 발생하지 않도록 할 수 있습니다.

InfiniBand SR-IOV 하드웨어 요구사항

필요한 PCIe InfiniBand SR-IOV 하드웨어에 대한 자세한 내용은 [“SR-IOV 하드웨어 및 소프트웨어 요구사항” \[76\]](#)을 참조하십시오.

InfiniBand SR-IOV 지원의 경우 루트 도메인에서 최소 Oracle Solaris 11.1 SRU 10 OS를 실행해야 합니다. I/O 도메인은 최소 Oracle Solaris 11.1 SRU 10 OS를 실행할 수 있습니다.

InfiniBand 가상 기능 만들기 및 삭제

▼ InfiniBand 가상 기능을 만드는 방법

이 절차에서는 InfiniBand SR-IOV 가상 기능을 만드는 방법에 대해 설명합니다.

1. 루트 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

2. `iovs=on`을 설정하여 I/O 가상화를 사용으로 설정합니다.

물리적 기능을 가진 버스에 대한 I/O 가상화가 사용으로 설정되지 않은 경우에만 이 단계를 수행합니다.

```
primary# ldm set-io iovs=on bus
```

3. 해당 루트 도메인에서 물리적 기능과 연관된 가상 기능을 하나 이상 만듭니다.

```
primary# ldm create-vf pf-name
```

만들려는 각 가상 기능에 대해 이 명령을 실행할 수 있습니다. `-n` 옵션을 사용하면 단일 명령으로 동일한 물리적 기능에서 둘 이상의 가상 기능을 만들 수도 있습니다. 예 10. “여러 SR-IOV 이더넷 가상 기능 만들기” 및 `ldm(1M)` 매뉴얼 페이지를 참조하십시오.

4. 루트 도메인을 재부트합니다.

다음 명령 중 하나를 실행합니다.

- 비primary 루트 도메인을 재부트합니다.

```
primary# ldm stop-domain -r root-domain
```

- primary 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

예 18 InfiniBand 가상 기능 만들기

다음 예에서는 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0` 물리적 기능에 대한 정보를 보여줍니다.

- 이 물리적 기능은 PCIE 슬롯 4에 있습니다.
- IOVIB 문자열은 물리적 기능이 InfiniBand SR-IOV 장치임을 나타냅니다.

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN  STATUS
----                                     -
pci_0                                   BUS   pci_0    primary
niu_0                                   NIU   niu_0    primary
/SYS/MB/RISER0/PCIE0                   PCIE  pci_0    primary  EMP
/SYS/MB/RISER1/PCIE1                   PCIE  pci_0    primary  EMP
/SYS/MB/RISER2/PCIE2                   PCIE  pci_0    primary  EMP
/SYS/MB/RISER0/PCIE3                   PCIE  pci_0    primary  OCC
/SYS/MB/RISER1/PCIE4                   PCIE  pci_0    primary  OCC
/SYS/MB/RISER2/PCIE5                   PCIE  pci_0    primary  EMP
/SYS/MB/SASHBA0                        PCIE  pci_0    primary  OCC
/SYS/MB/SASHBA1                        PCIE  pci_0    primary  OCC
/SYS/MB/NET0                           PCIE  pci_0    primary  OCC
/SYS/MB/NET2                           PCIE  pci_0    primary  OCC
/SYS/MB/RISER0/PCIE3/IOVIB.PF0        PF    pci_0    primary
/SYS/MB/RISER1/PCIE4/IOVIB.PF0        PF    pci_0    primary
```

```

/SYS/MB/NET0/IOVNET.PF0      PF      pci_0    primary
/SYS/MB/NET0/IOVNET.PF1      PF      pci_0    primary
/SYS/MB/NET2/IOVNET.PF0      PF      pci_0    primary
/SYS/MB/NET2/IOVNET.PF1      PF      pci_0    primary

```

다음 명령은 지정된 물리적 기능에 대한 세부정보를 보여줍니다. `maxvfs` 값은 장치에서 지원하는 최대 가상 기능 수를 나타냅니다.

```

primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                TYPE  BUS      DOMAIN  STATUS
-----
/SYS/MB/RISER1/PCIE4/IOVIB.PF0     PF    pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64

```

다음 예에서는 정적 가상 기능을 만드는 방법을 보여줍니다. 먼저 `primary` 도메인에서 지연된 재구성을 시작하고 `pci_0` PCIe 버스에서 I/O 가상화를 사용으로 설정합니다. `pci_0` 버스가 이미 `primary` 루트 도메인에 지정되었으므로 `ldm set-io` 명령을 사용하여 I/O 가상화를 사용으로 설정합니다.

```

primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.

```

```

primary# ldm set-io iov=on pci_0
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----

```

이제 `ldm create-vf` 명령을 사용하여 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0` 물리적 기능을 기반으로 가상 기능을 만듭니다.

```

primary# ldm create-vf /SYS/MB/RISER1/PCIE4/IOVIB.PF0
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
Created new vf: /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0

```

동일한 지연된 재구성 중 둘 이상의 가상 기능을 만들 수 있습니다. 다음 명령은 두번째 가상 기능을 만듭니다.

```

primary# ldm create-vf /SYS/MB/RISER1/PCIE4/IOVIB.PF0
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
Created new vf: /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1

```

마지막으로 변경사항이 적용되도록 `primary` 루트 도메인을 재부트합니다.

```

primary# shutdown -i6 -g0 -y
Shutdown started.

Changing to init state 6 - please wait
...

```

▼ InfiniBand 가상 기능을 삭제하는 방법

이 절차에서는 InfiniBand SR-IOV 가상 기능을 삭제하는 방법에 대해 설명합니다.

가상 기능은 현재 도메인에 지정되지 않은 경우 삭제할 수 있습니다. 가상 기능은 만든 순서와 역순으로만 삭제할 수 있으므로 마지막으로 만들어진 가상 기능만 삭제할 수 있습니다. 이로 인한 구성은 물리적 기능 드라이버를 통해 검증됩니다.

1. 루트 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

2. 해당 루트 도메인에서 물리적 기능과 연관된 가상 기능을 하나 이상 삭제합니다.

```
primary# ldm destroy-vf vf-name
```

삭제하려는 각 가상 기능에 대해 이 명령을 실행할 수 있습니다. `-n` 옵션을 사용하면 단일 명령으로 동일한 물리적 기능에서 둘 이상의 가상 기능을 삭제할 수도 있습니다. [예 12. “여러 이더넷 SR-IOV 가상 기능 삭제” 및 ldm\(1M\) 매뉴얼 페이지를 참조하십시오.](#)

3. 루트 도메인을 재부트합니다.

다음 명령 중 하나를 실행합니다.

- 비primary 루트 도메인을 재부트합니다.

```
primary# ldm stop-domain -r root-domain
```

- primary 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

예 19 InfiniBand 가상 기능 삭제

다음 예에서는 정적 InfiniBand 가상 기능 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1`을 삭제하는 방법을 보여줍니다.

`ldm list-io` 명령은 버스, 물리적 기능 및 가상 기능에 대한 정보를 보여줍니다.

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN STATUS
----
pci_0                                    BUS   pci_0    primary IOV
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0          PF    pci_0    primary
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0      VF    pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1      VF    pci_0
```

`ldm list-io -l` 명령을 사용하여 물리적 기능 및 관련 가상 기능에 대한 자세한 정보를 얻을 수 있습니다.

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                TYPE  BUS      DOMAIN STATUS
----                                -
/SYS/MB/RISER1/PCIE4/IOVIB.PF0      PF    pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0  VF    pci_0
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,1]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1  VF    pci_0
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,2]
```

가상 기능은 도메인에 지정되지 않은 경우에만 삭제할 수 있습니다. `ldm list-io -l` 출력의 DOMAIN 열은 가상 기능이 지정된 도메인의 이름을 보여줍니다. 또한 가상 기능은 만든 순서와 역순으로 삭제해야 합니다. 따라서 이 예에서는 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1` 가상 기능을 삭제해야만 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0` 가상 기능을 삭제할 수 있습니다.

적절한 가상 기능을 식별한 후 삭제할 수 있습니다. 먼저 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.

primary# ldm destroy-vf /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

지연된 재구성 상태에서 둘 이상의 `ldm destroy-vf` 명령을 실행할 수 있습니다. 따라서 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0`도 삭제할 수 있습니다.

마지막으로 변경사항이 적용되도록 `primary` 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
Shutdown started.

Changing to init state 6 - please wait
...
```

I/O 도메인에서 InfiniBand 가상 기능 추가 및 제거

▼ I/O 도메인에 InfiniBand 가상 기능을 추가하는 방법

이 절차에서는 InfiniBand SR-IOV 가상 기능을 I/O 도메인에 추가하는 방법을 설명합니다.

1. I/O 도메인을 중지합니다.

```
primary# ldm stop-domain domain-name
```

2. I/O 도메인에 하나 이상의 가상 기능을 추가합니다.

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 추가할 도메인의 이름을 지정합니다. 지정된 I/O 도메인은 비활성 또는 바인드 상태여야 합니다.

```
primary# ldm add-io vf-name domain-name
```

3. I/O 도메인을 시작합니다.

```
primary# ldm start-domain domain-name
```

예 20 InfiniBand 가상 기능 추가

다음 예에서는 *iodom1* I/O 도메인에 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2` 가상 기능을 추가하는 방법을 보여줍니다.

먼저 지정할 가상 기능을 식별합니다.

```
primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN STATUS
----                                     -
pci_0                                   BUS   pci_0    primary IOV
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0         PF    pci_0    primary
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0     VF    pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1     VF    pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2     VF    pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3     VF    pci_0
```

I/O 도메인에 가상 기능을 추가하려면 가상 기능이 지정되어 있지 않아야 합니다. DOMAIN 열은 가상 기능이 지정된 도메인의 이름을 나타냅니다. 이 예에서 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2`는 도메인에 지정되어 있지 않습니다.

도메인에 가상 기능을 추가하려면 도메인이 비활성 또는 바인드 상태여야 합니다.

```
primary# ldm list-domain
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary   active    -n-cv-  UART  32    64G     0.2%  0.2%  56m
iodom1    active    -n-    5000  8     8G      33%   33%   25m
```

`ldm list-domain` 출력은 *iodom1* I/O 도메인이 활성 상태여서 중지되어야 함을 보여줍니다.

```
primary# ldm stop iodom1
LDom iodom1 stopped
primary# ldm list-domain
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary   active    -n-cv-  UART  32    64G     0.0%  0.0%  57m
iodom1    bound     -      5000  8     8G
```

이제 I/O 도메인에 가상 기능을 추가할 수 있습니다.

```
primary# ldm add-io /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 iodom1
primary# ldm list-io
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2     VF    pci_0    iodom1
```

I/O 도메인이 중지된 상태에서 둘 이상의 가상 기능을 추가할 수 있습니다. 예를 들어, `iodom1`에 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3`과 같은 지정되지 않은 다른 가상 기능을 추가할 수 있습니다. 가상 기능을 추가한 후 I/O 도메인을 다시 시작할 수 있습니다.

```
primary# ldm start iodom1
LDom iodom1 started
primary# ldm list-domain
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary      active  -n-cv-  UART   32    64G     1.0%  1.0%  1h 18m
iodom1       active  -n----  5000   8     8G     36%   36%   1m
```

▼ I/O 도메인에서 InfiniBand 가상 기능을 제거하는 방법

이 절차에서는 InfiniBand SR-IOV 가상 기능을 I/O 도메인에서 제거하는 방법을 설명합니다.

1. I/O 도메인을 중지합니다.

```
primary# ldm stop-domain domain-name
```

2. I/O 도메인에서 하나 이상의 가상 기능을 제거합니다.

`vf-name`은 가상 기능의 익명 이름 또는 경로 이름입니다. 장치 이름을 사용하는 것이 좋습니다. `domain-name`은 가상 기능을 제거할 도메인의 이름을 지정합니다. 지정된 I/O 도메인은 비활성 또는 바인드 상태여야 합니다.

주 - I/O 도메인에서 가상 기능을 제거하기 전에 해당 도메인 부트에 반드시 필요한 가상 기능이 아닌지 확인하십시오.

```
primary# ldm remove-io vf-name domain-name
```

3. I/O 도메인을 시작합니다.

```
primary# ldm start-domain domain-name
```

예 21 InfiniBand 가상 기능 제거

다음 예에서는 `iodom1` I/O 도메인에서 `/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2` 가상 기능을 제거하는 방법을 보여줍니다.

먼저 제거할 가상 기능을 식별합니다.

```
primary# ldm list-io
NAME          TYPE  BUS      DOMAIN STATUS
-----
pci_0         BUS  pci_0    primary IOV
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0      PF  pci_0    primary
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0  VF  pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1  VF  pci_0
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2  VF  pci_0    iodom1
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3  VF  pci_0    iodom1
```

DOMAIN 열린 가상 기능이 지정된 도메인의 이름을 보여줍니다. /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 가상 기능이 iodom1에 지정되었습니다.

I/O 도메인에서 가상 기능을 제거하려면 도메인이 비활성 또는 바인드 상태여야 합니다. ldm list-domain 명령을 사용하여 도메인 상태를 확인합니다.

```
primary# ldm list-domain
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary       active    -n-cv-  UART   32    64G     0.3%  0.3%  29m
iodom1        active    -n----  5000   8     8G      17%   17%  11m
```

이 예에서 iodom1 도메인은 활성 상태이므로 중지되어야 합니다.

```
primary# ldm stop iodom1
LDM iodom1 stopped
primary# ldm list-domain
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary       active    -n-cv-  UART   32    64G     0.0%  0.0%  31m
iodom1        bound     -----  5000   8     8G
```

이제 iodom1에서 /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 가상 기능을 제거할 수 있습니다.

```
primary# ldm remove-io /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2 iodom1
primary# ldm list-io
NAME          TYPE  BUS  DOMAIN STATUS
-----
...
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2  VF   pci_0
...
```

가상 기능에 대한 DOMAIN 열이 비어 있습니다.

I/O 도메인이 중지된 상태에서 둘 이상의 가상 기능을 제거할 수 있습니다. 이 예에서는 /SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3 가상 기능도 제거할 수 있습니다. 가상 기능을 제거한 후 I/O 도메인을 다시 시작할 수 있습니다.

```
primary# ldm start iodom1
LDM iodom1 started
primary# ldm list-domain
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary       active    -n-cv-  UART   32    64G     0.3%  0.3%  39m
iodom1        active    -n----  5000   8     8G     9.4%  9.4%  5s
```

InfiniBand 가상 기능을 루트 도메인에 추가 및 루트 도메인에서 제거

▼ 루트 도메인에 InfiniBand 가상 기능을 추가하는 방법

이 절차에서는 InfiniBand SR-IOV 가상 기능을 루트 도메인에 추가하는 방법을 설명합니다.

1. 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain
```

2. 루트 도메인에 하나 이상의 가상 기능을 추가합니다.
*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *root-domain-name*은 가상 기능을 추가할 루트 도메인의 이름을 지정합니다.

```
primary# ldm add-io vf-name root-domain-name
```

3. 루트 도메인을 재부트합니다.
 다음 명령 중 하나를 실행합니다.

- 비primary 루트 도메인을 재부트합니다.

```
primary# ldm stop-domain -r root-domain-name
```

- primary 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

▼ 루트 도메인에서 InfiniBand 가상 기능을 제거하는 방법

이 절차에서는 InfiniBand SR-IOV 가상 기능을 루트 도메인에서 제거하는 방법을 설명합니다.

1. 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain
```

2. 루트 도메인에서 하나 이상의 가상 기능을 제거합니다.
*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *root-domain-name*은 가상 기능을 추가할 루트 도메인의 이름을 지정합니다.

```
primary# ldm remove-io vf-name root-domain-name
```

3. 루트 도메인을 재부트합니다.
 다음 명령 중 하나를 실행합니다.

- 비primary 루트 도메인을 재부트합니다.

```
primary# ldm stop-domain -r root-domain-name
```

- primary 루트 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

고급 SR-IOV 항목: InfiniBand SR-IOV

이 절에서는 InfiniBand 물리적 및 가상 기능을 식별하고 InfiniBand 물리적 기능과 가상 기능의 Logical Domains Manager 및 Oracle Solaris 뷰를 상관시키는 방법에 대해 설명합니다.

InfiniBand SR-IOV 가상 기능 나열

다음 예에서는 /SYS/MB/RISER1/PCIE4/IOVIB.PF0 물리적 기능에 대한 정보를 표시할 수 있는 여러 가지 방법을 보여줍니다. IOVIB 문자열을 포함하는 물리적 기능 이름은 InfiniBand SR-IOV 장치임을 나타냅니다.

```
primary# ldm list-io
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
pci_0                               BUS   pci_0    primary IOV
niu_0                                NIU   niu_0    primary
/SYS/MB/RISER0/PCIE0                PCIE  pci_0    primary EMP
/SYS/MB/RISER1/PCIE1                PCIE  pci_0    primary EMP
/SYS/MB/RISER2/PCIE2                PCIE  pci_0    primary EMP
/SYS/MB/RISER0/PCIE3                PCIE  pci_0    primary OCC
/SYS/MB/RISER1/PCIE4                PCIE  pci_0    primary OCC
/SYS/MB/RISER2/PCIE5                PCIE  pci_0    primary EMP
/SYS/MB/SASHBA0                     PCIE  pci_0    primary OCC
/SYS/MB/SASHBA1                     PCIE  pci_0    primary OCC
/SYS/MB/NET0                         PCIE  pci_0    primary OCC
/SYS/MB/NET2                         PCIE  pci_0    primary OCC
/SYS/MB/RISER0/PCIE3/IOVIB.PF0      PF    pci_0    primary
/SYS/MB/RISER1/PCIE4/IOVIB.PF0      PF    pci_0    primary
/SYS/MB/NET0/IOVNET.PF0              PF    pci_0    primary
/SYS/MB/NET0/IOVNET.PF1              PF    pci_0    primary
/SYS/MB/NET2/IOVNET.PF0              PF    pci_0    primary
/SYS/MB/NET2/IOVNET.PF1              PF    pci_0    primary
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF0   VF    pci_0    primary
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF1   VF    pci_0    primary
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF2   VF    pci_0    iodom1
/SYS/MB/RISER0/PCIE3/IOVIB.PF0.VF3   VF    pci_0    iodom1
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0   VF    pci_0    primary
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1   VF    pci_0    primary
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2   VF    pci_0    iodom1
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3   VF    pci_0    iodom1
```

ldm list-io -l 명령은 지정된 물리적 기능 장치 /SYS/MB/RISER1/PCIE4/IOVIB.PF0에 대한 보다 자세한 정보를 제공합니다. maxvfs 값은 물리적 장치가 지원하는 최대 가상 기능 수가 64개임을 보여줍니다. 물리적 기능과 연관된 각 가상 기능에 대해 출력은 다음을 보여줍니다.

- 기능 이름
- 기능 유형
- 버스 이름
- 도메인 이름
- 선택적 기능 상태
- 장치 경로

이 ldm list-io -l 출력은 VF0 및 VF1이 primary 도메인에 지정되었으며 VF2 및 VF3이 iodom1 I/O 도메인에 지정되었음을 보여줍니다.

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
/SYS/MB/RISER1/PCIE4/IOVIB.PF0      PF    pci_0    primary
```

```
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0      VF      pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,1]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1      VF      pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,2]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2      VF      pci_0    iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,3]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3      VF      pci_0    iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,4]
```

InfiniBand SR-IOV 기능 식별

이 절에서는 InfiniBand SR-IOV 장치를 식별하는 방법을 설명합니다.

ldm list-io -l 명령을 사용하여 각 물리적 기능 및 가상 기능과 연관된 Oracle Solaris 장치 경로 이름을 표시할 수 있습니다.

```
primary# ldm list-io -l /SYS/MB/RISER1/PCIE4/IOVIB.PF0
NAME                                TYPE    BUS      DOMAIN  STATUS
----                                -
/SYS/MB/RISER1/PCIE4/IOVIB.PF0      PF      pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0]
maxvfs = 64
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF0    VF      pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,1]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF1    VF      pci_0    primary
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,2]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF2    VF      pci_0    iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,3]
/SYS/MB/RISER1/PCIE4/IOVIB.PF0.VF3    VF      pci_0    iodom1
[pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0,4]
```

dladm show-phys -L 명령을 사용하여 각 IPoIB(IP over InfiniBand) 인스턴스를 물리적 카드에 일치시킬 수 있습니다. 예를 들어, 다음 명령은 슬롯 PCIE4에 이전 ldm list-io -l 예에서와 동일한 카드를 사용하는 IPoIB 인스턴스를 보여줍니다.

```
primary# dladm show-phys -L | grep PCIE4
net5          ibp0          PCIE4/PORT1
net6          ibp1          PCIE4/PORT2
net19         ibp8          PCIE4/PORT1
net9          ibp9          PCIE4/PORT2
net18         ibp4          PCIE4/PORT1
net11         ibp5          PCIE4/PORT2
```

각 InfiniBand HCA(호스트 채널 어댑터) 장치에는 GUID(전역적으로 고유한 ID)가 있습니다. 각 포트에 대해서도 GUID가 있습니다. 일반적으로 HCA에 대해 2개의 포트가 있습니다. InfiniBand HCA GUID는 어댑터를 고유하게 식별합니다. 포트 GUID는 각 HCA 포트를 고유하게 식별하고 네트워크 장치의 MAC 주소와 유사한 역할을 수행합니다. 이러한 16진수 GUID는 InfiniBand 관리 도구 및 진단 도구에서 사용됩니다.

dladm show-ib 명령을 사용하여 InfiniBand SR-IOV 장치에 대한 GUID 정보를 얻을 수 있습니다. 동일한 장치에 대한 물리적 기능 및 가상 기능에는 관련된 HCA GUID 값이 있습니다.

니다. HCA GUID의 11번째 16진수는 물리적 기능과 가상 기능 간의 관계를 보여줍니다. HCAGUID 및 PORTGUID 열에서는 선행 0이 숨겨집니다.

예를 들어, 물리적 기능 PF0에는 primary 도메인에 지정된 2개의 가상 기능인 VF0과 VF1이 있습니다. 각 가상 기능의 11번째 16진수는 관련된 물리적 기능에서 1씩 증분됩니다. 따라서 PF0에 대한 GUID가 8이면 VF0 및 VF1에 대한 GUID는 각각 9와 A가 됩니다.

다음 `dladm show-ib` 명령 출력은 net5 및 net6 링크가 물리적 기능 PF0에 속해 있음을 보여줍니다. net19 및 net9 링크는 동일한 장치의 VF0에 속하며, net18 및 net11 링크는 VF1에 속합니다.

```
primary# dladm show-ib
LINK          HCAGUID          PORTGUID          PORT STATE PKEYS
net6          21280001A17F56  21280001A17F58   2  up  FFFF
net5          21280001A17F56  21280001A17F57   1  up  FFFF
net19         21290001A17F56  140500000000001  1  up  FFFF
net9          21290001A17F56  140500000000008  2  up  FFFF
net18         212A0001A17F56  140500000000002  1  up  FFFF
net11         212A0001A17F56  140500000000009  2  up  FFFF
```

다음 `dladm show-phys` 출력의 장치는 링크 간의 관계 및 원래 InfiniBand 포트 장치(ibpX)를 보여줍니다.

```
primary# dladm show-phys
LINK          MEDIA          STATE          SPEED DUPLEX DEVICE
...
net6          Infiniband    up             32000 unknown ibp1
net5          Infiniband    up             32000 unknown ibp0
net19         Infiniband    up             32000 unknown ibp8
net9          Infiniband    up             32000 unknown ibp9
net18         Infiniband    up             32000 unknown ibp4
net11         Infiniband    up             32000 unknown ibp5
```

`ls -l` 명령을 사용하여 실제 IB 포트(InfiniBand 포트) 장치 경로를 표시할 수 있습니다. IB 포트 장치는 `ldm list-io -l` 출력에 표시된 장치 경로의 하위입니다. 물리적 기능의 단위 주소는 한 부분(예: `pciex15b3,673c@0`)이며 가상 기능의 단위 주소는 두 부분(예: `pciex15b3,1002@0,2`)입니다. 단위 주소의 두번째 부분은 가상 기능 번호보다 1이 큼니다. 이 예에서는 두번째 구성 요소가 2이므로 이 장치는 가상 기능 1입니다. 다음 출력은 `/dev/ibp0`이 물리적 기능이며 `/dev/ibp5`가 가상 기능임을 보여줍니다.

```
primary# ls -l /dev/ibp0
lrwxrwxrwx  1 root  root           83 Apr 18 12:02 /dev/ibp0 ->
../devices/pci@400/pci@1/pci@0/pci@0/pciex15b3,673c@0/hermon@0/ibport@1,0,ipib:ibp0
primary# ls -l /dev/ibp5
lrwxrwxrwx  1 root  root           85 Apr 22 23:29 /dev/ibp5 ->
../devices/pci@400/pci@1/pci@0/pci@0/pciex15b3,1002@0,2/hermon@3/ibport@2,0,ipib:ibp5
```

OpenFabrics `ibv_devices` 명령을 사용하여 OpenFabrics 장치 이름 및 노드 (HCA) GUID를 확인할 수 있습니다. 가상 기능이 있을 경우 Type 열은 기능이 물리적인지 아니면 가상인지 나타냅니다.

```
primary# ibv_devices
device          node GUID          type
-----
m1x4_4          0002c90300a38910  PF
m1x4_5          0021280001a17f56  PF
m1x4_0          0002cb0300a38910  VF
m1x4_1          0002ca0300a38910  VF
```

m1x4_2	00212a0001a17f56	VF
m1x4_3	0021290001a17f56	VF

광 섬유 채널 SR-IOV 가상 기능 사용

SR-IOV 광 섬유 채널 호스트 버스 어댑터(HBA)에는 각각 SR-IOV 물리적 기능으로 나타나는 하나 이상의 포트가 있을 수 있습니다. 장치 이름에서 `iovfc` 문자열로 광 섬유 채널 물리적 기능을 식별할 수 있습니다.

각 광 섬유 채널 물리적 기능에는 카드 제조업체가 제공하는 고유한 포트 및 노드 WWN (world-wide name) 값이 있습니다. 광 섬유 채널 물리적 기능에서 가상 기능을 만들 경우 가상 기능은 광 섬유 채널 HBA 장치와 같이 동작합니다. 각 가상 기능에는 SAN 패브릭의 포트 WWN 및 노드 WWN으로 지정된 고유한 ID가 있어야 합니다. Logical Domains Manager를 사용하여 포트 및 노드 WWN을 자동이나 수동으로 지정할 수 있습니다. 사용자 고유의 값을 지정하면 모든 가상 기능의 ID를 완전히 제어할 수 있습니다.

광 섬유 채널 HBA 가상 기능은 NPIV(N_Port ID Virtualization) 방법을 사용하여 SAN 패브릭에 로그인합니다. 이 NPIV 요구사항으로 인해 광 섬유 채널 HBA 포트를 NPIV 지원 광 섬유 채널 스위치에 연결해야 합니다. 가상 기능은 전적으로 SR-IOV 카드의 하드웨어나 펌웨어로 관리됩니다. 이러한 예외사항 이외에는 광 섬유 채널 가상 기능은 비SR-IOV 광 섬유 채널 HBA 장치와 동일하게 작동 및 동작합니다. SR-IOV 가상 기능은 비SR-IOV 장치와 동일한 기능을 가지므로 모든 유형의 SAN 스토리지 장치가 두 구성에서 지원됩니다.

가상 기능의 고유한 포트 및 노드 WWN 값을 통해 SAN 관리자는 비SR-IOV 광 섬유 채널 HBA 포트와 동일한 방법으로 가상 기능에 스토리지를 지정할 수 있습니다. 이 관리에는 영역 지정, LUN 마스킹 및 서비스 품질(QoS)이 포함됩니다. 루트 도메인의 물리적 기능에 표시되지 않고 특정 논리적 도메인에 배타적으로 액세스할 수 있도록 스토리지를 구성할 수 있습니다.

정적 및 동적 SR-IOV 방법을 모두 사용하여 광 섬유 채널 SR-IOV 장치를 관리할 수 있습니다.

광 섬유 채널 SR-IOV 하드웨어 요구사항

필요한 PCIe 광 섬유 채널 SR-IOV 하드웨어에 대한 자세한 내용은 [“SR-IOV 하드웨어 및 소프트웨어 요구사항” \[76\]](#)을 참조하십시오.

- 컨트롤 도메인.
 - QLogic 카드. 최소 Oracle Solaris 11.2 OS
 - Emulex 카드. 최소 Oracle Solaris 11.1 SRU 17 OS

- I/O 도메인.
 - QLogic 카드. 최소 Oracle Solaris 11.2 OS
 - Emulex 카드. 최소 Oracle Solaris 11.1 SRU 17 OS

광 섬유 채널 SR-IOV 요구사항 및 제한 사항

광 섬유 채널 SR-IOV 기능에는 다음 권장 사항 및 제한 사항이 있습니다.

- SR-IOV 카드는 SR-IOV 기능을 지원하는 최신 버전의 펌웨어를 실행해야 합니다.
- 광 섬유 채널 PCIe 카드는 NPIV를 지원하고 PCIe 카드와 호환되는 광 섬유 채널 스위치에 연결해야 합니다.
- 모든 시스템의 컨트롤 도메인을 동일한 SAN 패브릭에 연결하고 동일한 멀티캐스트 도메인의 일부가 되면 Logical Domains Manager는 고유한 `port-wwn` 및 `node-wwn` 등록 정보 값을 제대로 자동 생성합니다.

환경을 구성할 수 없는 경우 가상 기능을 만들 때 `node-wwn` 및 `port-wwn` 값을 수동으로 제공해야 합니다. 이 동작은 이름 지정 충돌이 발생하지 않도록 합니다. “[광 섬유 채널 가상 기능에 대한 WWN\(world-wide name\) 할당](#)” [117]을 참조하십시오.

광 섬유 채널 장치 클래스 관련 등록 정보

`ldm create-vf` 또는 `ldm set-io` 명령을 사용하여 다음 광 섬유 채널 가상 기능 등록 정보를 설정할 수 있습니다.

<code>bw-percent</code>	광 섬유 채널 가상 기능에 할당할 대역폭의 백분율을 지정합니다. 유효한 값은 0 - 100입니다. 광 섬유 채널 물리적 기능의 가상 기능에 지정한 총 대역폭 값은 100을 초과할 수 없습니다. 기본값은 0이므로 가상 기능은 동일한 물리적 기능을 공유하는 다른 가상 기능에 의해 이미 예약되지 않은 충분한 대역폭을 얻게 됩니다.
<code>node-wwn</code>	광 섬유 채널 가상 기능에 대한 노드 WWN(world-wide name)을 지정합니다. 유효한 값은 0이 아닌 값입니다. 기본적으로 이 값은 자동으로 할당됩니다. 이 값을 수동으로 지정할 경우 <code>port-wwn</code> 등록 정보에 대한 값도 지정해야 합니다. 자세한 내용은 “ 광 섬유 채널 가상 기능에 대한 WWN(world-wide name) 할당 ” [117]을 참조하십시오.
<code>port-wwn</code>	광 섬유 채널 가상 기능에 대한 포트 WWN을 지정합니다. 유효한 값은 0이 아닌 값입니다. 기본적으로 이 값은 자동으로 할당됩니다. 이 값을 수동으로 지정할 경우 <code>node-wwn</code> 등록 정보에 대한 값도 지정해야 합니다. 자세한 내용은 “ 광 섬유 채널 가상 기능에 대한 WWN(world-wide name) 할당 ” [117]을 참조하십시오.

광 섬유 채널 가상 기능이 사용 중일 때는 `node-wwn` 또는 `port-wwn` 등록 정보 값을 수정할 수 없습니다. 하지만 광 섬유 채널 가상 기능이 사용 중일 때도 `bw-percent` 등록 정보 값은 동적으로 수정할 수 있습니다.

광 섬유 채널 가상 기능에 대한 WWN(world-wide name) 할당

Logical Domains Manager는 광 섬유 채널 가상 기능에 대한 WWN(world-wide name)의 자동 할당과 수동 지정을 모두 지원합니다.

자동 WWN(world-wide name) 할당

Logical Domains Manager는 자동 MAC 주소 할당 풀에서 고유한 MAC 주소를 할당하고 IEEE 형식 `node-wwn` 및 `port-wwn` 등록 정보 값을 만듭니다.

```
port-wwn = 10:00:XX:XX:XX:XX:XX:XX
node-wwn = 20:00:XX:XX:XX:XX:XX:XX
```

`XX:XX:XX:XX:XX:XX`는 자동으로 할당된 MAC 주소입니다.

동일한 광 섬유 채널 패브릭에 연결된 모든 시스템의 컨트롤 도메인이 이더넷에도 연결되고 동일한 멀티캐스트 도메인의 일부인 경우 이 자동 할당 방법은 고유한 WWN을 생성합니다. 이 요구사항을 충족할 수 없는 경우 SAN에 필요한 고유한 WWN을 수동으로 지정해야 합니다.

수동 WWN(world-wide name) 할당

모든 방법을 사용하여 고유한 WWN을 만들 수 있습니다. 이 절에서는 Logical Domains Manager 수동 MAC 주소 할당 풀에서 WWN을 만드는 방법을 설명합니다. 할당하는 WWN의 고유성을 보장해야 합니다.

Logical Domains Manager에는 `00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF` 범위에서 수동 할당에 사용할 수 있는 256,000개의 MAC 주소 풀이 있습니다.

다음 예에서는 `00:14:4F:FC:00:01` MAC 주소를 기준으로 `port-wwn` 및 `node-wwn` 등록 정보 값을 보여줍니다.

```
port-wwn = 10:00:00:14:4F:FC:00:01
node-wwn = 20:00:00:14:4F:FC:00:01
```

`00:14:4F:FC:00:01`은 수동으로 할당된 MAC 주소입니다. 자동 MAC 주소 할당에 대한 자세한 내용은 [“자동 또는 수동 MAC 주소 지정” \[237\]](#)을 참조하십시오.

주 - SAN 스토리지의 예측 가능한 구성을 위해서는 WWN을 수동으로 지정하는 것이 가장 좋습니다.

모든 시스템이 이더넷으로 동일한 멀티캐스트 도메인에 연결되지 않았을 때 수동 WWN 할당 방법을 사용해야 합니다. 또한 이 방법을 사용하면 광 섬유 채널 가상 기능을 삭제하고 다시 만들 때 동일한 WWN이 사용되도록 보장할 수 있습니다.

광 섬유 채널 SR-IOV 가상 기능 만들기

이 절에서는 동적으로 가상 기능을 만들고 삭제하는 방법에 대해 설명합니다. 동적 방법을 사용하여 해당 작업을 수행할 수 없을 경우 가상 기능을 만들거나 삭제하기 전에 루트 도메인에서 지연된 재구성을 시작하십시오.

▼ 광 섬유 채널 SR-IOV 가상 기능을 만드는 방법

이 동적 방법을 사용할 수 없는 경우 정적 방법을 대신 사용하십시오. “정적 SR-IOV” [80]를 참조하십시오.

1. 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

물리적 기능의 이름에는 PCIe SR-IOV 카드 또는 내장 장치에 대한 위치 정보가 포함되어 있습니다.

2. 물리적 기능을 가진 버스에 대한 I/O 가상화가 아직 사용으로 설정되지 않은 경우 사용으로 설정합니다.

물리적 기능을 가진 버스에 대한 I/O 가상화가 사용으로 설정되지 않은 경우에만 이 단계를 수행합니다.

[PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법 \[83\]](#)을 참조하십시오.

3. 물리적 기능에서 동적이나 정적으로 단일 가상 기능 또는 여러 가상 기능을 만듭니다.

하나 이상의 가상 기능을 만든 후 게스트 도메인에 지정할 수 있습니다.

■ 동적 방법:

- 물리적 기능에서 모두 동시에 여러 가상 기능을 만들려면 다음 명령을 사용합니다.

```
primary# ldm create-vf -n number | max pf-name
```

`ldm create-vf -n max` 명령을 사용하여 한 번에 해당 물리적 기능에 대한 모든 가상 기능을 만듭니다. 이 명령은 각 가상 기능에 대한 포트 및 노드 WWN을 자동으로 할당하고 `bw-percent` 등록 정보를 기본값인 0으로 설정합니다. 이 값은 모든 가상 기능에 충분한 대역폭이 할당되도록 지정합니다.

팁 - 물리적 기능에 대한 모든 가상 기능을 한 번에 만듭니다. WWN을 수동으로 지정하려는 경우 먼저 모든 가상 기능을 만든 다음 `ldm set-io` 명령을 사용하여 각 가상 기능에 대한 WWN 값을 수동으로 지정합니다. 이 방법은 물리적 기능에서 가상 기능을 만들 때 상태 전환 횟수를 최소화합니다.

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

- 물리적 기능에서 하나의 가상 기능을 만들려면 다음 명령을 사용합니다.

```
ldm create-vf [bw-percent=value] [port-wwn=value node-wwn=value] pf-name
```

광 섬유 채널 클래스 관련 등록 정보 값을 수동으로 지정할 수도 있습니다.

주 - 때때로 OS가 IOV 장치를 프로브하는 동안 새로 만든 가상 기능을 즉시 사용하지 못할 수도 있습니다. `ldm list-io` 명령을 사용하여 상위 물리적 기능과 그 하위 가상 기능에서 Status 열에 INV 값이 있는지 여부를 확인합니다. 이 값이 보이면 `ldm list-io` 출력에서 Status 열에 INV 값이 더 이상 표시되지 않을 때까지 기다렸다가(약 45초 정도) 물리적 기능이나 하위 가상 기능을 사용하십시오. 이 상태가 지속되면 장치에 문제가 있는 것입니다.

`primary`를 포함한 루트 도메인 재부트 직후나 `ldm create-vf` 또는 `ldm destroy-vf` 명령을 사용한 직후에 장치 상태가 INV로 표시될 수 있습니다.

- 정적 방법:

- a. 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

- b. 물리적 기능에서 단일 가상 기능 또는 여러 가상 기능을 만듭니다.
앞에 나온 동일한 명령을 사용하여 가상 기능을 동적으로 만듭니다.

- c. 루트 도메인을 재부트합니다.

- 비`primary` 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# ldm stop-domain -r root-domain
```

- `primary` 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# shutdown -i6 -g0 -y
```

예 22 광 섬유 채널 물리적 기능에 대한 정보 표시

이 예는 `/SYS/MB/PCIE7/IOVFC.PF0` 물리적 기능에 대한 정보를 보여줍니다.

- 이 물리적 기능은 보드의 PCIe 슬롯 PCIe7에 있습니다.
- IOVFC 문자열은 물리적 기능이 광 섬유 채널 SR-IOV 장치임을 나타냅니다.

```
primary# ldm list-io
NAME                                     TYPE   BUS     DOMAIN  STATUS
```

```

-----
pci_0          BUS pci_0 primary IOV
pci_1          BUS pci_1 rootdom1 IOV
niu_0         NIU niu_0 primary
niu_1         NIU niu_1 primary
/SYS/MB/PCIE0 PCIE pci_0 primary OCC
/SYS/MB/PCIE2 PCIE pci_0 primary OCC
/SYS/MB/PCIE4 PCIE pci_0 primary OCC
/SYS/MB/PCIE6 PCIE pci_0 primary EMP
/SYS/MB/PCIE8 PCIE pci_0 primary EMP
/SYS/MB/SASHBA PCIE pci_0 primary OCC
/SYS/MB/NET0   PCIE pci_0 primary OCC
/SYS/MB/PCIE1 PCIE pci_1 rootdom1 OCC
/SYS/MB/PCIE3 PCIE pci_1 rootdom1 OCC
/SYS/MB/PCIE5 PCIE pci_1 rootdom1 OCC
/SYS/MB/PCIE7 PCIE pci_1 rootdom1 OCC
/SYS/MB/PCIE9 PCIE pci_1 rootdom1 OCC
/SYS/MB/NET2   PCIE pci_1 rootdom1 OCC
/SYS/MB/NET0/IOVNET.PF0 PF pci_0 primary
/SYS/MB/NET0/IOVNET.PF1 PF pci_0 primary
/SYS/MB/PCIE5/IOVNET.PF0 PF pci_1 rootdom1
/SYS/MB/PCIE5/IOVNET.PF1 PF pci_1 rootdom1
/SYS/MB/PCIE7/IOVFC.PF0 PF pci_1 rootdom1
/SYS/MB/PCIE7/IOVFC.PF1 PF pci_1 rootdom1
/SYS/MB/NET2/IOVNET.PF0 PF pci_1 rootdom1
/SYS/MB/NET2/IOVNET.PF1 PF pci_1 rootdom1

```

다음 명령은 지정된 물리적 기능에 대한 세부정보를 보여줍니다. maxvfs 값은 장치에서 지원하는 최대 가상 기능 수를 나타냅니다.

```

primary# ldm list-io -l /SYS/MB/PCIE7/IOVFC.PF0
NAME                                TYPE  BUS      DOMAIN  STATUS
-----
/SYS/MB/PCIE7/IOVFC.PF0            PF    pci_0    rootdom1
[pci@400/pci@1/pci@0/pci@6/SUNW,emlxs@0]
maxvfs = 8

```

예 23 선택적 등록 정보를 설정하지 않고 동적으로 광 섬유 채널 가상 기능 만들기

이 예에서는 선택적 등록 정보를 설정하지 않고 동적으로 가상 기능을 만듭니다. 이 경우 ldm create-vf 명령이 기본 대역폭 백분율, 포트 WWN(world-wide name) 및 노드 WWN 값을 자동으로 할당합니다.

I/O 가상화가 pci_1 PCIe 버스에서 사용으로 설정되어 있는지 확인합니다. PCIe 버스에 대한 I/O 가상화를 사용으로 설정하는 방법 [83]을 참조하십시오.

ldm create-vf 명령을 사용하여 /SYS/MB/PCIE7/IOVFC.PF0 물리적 기능에서 모든 가상 기능을 만들 수 있습니다.

```

primary# ldm create-vf -n max /SYS/MB/PCIE7/IOVFC.PF0
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF0
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF1
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF2
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF3
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF4
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF5
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF6
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF7

```

예 24 동적으로 광 섬유 채널 가상 기능 만들기 및 등록 정보 설정

이 예에서는 가상 기능을 동적으로 만들면서 `bw-percent` 등록 정보 값을 25로 설정하고 포트 및 노드 WWN을 지정합니다.

```
primary# ldm create-vf port-wwn=10:00:00:14:4F:FC:00:01 \
node-wwn=20:00:00:14:4F:FC:00:01 bw-percent=25 /SYS/MB/PCIE7/IOVFC.PF0
```

예 25 선택적 등록 정보를 설정하지 않고 정적으로 광 섬유 채널 가상 기능 만들기

이 예에서는 선택적 등록 정보를 설정하지 않고 정적으로 가상 기능을 만듭니다. 이 경우 `ldm create-vf` 명령이 기본 대역폭 백분율, 포트 WWN(world-wide name) 및 노드 WWN 값을 자동으로 할당합니다.

먼저 `rootdom1` 도메인에서 지연된 재구성을 시작합니다. 그런 다음 `pci_1` PCIe 버스에서 I/O 가상화를 사용으로 설정합니다. `pci_1` 버스가 이미 `rootdom1` 루트 도메인에 지정되었으므로 `ldm set-io` 명령을 사용하여 I/O 가상화를 사용으로 설정합니다.

```
primary# ldm start-reconf rootdom1
Initiating a delayed reconfiguration operation on the rootdom1 domain.
All configuration changes for other domains are disabled until the rootdom1
domain reboots, at which time the new configuration for the rootdom1 domain
will also take effect.
```

```
primary# ldm set-io iov=on pci_1
```

이제 `ldm create-vf` 명령을 사용하여 `/SYS/MB/PCIE7/IOVFC.PF0` 물리적 기능에서 모든 가상 기능을 만들 수 있습니다.

```
primary# ldm create-vf -n max /SYS/MB/PCIE7/IOVFC.PF0
```

```
-----
Notice: The rootdom1 domain is in the process of a delayed reconfiguration.
Any changes made to the rootdom1 domain will only take effect after it reboots.
-----
```

```
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF0
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF1
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF2
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF3
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF4
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF5
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF6
Created new vf: /SYS/MB/PCIE7/IOVFC.PF0.VF7
```

마지막으로 다음 방법 중 하나로 `rootdom1` 루트 도메인을 재부트하여 변경사항을 적용합니다.

- `rootdom1`이 비`primary` 루트 도메인

```
primary# ldm stop-domain -r rootdom1
```

- `rootdom1`이 `primary` 도메인

```
primary# shutdown -i6 -g0 -y
```

광 섬유 채널 SR-IOV 가상 기능 삭제

가상 기능은 현재 도메인에 지정되지 않은 경우 삭제할 수 있습니다. 가상 기능은 만든 순서와 역순으로만 삭제할 수 있으므로 마지막으로 만들어진 가상 기능만 삭제할 수 있습니다. 이로 인한 구성은 물리적 기능 드라이버를 통해 검증됩니다.

▼ 광 섬유 채널 SR-IOV 가상 기능을 삭제하는 방법

이 동적 방법을 사용할 수 없는 경우 정적 방법을 대신 사용하십시오. “정적 SR-IOV” [80]를 참조하십시오.

1. 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

2. 동적이나 정적으로 단일 가상 기능 또는 여러 가상 기능을 삭제합니다.

- 동적 방법:

- 물리적 기능에서 모든 가상 기능을 한 번에 삭제하려면 다음 명령을 사용합니다.

```
primary# ldm destroy-vf -n number | max pf-name
```

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

`ldm destroy-vf -n max` 명령을 사용하여 한 번에 해당 물리적 기능에 대한 모든 가상 기능을 삭제합니다.

`number`를 `-n` 옵션에 대한 인수로 지정할 경우 가상 기능의 마지막 `number`가 삭제됩니다. 하나의 물리적 기능 장치 드라이버 상태 전환에서만 이 작업을 수행하므로 이 방법을 사용하십시오.

- 지정된 가상 기능을 삭제하려면 다음과 같이 하십시오.

```
primary# ldm destroy-vf vf-name
```

영향을 받는 하드웨어 장치와 OS의 지연 때문에, 영향을 받는 물리적 기능과 나머지 하위 가상 기능을 즉시 사용하지 못할 수도 있습니다. `ldm list-io` 명령을 사용하여 상위 물리적 기능과 그 하위 가상 기능에서 Status 열에 INV 값이 있는지 여부를 확인합니다. 이 값이 보이면 `ldm list-io` 출력에서 Status 열에 INV 값이 더 이상 표시되지 않을 때까지 기다리십시오(약 45초 정도). 그 후에 물리적 기능이나 하위 가상 기능을 안전하게 사용할 수 있습니다. 이 상태가 지속되면 장치에 문제가 있는 것입니다.

`primary`를 포함한 루트 도메인 재부트 직후나 `ldm create-vf` 또는 `ldm destroy-vf` 명령을 사용한 직후에 장치 상태가 INV로 표시될 수 있습니다.

- 정적 방법:

a. 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf root-domain-name
```

b. 단일 가상 기능 또는 여러 가상 기능을 삭제합니다.

- 지정된 물리적 기능에서 모든 가상 기능을 동시에 삭제하려면 다음 명령을 사용합니다.

```
primary# ldm destroy-vf -n number | max pf-name
```

경로 이름 또는 익명 이름을 사용하여 가상 기능을 지정할 수 있습니다. 하지만 익명 이름을 사용하는 것이 좋습니다.

- 지정된 가상 기능을 삭제하려면 다음과 같이 하십시오.

```
primary# ldm destroy-vf vf-name
```

c. 루트 도메인을 재부트합니다.

- 비primary 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# ldm stop-domain -r root-domain
```

- primary 루트 도메인을 재부트하려면 다음과 같이 하십시오.

```
primary# shutdown -i6 -g0 -y
```

예 26 동적으로 여러 광 섬유 채널 SR-IOV 가상 기능 삭제

이 예는 /SYS/MB/PCIE5/IOVFC.PF1 물리적 기능에서 모든 가상 기능을 삭제하는 결과를 보여줍니다. ldm list-io 출력은 물리적 기능에 8개의 가상 기능이 있음을 보여줍니다. ldm destroy-vf -n max 명령은 모든 가상 기능을 삭제하고, 최종 ldm list-io 출력은 가상 기능이 하나도 남아 있지 않음을 보여줍니다.

```
primary# ldm list-io
...
/SYS/MB/PCIE5/IOVFC.PF1          PF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF0     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF1     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF2     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF3     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF4     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF5     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF6     VF      pci_1
/SYS/MB/PCIE5/IOVFC.PF1.VF7     VF      pci_1
primary# ldm destroy-vf -n max /SYS/MB/PCIE5/IOVFC.PF1
primary# ldm list-io
...
/SYS/MB/PCIE5/IOVFC.PF1          PF      pci_1
```

예 27 광 섬유 채널 가상 기능 삭제

이 예는 /SYS/MB/PCIE7/IOVFC.PF0 물리적 기능에서 가상 기능을 정적으로 삭제하는 방법을 보여줍니다.

```
primary# ldm start-reconf rootdom1
Initiating a delayed reconfiguration operation on the rootdom1 domain.
All configuration changes for other domains are disabled until the rootdom1
domain reboots, at which time the new configuration for the rootdom1 domain
will also take effect.

primary# ldm destroy-vf -n max /SYS/MB/PCIE7/IOVFC.PF0
primary# ldm stop-domain -r rootdom1
```

광 섬유 채널 SR-IOV 가상 기능 수정

ldm set-io 명령은 등록 정보 값을 변경하거나 새 등록 정보를 설정하여 가상 기능의 현재 구성을 수정합니다.

이 동적 방법을 사용할 수 없을 경우 정적 방법을 사용하십시오. [“정적 SR-IOV” \[80\]](#)를 참조하십시오.

ldm set-io 명령을 사용하여 bw-percent, port-wnn 및 node-wnn 등록 정보를 수정할 수 있습니다.

도메인에 가상 기능이 지정되는 동안 bw-percent 등록 정보만 동적으로 변경할 수 있습니다.

▼ 광 섬유 채널 SR-IOV 가상 기능 등록 정보를 수정하는 방법

1. 물리적 기능 장치를 식별합니다.

```
primary# ldm list-io
```

물리적 기능의 이름에는 PCIe SR-IOV 카드 또는 내장 장치에 대한 위치 정보가 포함되어 있습니다.

2. 가상 기능 등록 정보를 수정합니다.

```
ldm set-io [bw-percent=value] [port-wnn=value node-wnn=value] pf-name
```

언제라도 동적으로 변경할 수 있는 bw-percent 등록 정보 값과 달리, port-wnn 및 node-wnn 등록 정보 값은 가상 기능이 도메인에 지정되지 않은 경우에만 동적으로 수정할 수 있습니다.

예 28 광 섬유 채널 SR-IOV 가상 기능 등록 정보 수정

이 예에서는 지정된 가상 기능 /SYS/MB/PCIE7/IOVFC.PF0.VF0의 등록 정보를 수정하여 대역폭 백분율과 포트 및 노드 WWN 값을 지정합니다.

```
primary# ldm set-io port-wnn=10:00:00:14:4f:fc:f4:7c \
node-wnn=20:00:00:14:4f:fc:f4:7c bw-percent=25 /SYS/MB/PCIE7/IOVFC.PF0.VF0
```

I/O 도메인에서 광 섬유 채널 SR-IOV 가상 기능 추가 및 제거

▼ I/O 도메인에 광 섬유 채널 SR-IOV 가상 기능을 추가하는 방법

가상 기능을 동적으로 제거할 수 없는 경우 정적 방법을 사용하십시오. “정적 SR-IOV” [80]를 참조하십시오.

1. I/O 도메인에 추가할 가상 기능을 식별합니다.

```
primary# ldm list-io
```

2. 동적이나 정적으로 가상 기능을 추가합니다.

- 동적으로 가상 기능을 추가하려면 다음과 같이 하십시오.

```
primary# ldm add-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 추가할 도메인의 이름을 지정합니다.

도메인의 가상 기능에 대한 장치 경로 이름은 `list-io -l` 출력에 표시된 경로입니다.

- 정적으로 가상 기능을 추가하려면 다음과 같이 하십시오.

- a. 도메인을 중지한 다음 가상 기능을 추가합니다.

```
primary# ldm stop-domain domain-name
primary# ldm add-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 익명 이름을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 추가할 도메인의 이름을 지정합니다. 지정된 게스트는 비활성 또는 바인드 상태여야 합니다.

도메인의 가상 기능에 대한 장치 경로 이름은 `list-io -l` 출력에 표시된 경로입니다.

- b. 도메인을 다시 시작합니다.

```
primary# ldm start-domain domain-name
```

예 29 광 섬유 채널 가상 기능 추가

이 예는 동적으로 `ldg2` 도메인에 `/SYS/MB/PCIE7/IOVFC.PF0.VF0` 가상 기능을 추가하는 방법을 보여줍니다.

```
primary# ldm add-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 ldg2
```

동적으로 가상 기능을 추가할 수 없을 경우 정적 방법을 사용하십시오.

```
primary# ldm stop-domain ldg2
primary# ldm add-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 ldg2
primary# ldm start-domain ldg2
```

▼ I/O 도메인에서 광 섬유 채널 SR-IOV 가상 기능을 제거하는 방법

이 동적 방법을 사용할 수 없는 경우 정적 방법을 대신 사용하십시오. “정적 SR-IOV” [80]를 참조하십시오.



주의 - 도메인에서 가상 기능을 제거하기 전에 해당 도메인 부트에 반드시 필요한 가상 기능이 아닌지 확인하십시오.

1. I/O 도메인에서 제거할 가상 기능을 식별합니다.

```
primary# ldm list-io
```

2. 동적이나 정적으로 가상 기능을 제거합니다.

- 동적으로 가상 기능을 제거하려면 다음과 같이 하십시오.

```
primary# ldm remove-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 장치 익명을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 제거할 도메인의 이름을 지정합니다.

- 정적으로 가상 기능을 제거하려면 다음과 같이 하십시오.

- a. I/O 도메인을 중지합니다.

```
primary# ldm stop-domain domain-name
```

- b. 가상 기능을 제거합니다.

```
primary# ldm remove-io vf-name domain-name
```

*vf-name*은 가상 기능의 익명 이름 또는 경로 이름입니다. 장치 익명을 사용하는 것이 좋습니다. *domain-name*은 가상 기능을 제거할 도메인의 이름을 지정합니다. 지정된 게스트는 비활성 또는 바인드 상태여야 합니다.

- c. I/O 도메인을 시작합니다.

```
primary# ldm start-domain domain-name
```

예 30 동적으로 광 섬유 채널 가상 기능 제거

이 예는 동적으로 ldg2 도메인에서 /SYS/MB/PCIE7/IOVFC.PF0.VF0 가상 기능을 제거하는 방법을 보여줍니다.

```
primary# ldm remove-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 1dg2
```

명령이 성공하면 1dg2 도메인에서 가상 기능이 제거됩니다. 1dg2가 다시 시작되면 지정된 가상 기능이 더 이상 해당 도메인에 나타나지 않습니다.

동적으로 가상 기능을 제거할 수 없을 경우 정적 방법을 사용하십시오.

```
primary# ldm stop-domain 1dg2
primary# ldm remove-io /SYS/MB/PCIE7/IOVFC.PF0.VF0 1dg2
primary# ldm start-domain 1dg2
```

고급 SR-IOV 항목: 광 섬유 채널 SR-IOV

이 절에서는 광 섬유 채널 SR-IOV 가상 기능 사용과 관련된 몇 가지 고급 항목에 대해 설명합니다.

게스트 도메인에서 광 섬유 채널 가상 기능 액세스

1dg2 콘솔 로그는 지정된 광 섬유 채널 가상 기능 장치의 작업을 보여줍니다. fcadm 명령을 사용하여 광 섬유 채널 가상 기능 장치를 보고 액세스합니다.

```
1dg2# fcadm hba-port
HBA Port WWN: 100000144ffb8a99
  Port Mode: Initiator
  Port ID: 13d02
  OS Device Name: /dev/cfg/c3
  Manufacturer: Emulex
  Model: 7101684
  Firmware Version: 7101684 1.1.60.1
  FCode/BIOS Version: Boot:1.1.60.1 Fcode:4.03a4
  Serial Number: 4925382+133400002R
  Driver Name: emlxs
  Driver Version: 2.90.15.0 (2014.01.22.14.50)
  Type: N-port
  State: online
  Supported Speeds: 4Gb 8Gb 16Gb
  Current Speed: 16Gb
  Node WWN: 200000144ffb8a99
  NPIV Not Supported
```

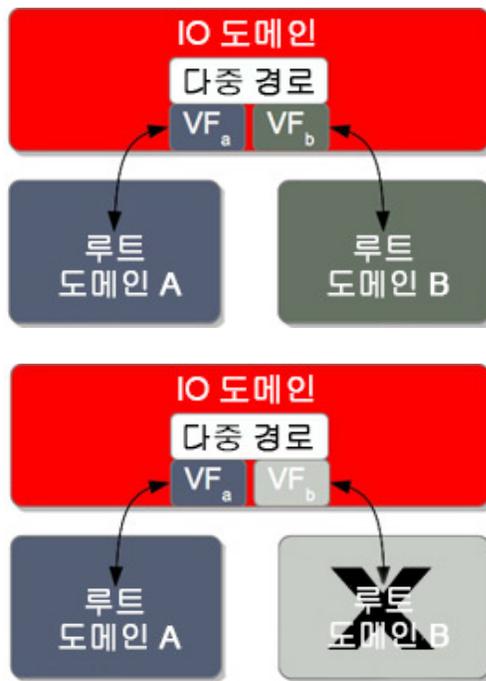
format 명령을 사용하여 표시 가능한 LUN을 표시합니다.

```
1dg2# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-25.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c3t21000024FF4C4BF8d0 <SUN-COMSTAR-1.0-10.00GB>
     /pci@340/pci@1/pci@0/pci@6/SUNw,emlxs@0,2/fp@0,0/ssd@w21000024ff4c4bf8,0
Specify disk (enter its number): ^D
1dg2#
```

I/O 도메인 복원성

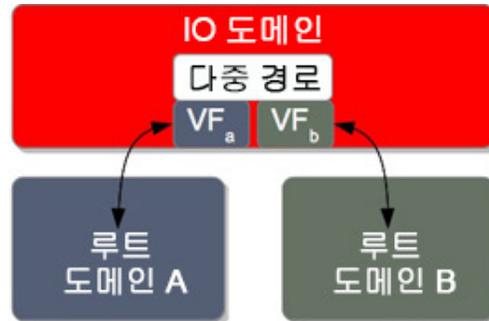
I/O 도메인 복원성은 연관된 루트 도메인 중 하나가 중단되더라도 실행을 계속하여 I/O 도메인의 가용성과 성능을 향상시킵니다. 루트 도메인이 중단되면 그 서비스를 사용하는 I/O 도메인은 영향을 받는 장치를 대체 I/O 경로로 페일오버하여 실행을 계속합니다. 루트 도메인에 서비스가 재개되면 복원형 I/O 도메인에서 영향을 받는 장치도 서비스가 재개되고 페일오버 기능이 복원됩니다.

다음 다이어그램은 구성된 루트 도메인 중 하나가 실패할 때의 상황과 루트 도메인에 서비스가 재개될 때의 상황을 보여주고 설명합니다.



각 루트 도메인은 I/O 도메인에 가상 기능을 제공합니다. I/O 도메인은 네트워크 장치용 IPMP와 광 섬유 채널 장치용 Oracle Solaris I/O 다중 경로와 같은 가상 장치 다중 경로를 사용합니다.

패닉이나 재부트로 인해 루트 도메인 B가 중단되면 가상 기능 B가 I/O 도메인에서 일시 중지되고 다중 경로가 작동하여 루트 도메인 A를 통해 모든 I/O 경로를 지정합니다.



루트 도메인 B에 서비스가 복원되면 가상 기능 B가 I/O 도메인에서 작동을 재개합니다. 다중 경로 그룹이 전체 중복성으로 복원됩니다.

이 구성에서 가상 기능은 가상 네트워크 장치 또는 가상 저장 장치일 수 있으며, 이는 가상 기능이나 가상 장치의 어떤 조합이든 I/O 도메인을 구성할 수 있음을 의미합니다.

복원형 및 비복원형 I/O 도메인을 모두 포함한 구성을 만들 수 있습니다. 예제는 “예제 - 복원형 및 비복원형 구성 사용” [133]을 참조하십시오.

복원형 I/O 도메인 요구사항

주 - Oracle Solaris 10 OS는 I/O 도메인 복원성을 제공하지 않습니다.

복원형 I/O 도메인은 다음 요구사항을 충족해야 합니다.

- 최소 Oracle Solaris 11.2 SRU 8 OS를 실행하고 `primary` 도메인은 최소 Oracle VM Server for SPARC 3.2 소프트웨어를 실행합니다.
- 다중 경로를 사용하여 가상 기능 및 가상 장치에 대한 페일오버 구성을 만듭니다. 이 구성이 가능하려면 가상 기능과 가상 장치가 동일 클래스(네트워크 또는 저장소)에 속해야 합니다.
- `master` 등록 정보 값이 루트 도메인 이름으로 설정되고, 이 도메인의 `failure-policy` 등록 정보는 `ignore`로 설정됩니다. 그밖에 `stop`, `reset`, `panic`과 같은 실패 정책 설정은 I/O 복원성보다 우선하며 I/O 도메인이 중단됩니다.
- I/O 도메인 복원성을 지원하는 SR-IOV 가상 기능, 가상 네트워크 장치, 가상 저장 장치만 사용합니다. <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1>을 참조하십시오.

I/O 도메인 복원성 제한 사항

- 루트 도메인에 SR-IOV 카드를 핫 플러그하고 I/O 도메인에 가상 기능을 지정하면, 루트 도메인이 실패할 때 I/O 도메인이 복원성을 제공하지 못할 수 있습니다. 따라서 루트 도메인

의 작동이 중지되는 동안에만 SR-IOV 카드를 추가해야 합니다. 그리고 루트 도메인이 부트된 후에 가상 기능을 지정합니다.

- 복원형 I/O 도메인이 있고 다음 방법 중 하나로 장치를 지정하면 I/O 도메인에 복원성이 사라집니다.
 - I/O 복원성을 지원하지 않는 카드에서 가상 기능 추가
 - 직접 I/O 기능을 사용하여 장치 직접 지정

이 경우 failure-policy를 ignore에서 reset 또는 stop으로 설정합니다.

복원형 I/O 도메인 구성

▼ 복원형 I/O 도메인을 구성하는 방법

시작하기 전에 I/O 도메인 복원성 기능을 지원하는 PCIe 카드만 사용합니다. <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1>을 참조하십시오.

I/O 도메인, 루트 도메인, 서비스 도메인, primary 도메인에서 최소 Oracle Solaris 11.2 SRU 8 OS와 Logical Domains Manager 3.2 소프트웨어가 실행되어야 합니다.

1. 루트 도메인에서 failure-policy 등록 정보를 ignore로 설정합니다.

```
primary# ldm set-domain failure-policy=ignore root-domain-name
```

주 - 복원성을 지원하지 않는 장치를 I/O 도메인에 추가하면 해당 도메인에 복원성이 사라집니다. 따라서 failure-policy 등록 정보 값을 stop, reset 또는 panic으로 재설정합니다.

도메인 종속성에 대한 자세한 내용은 “도메인 종속성 구성” [356]을 참조하십시오.

2. I/O 도메인에서 master 등록 정보를 루트 도메인 이름으로 설정합니다.

```
primary# ldm set-domain master=root-domain-name I/O-domain-name
```

3. 경로를 가로질러 다중 경로를 구성합니다.

- 이더넷. IPMP를 사용하여 경로를 가로질러 다중 경로를 구성합니다.
IPMP를 사용하여 다중 경로를 구성하는 방법은 [Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.3](#)를 참조하십시오.
- 광 섬유 채널. Oracle Solaris I/O 다중 경로를 사용하여 경로에 대해 다중 경로를 구성합니다.
Oracle Solaris I/O 다중 경로를 사용하여 다중 경로를 구성하는 방법은 [Managing SAN Devices and Multipathing in Oracle Solaris 11.3](#)를 참조하십시오.

예 31 IPMP를 사용하여 이더넷 SR-IOV 기능으로 다중 경로 구성

이 예제는 IPMP를 사용하여 복원형 I/O 도메인에 네트워크 가상 기능 장치를 구성하는 방법을 보여줍니다. 자세한 내용은 [Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.3](#)를 참조하십시오.

- 서로 다른 루트 도메인에 지정된 이더넷 SR-IOV 물리적 기능 2개를 식별합니다.
이 예제에서는 root-1 및 root-2 루트 도메인에 이더넷 SR-IOV 물리적 기능이 있습니다.

```
primary# ldm list-io | grep root-1 | grep PF
/SYS/PCI-EM8/IOVNET.PF0          PF      pci_1    root-1
primary# ldm list-io | grep root-2 | grep PF
/SYS/RIO/NET2/IOVNET.PF0        PF      pci_2    root-2
```

- 각각의 지정된 물리적 기능에 이더넷 가상 기능 2개를 만듭니다.

```
primary# ldm create-vf /SYS/MB/NET0/IOVNET.PF0
Created new vf: /SYS/PCI-EM8/IOVNET.PF0.VF0
primary# ldm create-vf /SYS/RIO/NET2/IOVNET.PF0
Created new vf: /SYS/RIO/NET2/IOVNET.PF0.VF0
```

- io-1 I/O 도메인에 이더넷 가상 기능을 지정합니다.

```
primary# ldm add-io /SYS/PCI-EM8/IOVNET.PF0.VF0 io-1
primary# ldm add-io /SYS/RIO/NET2/IOVNET.PF0.VF0 io-1
```

- I/O 도메인에서 이더넷 가상 기능을 IPMP 그룹으로 구성합니다.

- I/O 도메인에서 새로 추가된 네트워크 장치인 net1 및 net2를 식별합니다.

```
io-1# dladm show-phys
LINK          MEDIA          STATE  SPPED  DUPLEX  DEVICE
net0          Ethernet      up     0      unknown vnet0
net1          Ethernet      up     1000   full    igbvf0
net2          Ethernet      up     1000   full    igbvf1
```

- 새로 추가된 네트워크 장치에 대한 IP 인터페이스를 만듭니다.

```
io-1# ipadm create-ip net1
io-1# ipadm create-ip net2
```

- 2개의 네트워크 인터페이스에 대한 ipmp0 IPMP 그룹을 만듭니다.

```
io-1# ipadm create-ipmp -i net1 -i net2 ipmp0
```

- IPMP 그룹에 IP 주소를 지정합니다.
이 예제는 DHCP 옵션을 구성합니다.

```
io-1# ipadm create-addr -T dhcp ipmp0/v4
```

- IPMP 그룹 인터페이스의 상태를 확인합니다.

```
io-1# ipmpstat -g
```

예 32 Oracle Solaris I/O 다중 경로를 사용하여 광 섬유 채널 SR-IOV 기능으로 다중 경로 구성

이 예에서는 Oracle Solaris I/O 다중 경로를 사용하여 복원형 I/O 도메인에 대한 광 섬유 채널 가상 기능 장치를 구성하는 방법을 보여 줍니다. 자세한 내용은 [Managing SAN Devices and Multipathing in Oracle Solaris 11.3](#)를 참조하십시오.

1. 서로 다른 루트 도메인에 지정된 광 섬유 채널 SR-IOV 물리적 기능 2개를 식별합니다.
이 예제에서는 root-1 및 root-2 루트 도메인에 광 섬유 채널 SR-IOV 물리적 기능이 있습니다.

```
primary# ldm list-io | grep root-1 | grep PF
/SYS/PCI-EM4/IOVFC.PF0          PF      pci_1    root-1
primary# ldm list-io | grep root-2 | grep PF
/SYS/PCI-EM15/IOVFC.PF0       PF      pci_2    root-2
```

2. 각각의 지정된 물리적 기능에 가상 기능 2개를 만듭니다.
자세한 내용은 [광 섬유 채널 SR-IOV 가상 기능을 만드는 방법 \[118\]](#)을 참조하십시오.

```
primary# ldm create-vf port-wwn=10:00:00:14:4f:fc:60:00 \
node-wwn=20:00:00:14:4f:fc:60:00 /SYS/PCI-EM4/IOVFC.PF0
Created new vf: /SYS/PCI-EM4/IOVFC.PF0.VF0
primary# ldm create-vf port-wwn=10:00:00:14:4f:fc:70:00 \
node-wwn=20:00:00:14:4f:fc:70:00 /SYS/PCI-EM15/IOVFC.PF0
Created new vf: /SYS/PCI-EM15/IOVFC.PF0.VF0
```

3. io-1 I/O 도메인에 새로 만든 가상 기능을 추가합니다.

```
primary# ldm add-io /SYS/PCI-EM4/IOVFC.PF0.VF0 io-1
primary# ldm add-io /SYS/PCI-EM15/IOVFC.PF0.VF0 io-1
```

4. Oracle Solaris I/O 다중 경로가 `prtconf -v` 명령을 사용하여 I/O 도메인에서 사용으로 설정되었는지 여부를 확인합니다.

`fp` 장치에 대한 출력에 다음과 같은 장치 등록 정보 설정이 포함되어 있으면 Oracle Solaris I/O 다중 경로가 사용으로 설정됩니다.

```
mpxio-disable="no"
```

`mpxio-disable` 등록 정보가 `yes`로 설정된 경우 `/etc/driver/drv/fp.conf` 파일에서 등록 정보 값을 `no`로 업데이트하고 I/O 도메인을 재부트합니다.

`mpxio-disable` 장치 등록 정보가 `prtconf -v` 출력에 나타나지 않으면 `/etc/driver/drv/fp.conf` 파일에 `mpxio-disable="no"` 입력을 추가하고 I/O 도메인을 재부트합니다.

5. Oracle Solaris I/O 다중 경로 그룹의 상태를 확인합니다.

```
io-1# mpathadm show LU

Logical Unit: /dev/rdisk/c0t600A0B80002A384600003D6B544EECD0d0s2
  mpath-support: libmpscsi_vhci.so
  Vendor: SUN
  Product: CSM200_R
```

```

Revision: 0660
Name Type: unknown type
Name: 600a0b80002a384600003d6b544eecd0
Asymmetric: yes
Current Load Balance: round-robin
Logical Unit Group ID: NA
Auto Failback: on
Auto Probing: NA

Paths:
  Initiator Port Name: 100000144ffc6000
  Target Port Name: 201700a0b82a3846
  Override Path: NA
  Path State: OK
  Disabled: no

  Initiator Port Name: 100000144ffc7000
  Target Port Name: 201700a0b82a3846
  Override Path: NA
  Path State: OK
  Disabled: no

Target Port Groups:
  ID: 1
  Explicit Failover: yes
  Access State: active
  Target Ports:
    Name: 201700a0b82a3846
    Relative ID: 0

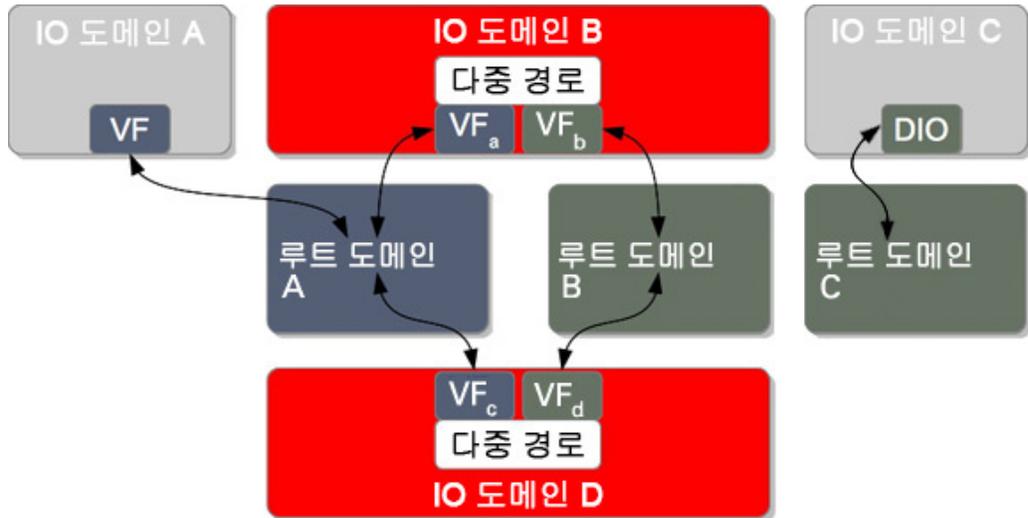
```

예제 - 복원형 및 비복원형 구성 사용

복원형 및 비복원형 도메인을 모두 포함한 구성을 사용할 수 있습니다.

다음 그림에서 I/O 도메인 A와 I/O 도메인 C는 어느 쪽도 다중 경로를 사용하지 않으므로 복원형이 아닙니다. I/O 도메인 A에 가상 기능이 있고 I/O 도메인 C에 직접 I/O 장치가 있습니다.

그림 4 복원형 및 비복원형 I/O 도메인을 포함한 구성



I/O 도메인 B와 I/O 도메인 D는 복원형입니다. I/O 도메인 A, B, D는 루트 도메인 A에 종속됩니다. I/O 도메인 B, D는 루트 도메인 B에 종속됩니다. I/O 도메인 C는 루트 도메인 C에 종속됩니다.

루트 도메인 A가 중단되면 I/O 도메인 A도 중단됩니다. 그러나 I/O 도메인 B, D는 대체 경로로 페일오버되어 응용 프로그램 실행을 계속합니다. 루트 도메인 C가 중단되면 루트 도메인 C의 *failure-policy* 등록 정보 값에 지정된 방법대로 I/O 도메인 C가 실패합니다.

비복원형 I/O 도메인이 구성된 상태로 루트 도메인 재부트

주 - I/O 도메인이 복원형인 경우 이를 서비스하는 루트 도메인이 중단되더라도 작동을 계속할 수 있습니다. 복원형 I/O 도메인 구성에 대한 자세한 내용은 [“I/O 도메인 복원성” \[128\]](#)을 참조하십시오.

I/O 도메인의 PCIe 슬롯과 마찬가지로 [“PCIe 끝점이 구성된 상태로 루트 도메인 재부트” \[140\]](#)에서 설명되는 사항은 I/O 도메인에 지정된 가상 기능과도 관련이 있습니다.

주 - 연관된 루트 도메인이 실행 중이 아니면 I/O 도메인을 시작할 수 없습니다.

직접 I/O를 사용하여 I/O 도메인 만들기

이 장에서는 다음과 같은 직접 I/O 항목을 다룹니다.

- “PCIe 끝점 장치를 지정하여 I/O 도메인 만들기” [135]
- “직접 I/O 하드웨어 및 소프트웨어 요구사항” [137]
- “현재 직접 I/O 기능 제한 사항” [138]
- “PCIe 끝점 장치 구성 계획” [139]
- “PCIe 끝점이 구성된 상태로 루트 도메인 재부트” [140]
- “PCIe 하드웨어 변경” [141]
- “PCIe 끝점 장치를 지정하여 I/O 도메인 만들기” [144]
- “직접 I/O 문제” [148]

PCIe 끝점 장치를 지정하여 I/O 도메인 만들기

도메인에 개별 PCIe 끝점(또는 직접 I/O 지정 가능) 장치를 지정할 수 있습니다. 이와 같이 PCIe 끝점 장치를 사용하면 I/O 도메인에 대한 장치 지정 단위가 증가합니다. 이 기능은 DIO (직접 I/O) 기능을 통해 전달됩니다.

DIO 기능을 사용하여 시스템에 있는 PCIe 버스 수보다 많은 I/O 도메인을 만들 수 있습니다. 그러면 가능한 I/O 도메인 수가 PCIe 끝점 장치 수에 의해서만 제한됩니다.

PCIe 끝점 장치는 다음 중 하나일 수 있습니다.

- 슬롯의 PCIe 카드
- 플랫폼이 식별한 내장 PCIe 장치

주 - 루트 도메인은 다른 루트 도메인에 대한 종속성을 가질 수 없으므로 PCIe 버스를 소유하는 루트 도메인은 다른 루트 도메인에 지정된 SR-IOV 가상 기능 또는 PCIe 끝점 장치를 가질 수 없습니다. 하지만 PCIe 버스의 PCIe 끝점 장치 또는 가상 기능을 해당 버스를 소유하는 루트 도메인에 지정할 수는 있습니다.

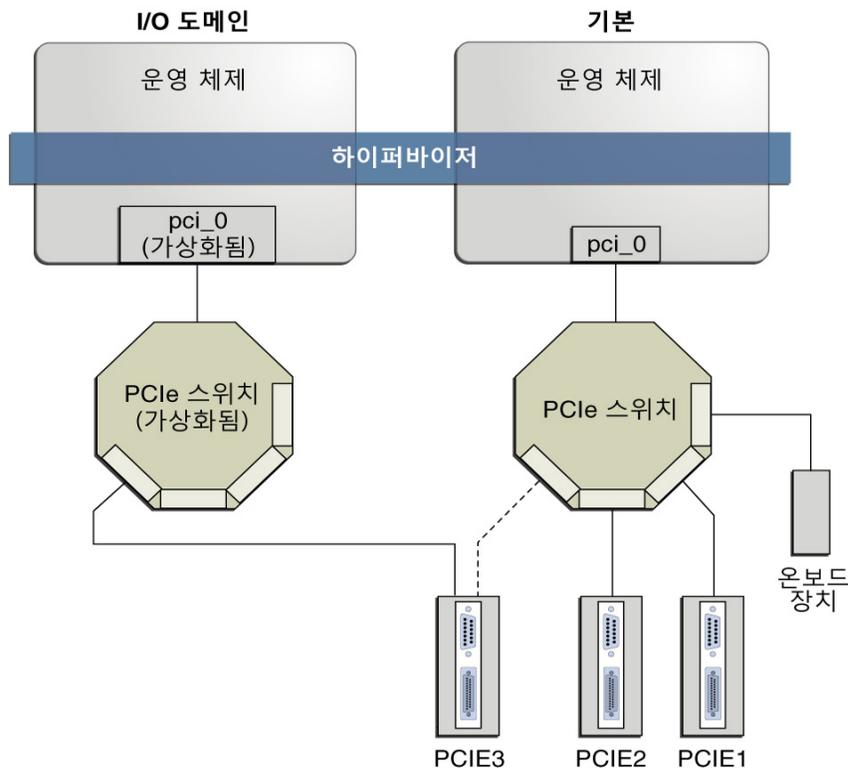
다음 다이어그램은 PCIe 끝점 장치 PCIe3이 I/O 도메인에 지정되었음을 보여줍니다. I/O 도메인의 버스 pci_0과 스위치는 모두 가상입니다. primary 도메인에서는 더 이상 PCIe3 끝점 장치에 액세스할 수 없습니다.

I/O 도메인에서 pci_0 블록과 스위치는 각각 가상 루트 컴플렉스와 가상 PCIe 스위치입니다. 이 블록과 스위치는 primary 도메인의 pci_0 블록 및 스위치와 유사합니다. primary 도메인에서 슬롯 PCIe3의 장치는 원래 장치의 “새도우” 형식이며 SUNW,assigned로 식별됩니다.



주의 - ldm remove-io 명령을 사용하여 primary 도메인에서 해당 장치를 제거한 후에는 Oracle Solaris 핫 플러그 작업을 사용하여 PCIe 끝점 장치를 작업 중 제거할 수 없습니다. PCIe 끝점 장치 교체 또는 제거에 대한 자세한 내용은 “PCIe 하드웨어 변경” [141]을 참조하십시오.

그림 5 I/O 도메인에 PCIe 끝점 장치 지정



PCIe 끝점 장치를 나열하려면 ldm list-io 명령을 사용하십시오.

DIO 기능이 슬롯의 PCIe 카드를 I/O 도메인에 지정할 수 있도록 허용하기는 하지만 특정 PCIe 카드만 지원됩니다. “[직접 I/O 하드웨어 및 소프트웨어 요구사항](#)” [137]을 참조하십시오.



주의 - 브릿지가 있는 PCIe 카드는 지원되지 않습니다. PCIe 기능 레벨 지정도 지원되지 않습니다. I/O 도메인에 지원되지 않는 PCIe 카드를 지정하면 예상하지 못한 동작이 발생할 수 있습니다.

다음 항목은 DIO 기능에 대한 중요한 세부정보를 설명합니다.

- 이 기능은 모든 소프트웨어 요구사항이 충족되는 경우에만 사용으로 설정됩니다. “[직접 I/O 하드웨어 및 소프트웨어 요구사항](#)” [137]을 참조하십시오.
- 루트 도메인에 지정된 PCIe 버스와 연결되어 있는 PCIe 끝점만 DIO 기능을 사용하는 다른 도메인에 지정할 수 있습니다.
- DIO를 사용하는 I/O 도메인은 루트 도메인이 실행 중인 경우에만 PCIe 끝점 장치에 대한 액세스 권한을 가집니다.
- 루트 도메인을 재부트하면 PCIe 끝점 장치가 있는 I/O 도메인이 영향을 받습니다. “[PCIe 끝점이 구성된 상태로 루트 도메인 재부트](#)” [140]를 참조하십시오. 루트 도메인은 다음과 같은 작업도 수행합니다.
 - PCIe 버스를 초기화 및 관리합니다.
 - I/O 도메인에 지정된 PCIe 끝점 장치로 트리거된 모든 버스 오류를 처리합니다. primary 도메인만 모든 PCIe 버스 관련 오류를 수신합니다.

직접 I/O 하드웨어 및 소프트웨어 요구사항

DIO(직접 I/O) 기능을 성공적으로 사용하여 직접 I/O 장치를 도메인에 지정하려면 적합한 소프트웨어를 실행하고 지원되는 PCIe 카드를 사용해야 합니다.

- 하드웨어 요구사항. 특정 PCIe 카드만 I/O 도메인에서 직접 I/O 끝점 장치로 사용할 수 있습니다. Oracle VM Server for SPARC 환경에서 다른 카드를 사용할 수도 있지만 DIO 기능에는 사용할 수 없습니다. 대신 전체 루트 컴플렉스가 지정된 I/O 도메인 및 서비스 도메인에 대해 사용할 수 있습니다.

플랫폼에서 사용할 수 있는 카드를 확인하려면 해당 플랫폼의 하드웨어 설명서를 참조하십시오. 지원되는 PCIe 카드에 대한 최신 목록을 보려면 <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1>을 참조하십시오.

주 - SPARC T7 시리즈 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에는 PCIe 버스를 여러 개 제공하는 I/O 컨트롤러가 있으므로, PCIe 카드를 여러 도메인에 지정할 수 있습니다. 자세한 내용은 7장. [PCIe 버스를 지정하여 루트 도메인 만들기](#)를 참조하십시오.

- 소프트웨어 요구사항. DIO 기능을 사용하려면 다음 도메인에서 지원되는 OS를 실행해야 합니다.
 - 루트 도메인. 최소 Oracle Solaris 11.3 OS.
권장 사용법은 모든 도메인에서 최소한 Oracle Solaris 10 1/13 OS와 [Oracle VM Server for SPARC 3.4 설치 설명서](#)의 “정규화된 Oracle Solaris OS 버전”에 설명된 필수 패치 또는 Oracle Solaris 11.3 OS를 실행하는 것입니다.
 - I/O 도메인. 최소 Oracle Solaris 11 OS. 추가 기능 지원이 최신 Oracle Solaris 11 릴리스에 포함됩니다.

주 - 플랫폼에서 지원되는 모든 PCIe 카드는 루트 도메인에서 지원됩니다. 지원되는 PCIe 카드 목록은 사용 중인 플랫폼에 대한 설명서를 참조하십시오. 단, 직접 I/O가 지원하는 PCIe 카드만 I/O 도메인에 지정할 수 있습니다.

직접 I/O 기능을 사용하여 PCIe 끝점 장치를 추가하거나 제거하려면 먼저 PCIe 버스 자체에서 I/O 가상화를 사용으로 설정해야 합니다.

ldm set-io 또는 ldm add-io 명령을 사용하여 iov 등록 정보를 on으로 설정할 수 있습니다. 또한 ldm add-domain 또는 ldm set-domain 명령을 사용하여 rc-add-policy 등록 정보를 iov로 설정할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

루트 도메인을 재부트하면 직접 I/O가 영향을 받으므로 루트 도메인에 대한 직접 I/O 관련 변경사항 적용을 최대화하고 루트 도메인 재부트를 최소화하도록 신중하게 직접 I/O 구성 변경을 계획해야 합니다.

현재 직접 I/O 기능 제한 사항

제한 사항을 해결하는 방법은 “[PCIe 끝점 장치 구성 계획](#)” [139]을 참조하십시오.

PCIe 끝점 장치를 비루트 도메인에 지정하거나 비루트 도메인에서 제거하는 작업은 해당 도메인이 중지되거나 비활성 상태인 경우에만 허용됩니다.

주 - Fujitsu M10 서버는 PCIe 끝점 장치의 동적 재구성을 지원합니다. 루트 도메인을 재부트하거나 I/O 도메인을 중지하지 않고도 PCIe 끝점 장치를 지정/제거할 수 있습니다.

이 기능에 대한 최신 정보는 <http://www.fujitsu.com/global/services/computing/server/sparc/downloads/manual/>에서 해당 모델의 *Fujitsu M10/SPARC M10 Systems System Operation and Administration Guide*를 참조하십시오.

주 - 직접 I/O 기능은 SPARC M7 시리즈 서버, SPARC T7 시리즈 서버 및 SPARC S7 시리즈 서버에서 지원되지 않습니다. 대신 PCIe 버스 지정 기능을 사용하십시오. [7장. PCIe 버스를 지정하여 루트 도메인 만들기](#)를 참조하십시오.

SPARC T5 및 SPARC M6 플랫폼을 비롯한 SPARC 시스템은 한정된 수의 인터럽트를 제공하므로 Oracle Solaris는 각 장치가 사용할 수 있는 인터럽트 수를 제한합니다. 기본 제한은 표준 시스템 구성 요구에 맞아야 하지만, 특정 시스템 구성에 따라 이 값을 조정해야 할 수도 있습니다. 자세한 내용은 “[인터럽트 제한 조정](#)” [369]을 참조하십시오.

PCIe 끝점 장치 구성 계획

루트 도메인 작동 중지 시간이 발생하지 않도록 PCIe 끝점 장치를 지정하거나 제거할 때는 사전에 신중하게 계획하십시오. 루트 도메인 재부트는 루트 도메인 자체에서 사용 가능한 서비스에만 영향을 끼치는 것이 아니라 PCIe 끝점 장치가 지정된 I/O 도메인에도 영향을 끼칩니다. 각 I/O 도메인에 대한 변경사항이 다른 도메인에 영향을 끼치지 않지만 사전 계획을 통해 해당 도메인이 제공하는 서비스에 끼치는 영향을 최소화할 수 있습니다.

지연된 재구성 상태에서는 더 많은 장치를 계속 추가하거나 제거한 다음 루트 도메인을 한 번만 재부트하여 모든 변경사항이 적용되도록 할 수 있습니다.

예는 [PCIe 끝점 장치를 지정하여 I/O 도메인을 만드는 방법](#) [144]을 참조하십시오.

DIO 장치 구성을 계획 및 수행하려면 다음과 같은 일반적인 단계를 따라야 합니다.

1. 시스템 하드웨어 구성을 이해하고 기록합니다.
 - 특히, 시스템에 있는 PCIe 카드의 부품 번호 및 기타 세부정보에 대한 내용을 기록합니다.
 - `ldm list-io -l` 및 `prtdiag -v` 명령을 사용하여 정보를 얻고 나중에 사용할 수 있도록 저장합니다.
2. `primary` 도메인에 있어야 할 PCIe 끝점 장치를 확인합니다.
 - 예를 들어, 다음에 대한 액세스를 제공하는 PCIe 끝점 장치를 확인합니다.
 - 부트 디스크 장치
 - 네트워크 장치
 - `primary` 도메인이 서비스로 제공하는 기타 장치
3. I/O 도메인에서 사용할 수 있는 모든 PCIe 끝점 장치를 제거합니다.
 - 이 단계를 수행하면 루트 도메인에서 I/O 도메인에 영향을 끼치는 후속 재부트 작업을 수행하지 않아도 됩니다.
 - `ldm remove-io` 명령을 사용하여 PCIe 끝점 장치를 제거합니다. `remove-io` 및 `add-io` 하위 명령에 장치를 지정할 때는 장치 경로보다 익명을 사용하는 것이 좋습니다.

주 - 지연된 재구성 중 원하는 모든 장치를 제거한 후 루트 도메인을 한 번만 재부트하면 모든 변경사항이 적용됩니다.

4. SP(서비스 프로세서)에 이 구성을 저장합니다.
 - `ldm add-config` 명령을 사용합니다.

5. 3단계에서 제거한 PCIe 끝점 장치가 해제되도록 루트 도메인을 재부트합니다.
6. 제거한 PCIe 끝점 장치가 더 이상 루트 도메인에 지정되어 있지 않은지 확인합니다.
`ldm list-io -l` 명령을 사용하여 제거한 장치가 출력에서 `SUNW,assigned-device`로 표시되는지 확인합니다.
7. 물리적 장치에 대한 직접 액세스 권한을 제공하도록 게스트 도메인에 사용 가능한 PCIe 끝점 장치를 지정합니다.
 이 지정이 완료되면 도메인 마이그레이션 기능을 통해 더 이상 다른 물리적 시스템에 게스트 도메인을 마이그레이션할 수 없습니다.
8. PCIe 끝점 장치를 게스트 도메인에 추가하거나 게스트 도메인에서 제거합니다.
`ldm add-io` 명령을 사용합니다.
 재부트 작업을 줄이고 해당 도메인이 제공하는 서비스의 작동 중지 시간이 발생하지 않도록 하여 I/O 도메인에 대한 변경을 최소화합니다.
9. (선택사항) PCIe 하드웨어를 변경합니다.
["PCIe 하드웨어 변경" \[141\]](#)을 참조하십시오.

PCIe 끝점이 구성된 상태로 루트 도메인 재부트

루트 도메인은 PCIe 버스의 소유자이며 버스 초기화 및 관리를 담당합니다. 루트 도메인은 활성 상태여야 하며 DIO 또는 SR-IOV 기능을 지원하는 Oracle Solaris OS 버전을 실행 중이어야 합니다. 루트 도메인을 종료, 정지 또는 재부트하면 PCIe 버스에 대한 액세스가 중단됩니다. PCIe 버스를 사용할 수 없는 경우 해당 버스의 PCIe 장치가 영향을 받아 이러한 장치를 사용하지 못할 수 있습니다.

PCIe 끝점 장치가 있는 I/O 도메인이 실행 중인 동안 루트 도메인이 재부트되면 해당 I/O 도메인의 동작을 예측할 수 없습니다. 예를 들어, 재부트 도중 또는 재부트 후 PCIe 끝점 장치가 있는 I/O 도메인에 패닉이 발생할 수 있습니다. 루트 도메인을 재부트할 때 수동으로 각 도메인을 중지하고 시작해야 합니다.

I/O 도메인이 복원형인 경우 PCIe 버스 소유자인 루트 도메인을 사용할 수 없게 되더라도 작동을 계속할 수 있습니다. ["I/O 도메인 복원성" \[128\]](#)을 참조하십시오.

주 - 연관된 루트 도메인이 실행 중이 아니면 I/O 도메인을 시작할 수 없습니다.

이러한 문제를 해결하려면 다음 단계 중 하나를 수행하십시오.

- 루트 도메인을 종료하기 전에 PCIe 끝점 장치가 지정된 시스템에서 수동으로 도메인을 종료합니다.
 이 단계를 수행하면 루트 도메인을 종료, 정지 또는 재부트하기 전에 해당 도메인이 완전히 종료됩니다.

PCIe 끝점 장치가 지정된 도메인을 모두 찾으려면 `ldm list-io` 명령을 실행합니다. 이 명령을 사용하면 시스템의 도메인에 지정된 PCIe 끝점 장치를 나열할 수 있습니다. 이 명령 출력에 대한 자세한 설명은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

발견된 각 도메인에 대해 `ldm stop` 명령을 실행하여 도메인을 중지합니다.

- 루트 도메인과 PCIe 끝점 장치가 지정된 도메인 간의 도메인 종속성 관계를 구성합니다.

이 종속성 관계를 사용하면 어떤 이유로든 루트 도메인이 재부트될 때 PCIe 끝점 장치가 있는 도메인이 자동으로 다시 시작됩니다.

이 종속성 관계는 해당 도메인을 강제로 재설정하며 완전히 종료할 수 없습니다. 단, 종속성 관계는 수동으로 종료된 도메인에는 영향을 끼치지 않습니다.

```
primary# ldm set-domain failure-policy=reset primary
primary# ldm set-domain master=primary domain-name
```

예 33 비primary 루트 도메인 및 I/O 도메인이 있는 구성에 대한 실패 정책 종속성 구성

다음 예에서는 비primary 루트 도메인 및 I/O 도메인이 있는 구성에서 실패 정책 종속성을 구성하는 방법에 대해 설명합니다.

이 예에서 `ldg1`은 비primary 루트 도메인입니다. `ldg2`는 PCIe SR-IOV 가상 기능이 있거나 `ldg1` 도메인 소유의 루트 컴플렉스에서 PCIe 끝점 장치가 지정된 I/O 도메인입니다.

```
primary# ldm set-domain failure-policy=stop ldg1
primary# ldm set-domain master=ldg1 ldg2
```

이 종속성 관계를 사용하면 `ldg1` 루트 도메인이 재부트될 때 I/O 도메인이 중지됩니다.

- 비primary 루트 도메인 재부트일 경우 이 종속성 관계에 따라 I/O 도메인이 중지됩니다. 비primary 루트 도메인이 부트된 후 I/O 도메인을 시작합니다.

```
primary# ldm start ldg2
```

- primary 도메인 재부트일 경우 이 정책 설정에 따라 비primary 루트 도메인과 종속 I/O 도메인이 중지됩니다. primary 도메인이 부트되면 먼저 비primary 루트 도메인을 시작해야 합니다. 도메인이 부트되면 I/O 도메인을 시작합니다.

```
primary# ldm start ldg1
```

`ldg1` 도메인이 활성 상태가 될 때까지 기다렸다가 I/O 도메인을 시작합니다.

```
primary# ldm start ldg2
```

PCIe 하드웨어 변경

다음 단계는 PCIe 끝점 지정을 잘못 구성하지 않도록 하는 데 유용합니다. 특정 하드웨어 설치 및 제거에 대한 플랫폼 관련 정보는 사용 중인 플랫폼에 대한 설명서를 참조하십시오.

- 빈 슬롯에 PCIe 카드를 설치 중인 경우 필요한 작업이 없습니다. 이 PCIe 카드는 PCIe 버스를 소유한 도메인이 자동으로 소유합니다.
I/O 도메인에 새 PCIe 카드를 지정하려면 `ldm remove-io` 명령을 사용하여 먼저 루트 도메인에서 카드를 제거하십시오. 그런 다음 `ldm add-io` 명령을 사용하여 I/O 도메인에 카드를 지정하십시오.
- PCIe 카드가 시스템에서 제거되어 루트 도메인에 지정된 경우 필요한 작업이 없습니다.
- I/O 도메인에 지정된 PCIe 카드를 제거하려면 먼저 I/O 도메인에서 장치를 제거하십시오. 그런 다음 시스템에서 물리적으로 장치를 제거하기 전에 루트 도메인에 장치를 추가하십시오.
- I/O 도메인에 지정된 PCIe 카드를 교체하려면 DIO 기능이 새 카드를 지원하는지 확인하십시오.
지원하는 경우 현재 I/O 도메인에 자동으로 새 카드가 지정되므로 필요한 작업이 없습니다.
지원하지 않는 경우 `ldm remove-io` 명령을 사용하여 먼저 I/O 도메인에서 해당 PCIe 카드를 제거하십시오. 다음으로 `ldm add-io` 명령을 사용하여 루트 도메인에 해당 PCIe 카드를 재지정하십시오. 그런 다음 루트 도메인에 지정된 PCIe 카드를 물리적으로 다른 PCIe 카드로 교체하십시오. 이러한 단계를 수행하면 DIO 기능이 지원하지 않는 구성을 피할 수 있습니다.

PCIe 카드를 제거할 때 게스트 도메인 가동 중단 최소화

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템에서 PCIe 카드를 제거하거나 교체하는 동안 이 하드웨어에 종속된 도메인을 사용할 수 없습니다. 이러한 게스트 도메인 가동 중단을 최소화하려면 핫 플러그 기능을 사용하여 카드를 물리적으로 제거할 수 있도록 시스템을 준비해야 합니다.

▼ PCIe 카드를 제거할 때 게스트 도메인 가동 중단을 최소화하는 방법

이 절차에 따라 직접 I/O 또는 SR-IOV 장치가 지정되지 않았고 다중 경로가 구성된 게스트 도메인의 가동 중단을 피할 수 있습니다. 이 절차에는 `primary` 도메인 재부트가 2번 필요합니다.

주 - PCIe 카드가 비`primary` 루트 도메인이 소유한 루트 컴플렉스에 있을 때는 이 절차가 적용되지 않습니다. 대신 [How to Replace PCIe Direct I/O Cards Assigned to an Oracle VM Server for SPARC Guest Domain \(Doc ID 1684273.1\)](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrcLoop=226878266536565&id=1684273.1&adf.ctrl-state=bo9fbmr1n_49) (https://support.oracle.com/epmos/faces/DocumentDisplay?_afrcLoop=226878266536565&id=1684273.1&adf.ctrl-state=bo9fbmr1n_49)을 참조하십시오.

1. **PCIe 슬롯이 지정된 게스트 도메인을 중지합니다.**
`primary# ldm stop domain-name`
2. **게스트 도메인에서 PCIe 슬롯을 제거합니다.**

```
primary# ldm remove-io PCIe-slot domain-name
```

3. **PCIe 슬롯과 SR-IOV 가상 기능이 지정된 게스트 도메인을 중지합니다.**

```
primary# ldm stop domain-name
```

주 - PCIe 버스가 지정된 게스트 도메인은 중지할 필요가 없습니다. 게스트 도메인에 네트워크 및 디스크 장치로 가는 대체 경로를 제공하고 있기 때문입니다.

4. 이 슬롯을 지정할 수 있도록 **primary** 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf primary
```

5. **primary** 도메인에 **PCIe** 슬롯을 추가합니다.

```
primary# ldm add-io PCIe-slot domain-name
```

6. **primary** 도메인을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

7. 핫 플러그 명령을 사용하여 **PCIe** 카드를 교체합니다.

Oracle Solaris OS 핫 플러그 기능에 대한 자세한 내용은 [Managing Devices in Oracle Solaris 11.3](#) 의 2 장, “Dynamically Configuring Devices”을 참조하십시오.

8. 카드를 교체한 후 이 동일 **PCIe** 슬롯을 게스트 도메인에 재지정하려면 다음 단계를 수행합니다.

- a. **primary** 도메인에서 지연된 재구성을 시작합니다.

```
primary# ldm start-reconf primary
```

- b. **primary** 도메인에서 **PCIe** 슬롯을 제거합니다.

```
primary# ldm remove-io PCIe-slot domain-name
```

- c. **primary** 도메인을 재부트하여 **PCIe** 슬롯 제거를 적용합니다.

```
primary# shutdown -i6 -g0 -y
```

- d. 게스트 도메인에 **PCIe** 슬롯을 재지정합니다.

```
primary# ldm add-io PCIe-slot domain-name
```

- e. **PCIe** 슬롯과 **SR-IOV** 가상 기능을 지정할 게스트 도메인을 시작합니다.

```
primary# ldm start domain-name
```

PCIe 끝점 장치를 지정하여 I/O 도메인 만들기

▼ PCIe 끝점 장치를 지정하여 I/O 도메인을 만드는 방법

작동 중지 시간이 최소화되도록 사전에 모든 DIO 배포를 계획하십시오.



주의 - DIO 도메인에 SPARC T3-1 또는 SPARC T4-1 시스템의 /SYS/MB/SASHBA1 슬롯을 지정할 경우 primary 도메인이 내장 DVD 장치에 대한 액세스 권한을 손실합니다.

SPARC T3-1 및 SPARC T4-1 시스템에는 /SYS/MB/SASHBA0 및 /SYS/MB/SASHBA1 경로로 표시되는 내장 저장소용 DIO 슬롯이 2개 있습니다. 멀티헤드 내장 디스크를 호스트하는 것 외에 /SYS/MB/SASHBA1 슬롯은 내장 DVD 장치를 호스트합니다. 따라서 DIO 도메인에 /SYS/MB/SASHBA1을 지정할 경우 primary 도메인이 내장 DVD 장치에 대한 액세스 권한을 손실합니다.

SPARC T3-2 및 SPARC T4-2 시스템에는 모든 내장 디스크와 내장 DVD 장치를 호스트하는 단일 SASHBA 슬롯이 있습니다. 따라서 DIO 도메인에 SASHBA를 지정할 경우 내장 디스크 및 내장 DVD 장치가 DIO 도메인에 대여되어 primary 도메인에서 사용할 수 없게 됩니다.

PCIe 끝점 장치를 추가하여 I/O 도메인을 만드는 예는 “[PCIe 끝점 장치 구성 계획](#)” [139]을 참조하십시오.

주 - 이 릴리스에서는 DefaultFixed NCP를 사용하여 Oracle Solaris 11 시스템에서 데이터 링크와 네트워크 인터페이스를 구성하십시오.

Oracle Solaris 11 OS에 포함된 NCP는 다음과 같습니다.

- DefaultFixed - dladm 또는 ipadm 명령을 사용하여 네트워킹을 관리할 수 있도록 해줍니다.
- Automatic - netcfg 또는 netadm 명령을 사용하여 네트워킹을 관리할 수 있도록 해줍니다.

netadm list 명령을 사용하여 DefaultFixed NCP가 사용으로 설정되어 있는지 확인하십시오. [Oracle Solaris Administration: Network Interfaces and Network Virtualization](#) 의 7 장, “Using Datalink and Interface Configuration Commands on Profiles”을 참조하십시오.

1. 현재 시스템에 설치된 장치를 식별하고 아카이브합니다.

ldm list-io -l 명령의 출력은 현재 I/O 장치가 구성된 방법을 보여줍니다. prttdiag -v 명령을 사용하여 보다 자세한 정보를 얻을 수 있습니다.

주 - I/O 도메인에 장치가 지정된 후에는 I/O 도메인에서만 장치 ID를 확인할 수 있습니다.

```
primary# ldm list-io -l
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
niu_0                               NIU   niu_0  primary
```

```

[niu@480]
niu_1                               NIU    niu_1    primary
[niu@580]
pci_0                                BUS    pci_0    primary
[pci@400]
pci_1                                BUS    pci_1    primary
[pci@500]
/SYS/MB/PCIE0                       PCIE   pci_0    primary  OCC
[pci@400/pci@2/pci@0/pci@8]
  SUNW,emlxs@0/fp/disk
  SUNW,emlxs@0/fp/tape
  SUNW,emlxs@0/fp@0,0
  SUNW,emlxs@0,1/fp/disk
  SUNW,emlxs@0,1/fp/tape
  SUNW,emlxs@0,1/fp@0,0
/SYS/MB/PCIE2                       PCIE   pci_0    primary  OCC
[pci@400/pci@2/pci@0/pci@4]
  pci/scsi/disk
  pci/scsi/tape
  pci/scsi/disk
  pci/scsi/tape
/SYS/MB/PCIE4                       PCIE   pci_0    primary  OCC
[pci@400/pci@2/pci@0/pci@0]
  ethernet@0
  ethernet@0,1
  SUNW,qlc@0,2/fp/disk
  SUNW,qlc@0,2/fp@0,0
  SUNW,qlc@0,3/fp/disk
  SUNW,qlc@0,3/fp@0,0
/SYS/MB/PCIE6                       PCIE   pci_0    primary  EMP
[pci@400/pci@1/pci@0/pci@8]
/SYS/MB/PCIE8                       PCIE   pci_0    primary  EMP
[pci@400/pci@1/pci@0/pci@c]
/SYS/MB/SASHBA                      PCIE   pci_0    primary  OCC
[pci@400/pci@2/pci@0/pci@e]
  scsi@0/iport@1
  scsi@0/iport@2
  scsi@0/iport@4
  scsi@0/iport@8
  scsi@0/iport@80/cdrom@p7,0
  scsi@0/iport@v0
/SYS/MB/NET0                         PCIE   pci_0    primary  OCC
[pci@400/pci@1/pci@0/pci@4]
  network@0
  network@0,1
/SYS/MB/PCIE1                       PCIE   pci_1    primary  OCC
[pci@500/pci@2/pci@0/pci@a]
  SUNW,qlc@0/fp/disk
  SUNW,qlc@0/fp@0,0
  SUNW,qlc@0,1/fp/disk
  SUNW,qlc@0,1/fp@0,0
/SYS/MB/PCIE3                       PCIE   pci_1    primary  OCC
[pci@500/pci@2/pci@0/pci@6]
  network@0
  network@0,1
  network@0,2
  network@0,3
/SYS/MB/PCIE5                       PCIE   pci_1    primary  OCC
[pci@500/pci@2/pci@0/pci@0]
  network@0
  network@0,1
/SYS/MB/PCIE7                       PCIE   pci_1    primary  EMP
[pci@500/pci@1/pci@0/pci@6]
/SYS/MB/PCIE9                       PCIE   pci_1    primary  EMP

```

```

[pci@500/pci@1/pci@0/pci@0]
/SYS/MB/NET2                    PCIE  pci_1  primary  OCC
[pci@500/pci@1/pci@0/pci@5]
  network@0
  network@0,1
  ethernet@0,80
/SYS/MB/NET0/IOVNET.PF0         PF    pci_0  primary
[pci@400/pci@1/pci@0/pci@4/network@0]
  maxvfs = 7
/SYS/MB/NET0/IOVNET.PF1         PF    pci_0  primary
[pci@400/pci@1/pci@0/pci@4/network@0,1]
  maxvfs = 7
/SYS/MB/PCIE5/IOVNET.PF0        PF    pci_1  primary
[pci@500/pci@2/pci@0/pci@0/network@0]
  maxvfs = 63
/SYS/MB/PCIE5/IOVNET.PF1        PF    pci_1  primary
[pci@500/pci@2/pci@0/pci@0/network@0,1]
  maxvfs = 63
/SYS/MB/NET2/IOVNET.PF0         PF    pci_1  primary
[pci@500/pci@1/pci@0/pci@5/network@0]
  maxvfs = 7
/SYS/MB/NET2/IOVNET.PF1         PF    pci_1  primary
[pci@500/pci@1/pci@0/pci@5/network@0,1]
  maxvfs = 7

```

2. 보존해야 할 부트 디스크의 장치 경로를 확인합니다.
[PCIe 버스를 지정하여 루트 도메인을 만드는 방법 \[68\]](#)의 2단계를 참조하십시오.
3. 블록 장치가 연결된 물리적 장치를 확인합니다.
[PCIe 버스를 지정하여 루트 도메인을 만드는 방법 \[68\]](#)의 3단계를 참조하십시오.
4. 시스템이 사용하는 네트워크 인터페이스를 확인합니다.
[PCIe 버스를 지정하여 루트 도메인을 만드는 방법 \[68\]](#)의 4단계를 참조하십시오.

5. 네트워크 인터페이스가 연결된 물리적 장치를 확인합니다.
 다음 명령은 igb0 네트워크 인터페이스를 사용합니다.

```

primary# ls -l /dev/igb0
lrwxrwxrwx 1 root root          46 Jul 30 17:29 /dev/igb0 ->
../devices/pci@500/pci@0/pci@8/network@0:igb0

```

이 예에서 primary 도메인이 사용하는 네트워크 인터페이스에 대한 물리적 장치는 1단계의 MB/NET0 목록에 해당하는 PCIe 끝점 장치(pci@500/pci@0/pci@8)에 연결되어 있습니다. 따라서 primary 도메인에서 이 장치를 제거하지 마십시오. primary 도메인이 사용하지 않는 기타 모든 PCIe 장치는 다른 도메인에 지정해도 됩니다.

primary 도메인이 사용하는 네트워크 인터페이스가 다른 도메인에 지정할 버스에 있을 경우 다른 네트워크 인터페이스를 사용하도록 primary 도메인을 재구성해야 합니다.

6. I/O 도메인에서 사용할 수 있는 PCIe 끝점 장치를 제거합니다.
 이 예에서는 primary 도메인이 사용하고 있지 않는 PCIE2, PCIE3, PCIE4 및 PCIE5 끝점 장치를 제거할 수 있습니다.

a. PCIe 끝점 장치를 제거합니다.

주의 - primary 도메인이 사용하거나 필요로 하는 장치는 제거하지 마십시오. 네트워크 포트나 usbecm 장치와 같이 도메인이 사용하는 장치가 있는 버스를 제거하지 마십시오.

실수로 잘못된 장치를 제거할 경우 `ldm cancel-reconf primary` 명령을 사용하여 primary 도메인에서 지연된 재구성을 취소하십시오.

여러 번 재부트하지 않아도 되도록 다중 장치를 한 번에 제거할 수 있습니다.

```
primary# ldm start-reconf primary
primary# ldm set-io iov=on pci_1
All configuration changes for other domains are disabled until the primary
domain reboots, at which time the new configuration for the primary domain
will also take effect.
primary# ldm remove-io /SYS/MB/PCIE1 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
primary# ldm remove-io /SYS/MB/PCIE3 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
primary# ldm remove-io /SYS/MB/PCIE5 primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

b. SP(서비스 프로세서)에 새 구성을 저장합니다.

다음 명령은 dio라는 파일에 구성을 저장합니다.

```
primary# ldm add-config dio
```

c. PCIe 끝점 장치 제거가 반영되도록 시스템을 재부트합니다.

```
primary# shutdown -i6 -g0 -y
```

7. primary 도메인에 로그인하고 PCIe 끝점 장치가 더 이상 도메인에 지정되어 있지 않은지 확인합니다.

```
primary# ldm list-io
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
niu_0                               NIU   niu_0    primary
niu_1                               NIU   niu_1    primary
pci_0                               BUS   pci_0    primary
pci_1                               BUS   pci_1    primary  IOV
/SYS/MB/PCIE0                       PCIE  pci_0    primary  OCC
/SYS/MB/PCIE2                       PCIE  pci_0    primary  OCC
/SYS/MB/PCIE4                       PCIE  pci_0    primary  OCC
/SYS/MB/PCIE6                       PCIE  pci_0    primary  EMP
/SYS/MB/PCIE8                       PCIE  pci_0    primary  EMP
```

```

/SYS/MB/SASHBA          PCIE  pci_0  primary OCC
/SYS/MB/NET0            PCIE  pci_0  primary OCC
/SYS/MB/PCIE1          PCIE  pci_1             OCC
/SYS/MB/PCIE3          PCIE  pci_1             OCC
/SYS/MB/PCIE5          PCIE  pci_1             OCC
/SYS/MB/PCIE7          PCIE  pci_1  primary EMP
/SYS/MB/PCIE9          PCIE  pci_1  primary EMP
/SYS/MB/NET2           PCIE  pci_1  primary OCC
/SYS/MB/NET0/IOVNET.PF0 PF    pci_0  primary
/SYS/MB/NET0/IOVNET.PF1 PF    pci_0  primary
/SYS/MB/NET2/IOVNET.PF0 PF    pci_1  primary
/SYS/MB/NET2/IOVNET.PF1 PF    pci_1  primary
    
```

주 - `ldm list-io -l` 출력에는 제거된 PCIe 끝점 장치가 `SUNW,assigned-device`로 표시될 수 있습니다. `primary` 도메인에서 더 이상 실제 정보를 사용할 수 없지만 장치가 지정된 도메인에는 이 정보가 들어 있습니다.

8. 도메인에 PCIe 끝점 장치를 지정합니다.

a. `ldg1` 도메인에 PCIe3 장치를 추가합니다.

```
primary# ldm add-io /SYS/MB/PCIE3 ldg1
```

b. 바인드 후 `ldg1` 도메인을 시작합니다.

```
primary# ldm bind ldg1
primary# ldm start ldg1
LDom ldg1 started
```

9. `ldg1` 도메인에 로그인하고 장치를 사용할 수 있는지 확인합니다.

네트워크 장치를 사용할 수 있는지 확인한 다음 도메인에서 사용할 네트워크 장치를 구성합니다.

```

primary# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      unknown  0      unknown  nxge0
net1          Ethernet      unknown  0      unknown  nxge1
net2          Ethernet      unknown  0      unknown  nxge2
net3          Ethernet      unknown  0      unknown  nxge3
    
```

직접 I/O 문제

`ldm list-io -l` 출력에서 PCI Express 이중 10Gbit 이더넷 파이버 카드에 4개의 하위 장치가 표시됨

PCI Express 이중 10Gbit 이더넷 파이버 카드(X1027A-Z)가 설치된 시스템에서 `ldm list-io -l` 명령을 실행할 경우 출력에 다음이 표시될 수 있습니다.

```
primary# ldm list-io -l
...
pci@500/pci@0/pci@c PCIE5 OCC primary
network@0
network@0,1
ethernet
ethernet
```

이 이더넷 카드에 포트가 2개 뿐이지만 출력에서는 하위 장치가 4개 있는 것으로 표시됩니다. 이러한 비정상 동작은 카드에 포함된 PCI 기능이 4개이기 때문에 발생합니다. 이러한 기능 중 2개는 내부적으로 사용 안함으로 설정되어 있고 `ldm ls-io -l` 출력에서 `ethernet`으로 표시됩니다.

`ldm list-io -l` 출력에서 `ethernet` 항목은 무시할 수 있습니다.

비primary 루트 도메인 사용

이 장에서는 다음과 같은 비primary 루트 도메인 항목을 다룹니다.

- “비primary 루트 도메인 개요” [151]
- “비primary 루트 도메인 요구사항” [152]
- “비primary 루트 도메인 제한 사항” [153]
- “비primary 루트 도메인 예” [154]

비primary 루트 도메인 개요

루트 도메인에는 PCIe 루트 컴플렉스가 지정됩니다. 이 도메인은 PCIe 패브릭을 소유하며 패브릭 오류 처리와 같은 모든 패브릭 관련 서비스를 제공합니다. 루트 도메인은 물리적 I/O 장치를 소유하고 직접 액세스를 제공하므로 I/O 도메인이기도 합니다. primary 도메인은 기본 루트 도메인입니다.

루트 도메인에 지정된 PCIe 버스에 대해 직접 I/O 및 SR-IOV 작업을 수행할 수 있습니다. 이제 비primary 루트 도메인을 비롯하여 모든 루트 도메인에 대해 다음 작업을 수행할 수 있습니다.

- PCIe 슬롯의 상태 표시
- 존재하는 SR-IOV 물리적 기능 표시
- I/O 도메인 또는 루트 도메인에 PCIe 슬롯 지정
- I/O 도메인 또는 루트 도메인에서 PCIe 슬롯 제거
- 물리적 기능에서 가상 기능 만들기
- 가상 기능을 삭제합니다
- 다른 도메인에 가상 기능 지정
- 다른 도메인에서 가상 기능 제거

Logical Domains Manager는 비primary 루트 도메인에서 실행되는 Logical Domains 에이전트로부터 PCIe 끝점 장치 및 SR-IOV 물리적 기능 장치를 가져옵니다. 루트 도메인이 처음 검색된 후 부트될 때까지 루트 도메인의 작동이 중지되는 동안 이 정보가 캐시됩니다.

루트 도메인이 활성 상태인 경우에만 직접 I/O 및 SR-IOV 작업을 수행할 수 있습니다. Logical Domains Manager는 해당 시점에 존재하는 실제 장치에서 작동합니다. 다음 작업이 발생하는 경우 캐시된 데이터가 새로고침될 수 있습니다.

- Logical Domains 에이전트가 지정된 루트 도메인에서 다시 시작되는 경우
- 지정된 루트 도메인에서 핫 플러그 작업과 같은 하드웨어 변경이 수행된 경우

`ldm list-io` 명령을 사용하여 PCIe 끝점 장치 상태를 확인할 수 있습니다. 출력은 각각의 비primary 루트 도메인이 소유한 루트 컴플렉스의 하위 장치 및 물리적 기능 장치도 보여줍니다.

루트 도메인에 다음 명령을 적용할 수 있습니다.

- `ldm add-io`
- `ldm remove-io`
- `ldm set-io`
- `ldm create-vf`
- `ldm destroy-vf`
- `ldm start-reconf`
- `ldm cancel-reconf`

지연된 재구성 지원이 비primary 루트 도메인을 포함하도록 확대되었습니다. 단, `ldm add-io`, `ldm remove-io`, `ldm set-io`, `ldm create-vf` 및 `ldm destroy-vf` 명령을 실행하는 데만 사용할 수 있습니다. 다음과 같은 동적 작업을 사용하여 완료할 수 없는 작업에 지연된 재구성을 사용할 수 있습니다.

- 직접 I/O 작업 수행
- 동적 SR-IOV 구성 요구사항을 충족하지 않는 물리적 기능을 기반으로 가상 기능 만들기 및 삭제



주의 - 작동 중지 시간이 최소화되도록 사전 계획을 통해 루트 도메인 재부트 횟수를 최소화하십시오.

비primary 루트 도메인 요구사항

비primary 루트 도메인을 컨트롤 도메인에 추가로 사용하여 다른 도메인에 직접 I/O 및 SR-IOV 기능을 제공할 수 있습니다. 이 기능은 SPARC T4 서버 및 Fujitsu M10 서버부터 지원됩니다.

- 하드웨어 요구사항.

[https://support.oracle.com/CSP/main/article?](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1)

[cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1325454.1)에 설명된 직접 I/O 및 SR-IOV용 PCIe

카드 외에도 다른 PCIe 카드를 사용할 수 있지만, DIO 및 SR-IOV에는 사용할 수 없습니다. 플랫폼에서 사용할 수 있는 카드를 확인하려면 해당 플랫폼의 하드웨어 설명서를 참조하십시오.

- **펌웨어 요구사항.**

SPARC T4 플랫폼은 최소 8.4.0.a 버전의 시스템 펌웨어를 실행해야 합니다.

SPARC T5 서버, SPARC M5 서버 및 SPARC M6 서버에서는 최소 9.1.0.x 버전의 시스템 펌웨어를 실행해야 합니다.

SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버에서는 최소 9.4.3 버전의 시스템 펌웨어를 실행해야 합니다.

Fujitsu M10 서버는 시스템 펌웨어 버전 XCP2210 이상을 실행해야 합니다.

- **소프트웨어 요구사항.**

비primary 도메인에서 최소 Oracle Solaris 11.2 OS를 실행해야 합니다.

비primary 루트 도메인 제한 사항

비primary 루트 도메인을 사용할 때는 다음 제한 사항이 적용됩니다.

- 연관된 루트 도메인이 실행 중이 아니면 I/O 도메인을 시작할 수 없습니다.
- 지연된 재구성에 대한 지원이 비primary 루트 도메인으로 확대되었습니다. 루트 도메인이 재부트되거나 지연된 재구성이 취소될 때까지 다음 명령만 실행할 수 있습니다.
 - `ldm add-io`
 - `ldm remove-io`
 - `ldm set-io`
 - `ldm create-vf`
 - `ldm destroy-vf`
- 다음 작업을 수행하려면 루트 도메인이 활성 상태여야 하며 부트되어야 합니다.
 - SR-IOV 가상 기능 만들기 및 삭제
 - PCIe 슬롯 추가 및 제거
 - SR-IOV 가상 기능 추가 및 제거
- PCIe 슬롯에 대한 `ldm add-io` 및 `ldm remove-io` 직접 I/O 작업을 수행하는 경우 루트 도메인에서 지연된 재구성을 시작해야 합니다.
- 구성이 동적 I/O 가상화 요구사항을 충족하지 않을 경우 다음과 같은 SR-IOV 가상 기능 작업에 대해 지연된 재구성을 사용해야 합니다.
 - `ldm create-vf`
 - `ldm destroy-vf`
 - `ldm add-io`
 - `ldm remove-io`

- `ldm set-io`
- 루트 도메인을 재부트하면 루트 도메인이 소유하는 PCIe 버스의 장치가 있는 모든 I/O 도메인이 영향을 받습니다. “PCIe 끝점이 구성된 상태로 루트 도메인 재부트” [140]를 참조하십시오.
- 루트 도메인 간에는 SR-IOV 가상 기능 또는 PCIe 슬롯을 지정할 수 없습니다. 이 제한 사항은 순환 종속성을 방지합니다.

비primary 루트 도메인 예

다음 예에서는 PCIe 버스에 대한 I/O 가상화를 사용으로 설정하고, 비primary 루트 도메인의 직접 I/O 장치를 관리하고, 비primary 루트 도메인의 SR-IOV 가상 기능을 관리하는 방법에 대해 설명합니다.

PCIe 버스에 대한 I/O 가상화 사용

다음 예에서는 `ldm add-io` 및 `ldm set-io` 명령을 사용하여 I/O 가상화를 사용으로 설정하는 방법을 보여줍니다.

다음 SPARC T4-2 I/O 구성은 버스 `pci_1`이 이미 primary 도메인에서 제거되었음을 보여줍니다.

```
primary# ldm list-io
NAME                                TYPE  BUS      DOMAIN  STATUS
----                                -
pci_0                               BUS   pci_0    primary IOV
pci_1                               BUS   pci_1
niu_0                               NIU   niu_0    primary
niu_1                               NIU   niu_1    primary
/SYS/MB/PCIE0                       PCIE  pci_0    primary OCC
/SYS/MB/PCIE2                       PCIE  pci_0    primary OCC
/SYS/MB/PCIE4                       PCIE  pci_0    primary OCC
/SYS/MB/PCIE6                       PCIE  pci_0    primary EMP
/SYS/MB/PCIE8                       PCIE  pci_0    primary EMP
/SYS/MB/SASHBA                      PCIE  pci_0    primary OCC
/SYS/MB/NET0                        PCIE  pci_0    primary OCC
/SYS/MB/PCIE1                       PCIE  pci_1
/SYS/MB/PCIE3                       PCIE  pci_1
/SYS/MB/PCIE5                       PCIE  pci_1
/SYS/MB/PCIE7                       PCIE  pci_1
/SYS/MB/PCIE9                       PCIE  pci_1
/SYS/MB/NET2                        PCIE  pci_1
/SYS/MB/NET0/IOVNET.PF0             PF    pci_0    primary
/SYS/MB/NET0/IOVNET.PF1             PF    pci_0    primary
```

다음 목록은 게스트 도메인이 바인드 상태임을 보여줍니다.

```
primary# ldm list
NAME                                STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
```

```

primary          active   -n-cv-  UART   8      8G      0.6%  0.6%  8m
rootdom1        bound   ----- 5000   8      4G
ldg2            bound   ----- 5001   8      4G
ldg3            bound   ----- 5002   8      4G

```

다음 `ldm add-io` 명령은 해당 버스에 대해 I/O 가상화가 사용으로 설정된 `rootdom1` 도메인에 `pci_1` 버스를 추가합니다. `ldm start` 명령은 `rootdom1` 도메인을 시작합니다.

```

primary# ldm add-io iov=on pci_1 rootdom1
primary# ldm start rootdom1
LDom rootdom1 started

```

지정된 PCIe 버스가 루트 도메인에 이미 지정된 경우 `ldm set-io` 명령을 사용하여 I/O 가상화를 사용으로 설정하십시오.

```

primary# ldm start-reconf rootdom1
primary# ldm set-io iov=on pci_1
primary# ldm stop-domain -r rootdom1

```

I/O 장치를 구성하려면 루트 도메인에서 해당 OS가 실행 중이어야 합니다. `rootdom1` 게스트 도메인의 콘솔에 연결한 다음 게스트 도메인이 자동 부트로 설정되지 않은 경우 `rootdom1` 루트 도메인의 OS를 부트하십시오.

```

primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Connecting to console "rootdom1" in group "rootdom1" ....
Press ~? for control options ..
ok> boot
...
primary#

```

다음 명령은 `pci_1` PCIe 버스 및 해당 하위 항목이 이제 `rootdom1` 루트 도메인의 소유가 되었음을 보여줍니다.

```

primary# ldm list-io
NAME                                     TYPE  BUS      DOMAIN  STATUS
----                                     -
pci_0                                    BUS   pci_0    primary IOV
pci_1                                    BUS   pci_1    rootdom1 IOV
niu_0                                    NIU   niu_0    primary
niu_1                                    NIU   niu_1    primary
/SYS/MB/PCIE0                           PCIE  pci_0    primary OCC
/SYS/MB/PCIE2                           PCIE  pci_0    primary OCC
/SYS/MB/PCIE4                           PCIE  pci_0    primary OCC
/SYS/MB/PCIE6                           PCIE  pci_0    primary EMP
/SYS/MB/PCIE8                           PCIE  pci_0    primary EMP
/SYS/MB/SASHBA                          PCIE  pci_0    primary OCC
/SYS/MB/NET0                             PCIE  pci_0    primary OCC
/SYS/MB/PCIE1                           PCIE  pci_1    rootdom1 OCC
/SYS/MB/PCIE3                           PCIE  pci_1    rootdom1 OCC
/SYS/MB/PCIE5                           PCIE  pci_1    rootdom1 OCC
/SYS/MB/PCIE7                           PCIE  pci_1    rootdom1 OCC
/SYS/MB/PCIE9                           PCIE  pci_1    rootdom1 EMP
/SYS/MB/NET2                             PCIE  pci_1    rootdom1 OCC
/SYS/MB/NET0/IOVNET.PF0                 PF    pci_0    primary
/SYS/MB/NET0/IOVNET.PF1                 PF    pci_0    primary
/SYS/MB/PCIE5/IOVNET.PF0                 PF    pci_1    rootdom1
/SYS/MB/PCIE5/IOVNET.PF1                 PF    pci_1    rootdom1
/SYS/MB/NET2/IOVNET.PF0                 PF    pci_1    rootdom1

```

```
/SYS/MB/NET2/IOVNET.PF1          PF      pci_1    rootdom1
```

비primary 루트 도메인에서 직접 I/O 장치 관리

다음 예에서는 비primary 루트 도메인에서 직접 I/O 장치를 관리하는 방법을 보여줍니다.

다음 명령을 실행하면 오류가 발생합니다. 루트 도메인에서 활성 상태인 슬롯을 제거하려고 시도하기 때문입니다.

```
primary# ldm remove-io /SYS/MB/PCIE7 ldg1
Dynamic I/O operations on PCIe slots are not supported.
Use start-reconf command to trigger delayed reconfiguration and make I/O
changes statically.
```

다음 명령은 먼저 루트 도메인에서 지연된 재구성을 시작하여 슬롯을 제거하는 올바른 방법을 보여줍니다.

```
primary# ldm start-reconf ldg1
Initiating a delayed reconfiguration operation on the ldg1 domain.
All configuration changes for other domains are disabled until the ldg1
domain reboots, at which time the new configuration for the ldg1 domain
will also take effect.
primary# ldm remove-io /SYS/MB/PCIE7 ldg1
-----
Notice: The ldg1 domain is in the process of a delayed reconfiguration.
Any changes made to the ldg1 domain will only take effect after it reboots.
-----
primary# ldm stop-domain -r ldg1
```

다음 `ldm list-io` 명령은 `/SYS/MB/PCIE7` 슬롯이 더 이상 루트 도메인에 없음을 확인합니다.

```
primary# ldm list-io
```

NAME	TYPE	BUS	DOMAIN	STATUS
pci_0	BUS	pci_0	primary	IOV
pci_1	BUS	pci_1	ldg1	IOV
niu_0	NIU	niu_0	primary	
niu_1	NIU	niu_1	primary	
/SYS/MB/PCIE0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE2	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE4	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE6	PCIE	pci_0	primary	EMP
/SYS/MB/PCIE8	PCIE	pci_0	primary	EMP
/SYS/MB/SASHBA	PCIE	pci_0	primary	OCC
/SYS/MB/NET0	PCIE	pci_0	primary	OCC
/SYS/MB/PCIE1	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE3	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE5	PCIE	pci_1	ldg1	OCC
/SYS/MB/PCIE7	PCIE	pci_1		OCC
/SYS/MB/PCIE9	PCIE	pci_1	ldg1	EMP
/SYS/MB/NET2	PCIE	pci_1	ldg1	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	ldg1	

다음 명령은 /SYS/MB/PCIE7 슬롯을 ldg2 도메인에 지정합니다. ldm start 명령은 ldg2 도메인을 시작합니다.

```
primary# ldm add-io /SYS/MB/PCIE7 ldg2
primary# ldm start ldg2
LDom ldg2 started
```

비primary 루트 도메인에서 SR-IOV 가상 기능 관리

다음 명령은 비primary 루트 도메인에 속하는 2개의 개별 물리적 기능을 기반으로 가상 기능을 2개 만듭니다.

```
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF0
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF1
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
```

-n 옵션을 사용하면 다음 두 명령을 사용하여 2개의 가상 기능을 만들 수도 있습니다.

```
primary# ldm create-vf -n 2 /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF1
primary# ldm create-vf -n 2 /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
```

지정된 물리적 기능에서 동적으로 가상 기능을 만들지 못한 경우 지연된 재구성을 시작하여 정적으로 가상 기능을 만드십시오.

```
primary# ldm start-reconf ldg1
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF0
primary# ldm create-vf /SYS/MB/PCIE5/IOVNET.PF0
Created new vf: /SYS/MB/PCIE5/IOVNET.PF0.VF1
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF0
primary# ldm create-vf /SYS/MB/NET2/IOVNET.PF1
Created new vf: /SYS/MB/NET2/IOVNET.PF1.VF1
```

그런 다음 변경사항이 적용되도록 루트 도메인 ldg1을 재부트하십시오.

```
primary# ldm stop-domain -r ldg1
```

다음 명령은 새 가상 기능을 보여줍니다.

```
primary# ldm list-io
NAME                                TYPE  BUS    DOMAIN  STATUS
----                                -
pci_0                               BUS   pci_0  primary IOV
pci_1                               BUS   pci_1  ldg1    IOV
niu_0                               NIU   niu_0  primary
niu_1                               NIU   niu_1  primary
```

```

/SYS/MB/PCIE0          PCIE  pci_0  primary OCC
/SYS/MB/PCIE2          PCIE  pci_0  primary OCC
/SYS/MB/PCIE4          PCIE  pci_0  primary OCC
/SYS/MB/PCIE6          PCIE  pci_0  primary EMP
/SYS/MB/PCIE8          PCIE  pci_0  primary EMP
/SYS/MB/SASHBA         PCIE  pci_0  primary OCC
/SYS/MB/NET0           PCIE  pci_0  primary OCC
/SYS/MB/PCIE1          PCIE  pci_1  ldg1   OCC
/SYS/MB/PCIE3          PCIE  pci_1  ldg1   OCC
/SYS/MB/PCIE5          PCIE  pci_1  ldg1   OCC
/SYS/MB/PCIE7          PCIE  pci_1  ldg2   OCC
/SYS/MB/PCIE9          PCIE  pci_1  ldg1   EMP
/SYS/MB/NET2           PCIE  pci_1  ldg1   OCC
/SYS/MB/NET0/IOVNET.PF0 PF    pci_0  primary
/SYS/MB/NET0/IOVNET.PF1 PF    pci_0  primary
/SYS/MB/PCIE5/IOVNET.PF0 PF    pci_1  ldg1
/SYS/MB/PCIE5/IOVNET.PF1 PF    pci_1  ldg1
/SYS/MB/NET2/IOVNET.PF0 PF    pci_1  ldg1
/SYS/MB/NET2/IOVNET.PF1 PF    pci_1  ldg1
/SYS/MB/PCIE5/IOVNET.PF0.VF0 VF    pci_1
/SYS/MB/PCIE5/IOVNET.PF0.VF1 VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF0 VF    pci_1
/SYS/MB/NET2/IOVNET.PF1.VF1 VF    pci_1

```

다음 명령은 동적으로 /SYS/MB/PCIE5/IOVNET.PF0.VF1 가상 기능을 ldg1 비primary 루트 도메인에 추가합니다.

```
primary# ldm add-io /SYS/MB/PCIE5/IOVNET.PF0.VF1 ldg1
```

다음 명령은 동적으로 /SYS/MB/NET2/IOVNET.PF1.VF0 가상 기능을 ldg2 도메인에 추가합니다.

```
primary# ldm add-io /SYS/MB/NET2/IOVNET.PF1.VF0 ldg2
```

다음 명령은 /SYS/MB/NET2/IOVNET.PF1.VF1 가상 기능을 바인드된 ldg3 도메인에 추가합니다.

```
primary# ldm add-io /SYS/MB/NET2/IOVNET.PF1.VF1 ldg3
primary# ldm start ldg3
LDom ldg3 started
```

ldg3 도메인의 콘솔에 연결한 다음 해당 OS를 부트하십시오.

다음 출력은 모든 지정 사항이 예상대로 표시됨을 보여줍니다. 하나의 가상 기능이 지정되지 않았으므로 동적으로 ldg1, ldg2 또는 ldg3 도메인에 지정할 수 있습니다.

```

# ldm list-io
NAME                TYPE  BUS      DOMAIN  STATUS
----                -
pci_0                BUS   pci_0    primary IOV
pci_1                BUS   pci_1    ldg1    IOV
niu_0                NIU   niu_0    primary
niu_1                NIU   niu_1    primary
/SYS/MB/PCIE0        PCIE  pci_0    primary OCC
/SYS/MB/PCIE2        PCIE  pci_0    primary OCC
/SYS/MB/PCIE4        PCIE  pci_0    primary OCC
/SYS/MB/PCIE6        PCIE  pci_0    primary EMP
/SYS/MB/PCIE8        PCIE  pci_0    primary EMP
/SYS/MB/SASHBA       PCIE  pci_0    primary OCC
/SYS/MB/NET0         PCIE  pci_0    primary OCC
/SYS/MB/PCIE1        PCIE  pci_1    ldg1    OCC
/SYS/MB/PCIE3        PCIE  pci_1    ldg1    OCC
/SYS/MB/PCIE5        PCIE  pci_1    ldg1    OCC

```

/SYS/MB/PCIE7	PCIE	pci_1	ldg2	OCC
/SYS/MB/PCIE9	PCIE	pci_1	ldg1	EMP
/SYS/MB/NET2	PCIE	pci_1	ldg1	OCC
/SYS/MB/NET0/IOVNET.PF0	PF	pci_0	primary	
/SYS/MB/NET0/IOVNET.PF1	PF	pci_0	primary	
/SYS/MB/PCIE5/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF0	PF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF1	PF	pci_1	ldg1	
/SYS/MB/PCIE5/IOVNET.PF0.VF0	VF	pci_1		
/SYS/MB/PCIE5/IOVNET.PF0.VF1	VF	pci_1	ldg1	
/SYS/MB/NET2/IOVNET.PF1.VF0	VF	pci_1	ldg2	
/SYS/MB/NET2/IOVNET.PF1.VF1	VF	pci_1	ldg3	

◆◆◆ 11 장

가상 디스크 사용

이 장에서는 Oracle VM Server for SPARC 소프트웨어에서 가상 디스크를 사용하는 방법에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- “가상 디스크 소개” [161]
- “가상 디스크 식별자 및 장치 이름” [163]
- “가상 디스크 관리” [164]
- “가상 디스크 표시” [166]
- “가상 디스크 백엔드 옵션” [168]
- “가상 디스크 백엔드” [169]
- “가상 디스크 다중 경로 구성” [176]
- “CD, DVD 및 ISO 이미지” [181]
- “가상 디스크 시간 초과” [184]
- “가상 디스크 및 SCSI” [185]
- “가상 디스크 및 `format` 명령” [185]
- “가상 디스크에서 ZFS 사용” [186]
- “Oracle VM Server for SPARC 환경에서 볼륨 관리자 사용” [190]
- “가상 디스크 문제” [193]

가상 디스크 소개

가상 디스크에는 두 구성요소(게스트 도메인에 나타나는 가상 디스크 자체 및 데이터가 저장되고 가상 I/O가 전송되는 가상 디스크 백엔드)가 포함됩니다. 가상 디스크 백엔드는 가상 디스크 서버(`vds`) 드라이버로 서비스 도메인에서 내보냅니다. `vds` 드라이버는 LDC(논리적 도메인 채널)를 사용하여 하이퍼바이저를 통해 게스트 도메인의 가상 디스크 클라이언트(`vdclient`) 드라이버와 통신합니다. 마지막으로 가상 디스크는 게스트 도메인에서 `/dev/[r]dsk/cXdYsZ` 장치로 나타납니다.

주 - 디스크 경로 이름의 일부로 `/dev/dsk` 또는 `/dev/rdisk`를 사용하여 디스크를 참조할 수 있습니다. 어느 쪽을 참조해도 결과는 같습니다.



주의 - 전체 디스크를 나타내기 위해 `d0` 장치를 사용하지 마십시오. 이 장치는 디스크에 EFI 레이블이 있고 VTOC 레이블이 없을 때만 전체 디스크를 나타냅니다. `d0` 장치를 사용하면 가상 디스크가 단일 슬라이스 디스크가 되어서 디스크 시작 부분을 기록할 때 디스크 레이블이 손상될 수 있습니다.

대신 `s2` 슬라이스를 사용하여 전체 디스크를 가상화하십시오. `s2` 슬라이스는 레이블과 무관합니다.

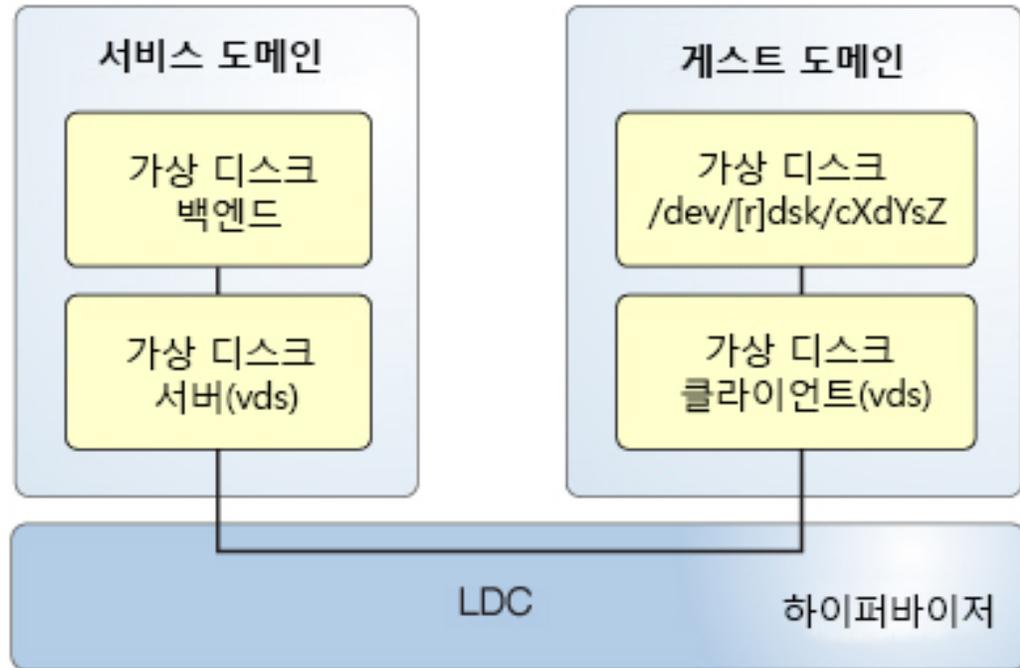
가상 디스크 백엔드는 물리적이거나 논리적일 수 있습니다. 물리적 장치에는 다음이 포함될 수 있습니다.

- 물리적 디스크 또는 디스크 논리 장치 번호(LUN)
- 물리적 디스크 슬라이스

논리적 장치는 다음 중 하나가 될 수 있습니다.

- 로컬 파일 시스템(예: ZFS 또는 UFS) 또는 NFS를 통해 제공되는 원격 파일 시스템의 파일
- Volume Manager(예: ZFS, VxVM 또는 Solaris Volume Manager)의 논리적 볼륨
- 서비스 도메인에서 액세스 가능한 디스크 의사 장치

그림 6 Oracle VM Server for SPARC의 가상 디스크



서버에서 최대 가상 디스크 수를 사용하려면 조정 가능 `segkpsize` 커널에 값이 524288 이상 있는지 확인합니다. 부족한 `segkpsize` 값으로 부트 중 또는 가상 디스크를 동적으로 추가하는 중 게스트 도메인 중단이 발생할 수 있습니다. `segkpsize`에 대한 자세한 내용은 [Oracle Solaris 11.3 Tunable Parameters Reference Manual](#)의 “`segkpsize`”를 참조하십시오.

가상 디스크 식별자 및 장치 이름

`ldm add-vdisk` 명령을 사용하여 가상 디스크를 도메인에 추가하는 경우 `id` 등록 정보를 설정하여 해당 장치 번호를 지정할 수 있습니다.

```
ldm add-vdisk [id=disk-id] disk-name volume-name@service-name domain-name
```

도메인의 각 가상 디스크는 도메인이 바인드될 때 지정되는 고유한 장치 번호를 가집니다. 가상 디스크가 명시적인 장치 번호와 함께 추가될 경우(`id` 등록 정보를 설정하여) 지정된 장치 번호가 사용됩니다. 그렇지 않은 경우 시스템에서 사용 가능한 가장 낮은 장치 번호를 자동으로

지정합니다. 이 경우 지정되는 장치 번호는 도메인에 가상 디스크가 추가된 방법에 따라 달라 집니다. 결국 가상 디스크에 지정되는 장치 번호는 도메인이 바인드될 때 `ldm list-bindings` 명령의 출력에 표시됩니다.

가상 디스크가 있는 도메인에서 Oracle Solaris OS를 실행 중인 경우 각 가상 디스크는 도메 인에서 `c0dn` 디스크 장치로 나타나며, 여기서 `n`은 가상 디스크의 장치 번호입니다.

다음 예에서 `ldg1` 도메인에는 `rootdisk` 및 `pdisk`의 두 가상 디스크가 있습니다. `rootdisk`의 장 치 번호는 `0(disk@0)`이고 도메인에서 디스크 장치 `c0d0`으로 나타납니다. `pdisk`의 장치 번호는 `1(disk@1)`이고 도메인에서 디스크 장치 `c0d1`로 나타납니다.

```
primary# ldm list-bindings ldg1
...
DISK
NAME                VOLUME                TOUT DEVICE  SERVER      MPGROUP
rootdisk            dsk_nevada@primary-vds0  disk@0  primary
pdisk               c3t40d1@primary-vds0    disk@1  primary
...
```



주의 - 장치 번호가 가상 디스크에 명시적으로 지정되지 않은 경우 도메인이 바인드 해제되고 나중에 재바인드될 때 해당 장치 번호를 변경할 수 있습니다. 이 경우 도메인에서 실행 중인 OS에서 지정한 장치 이름도 변경할 수 있으며 시스템의 기존 구성이 달라집니다. 예를 들어, 가상 디스크가 도메인의 구성에서 제거될 경우 이러한 상황이 발생할 수 있습니다.

가상 디스크 관리

이 절에서는 게스트 도메인에 가상 디스크 추가, 가상 디스크와 시간 초과 옵션 변경 및 게스트 도메인에서 가상 디스크 제거에 대해 설명합니다. 가상 디스크 옵션에 대한 설명은 “[가상 디스크 백엔드 옵션](#)” [168]을 설명하십시오. 가상 디스크 시간 초과에 대한 설명은 “[가상 디스크 시간 초과](#)” [184]를 참조하십시오.

가상 디스크 백엔드는 동일하거나 다른 가상 디스크 서버를 통해 여러 번 내보낼 수 있습니다. 그런 다음 각 가상 디스크 백엔드의 내보낸 인스턴스를 동일하거나 다른 게스트 도메인에 지정 할 수 있습니다.

가상 디스크 백엔드를 여러 번 내보낼 경우 배타적(`excl`) 옵션을 사용하여 내보내면 안됩니다. `excl` 옵션을 지정하면 백엔드 내보내기가 한 번만 허용됩니다. 백엔드는 `ro` 옵션을 사용하여 읽 기 전용 장치로 안전하게 여러 번 내보낼 수 있습니다.

가상 디스크 장치를 도메인에 지정하면 가상 디스크 서비스를 제공하는 도메인에 암시적 종속 성이 생깁니다. `ldm list-dependencies` 명령을 사용하여 해당 종속성을 보거나 가상 디스크 서비 스에 종속된 도메인을 볼 수 있습니다. “[도메인 I/O 종속성 나열](#)” [371]을 참조하십시오.

▼ 가상 디스크를 추가하는 방법

1. 서비스 도메인에서 가상 디스크 백엔드를 내보냅니다.

```
ldm add-vdsdev [-fq] [options={ro,slice,excl}] [mpgroup=mpgroup] \  
backend volume-name@service-name
```

2. 게스트 도메인에 백엔드를 지정합니다.

```
ldm add-vdisk [timeout=seconds] [id=disk-id] disk-name volume-name@service-name domain-name
```

`id` 등록 정보를 설정하여 새 가상 디스크 장치의 사용자정의 ID를 지정할 수 있습니다. 기본적으로 ID 값은 자동으로 생성되므로 OS의 기존 장치 이름과 일치시켜야 하는 경우에만 이 등록 정보를 설정하십시오. “가상 디스크 식별자 및 장치 이름” [163]을 참조하십시오.

주 - 백엔드는 게스트 도메인(*domain-name*)이 바인드될 때 실제로 서비스 도메인에서 내보내지고 게스트 도메인에 지정됩니다.

▼ 가상 디스크 백엔드를 여러 번 내보내는 방법



주의 - 가상 디스크 백엔드를 여러 번 내보낼 경우 게스트 도메인에서 실행 중이고 해당 가상 디스크를 사용 중인 응용 프로그램이 데이터 일관성 보장을 위해 동시 쓰기 액세스를 조정하고 동기화해야 합니다.

다음 예에서는 동일한 가상 디스크 서비스를 통해 동일 가상 디스크를 서로 다른 두 게스트 도메인에 추가하는 방법을 설명합니다.

1. 서비스 도메인에서 가상 디스크 백엔드를 두 번 내보냅니다.

```
ldm add-vdsdev [options={ro,slice}] backend volume1@service-name  
# ldm add-vdsdev -f [options={ro,slice}] backend volume2@service-name
```

두번째 `ldm add-vdsdev` 명령에서는 `-f` 옵션을 사용하여 백엔드의 두번째 내보내기를 수행하고 있습니다. 두 명령에 대해 동일한 백엔드 경로를 사용할 때 및 가상 디스크 서버가 동일한 서비스 도메인에 있을 때 이 옵션을 사용하십시오.

2. 각 게스트 도메인에 내보낸 백엔드를 지정합니다.

`disk-name`은 `ldom1` 및 `ldom2`에 대해 서로 다를 수 있습니다.

```
ldm add-vdisk [timeout=seconds] disk-name volume1@service-name ldom1  
# ldm add-vdisk [timeout=seconds] disk-name volume2@service-name ldom2
```

▼ 가상 디스크 옵션을 변경하는 방법

가상 디스크 옵션에 대한 자세한 내용은 “[가상 디스크 백엔드 옵션](#)” [168]을 참조하십시오.

- 서비스 도메인에서 백엔드를 내보낸 후 가상 디스크 옵션을 변경할 수 있습니다.

```
primary# ldm set-vdsdev options=[{ro,slice,excl}] volume-name@service-name
```

▼ 시간 초과 옵션을 변경하는 방법

가상 디스크 옵션에 대한 자세한 내용은 “[가상 디스크 백엔드 옵션](#)” [168]을 참조하십시오.

- 게스트 도메인에 가상 디스크를 지정한 후 가상 디스크 시간 초과를 변경할 수 있습니다.

```
primary# ldm set-vdisk timeout=seconds disk-name domain-name
```

▼ 가상 디스크를 제거하는 방법

1. 게스트 도메인에서 가상 디스크를 제거합니다.

```
primary# ldm rm-vdisk disk-name domain-name
```

2. 서비스 도메인에서 해당 백엔드 내보내기를 중지합니다.

```
primary# ldm rm-vdsdev volume-name@service-name
```

가상 디스크 표시

백엔드를 가상 디스크로 내보낼 경우 게스트 도메인에서 전체 디스크나 단일 슬라이스 디스크로 나타날 수 있습니다. 나타나는 방식은 백엔드의 유형 및 내보내는 데 사용된 옵션에 따라 달라집니다.

주 - NVMe(Non-Volatile Memory Express) 저장소는 SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버부터 제공됩니다. 이 저장소는 디스크 드라이브 또는 Flash Accelerator F160 PCIe 카드일 수 있습니다. 이 디스크 유형은 가상 디스크 백엔드를 구성하는 데 사용할 수 있습니다.

Oracle Solaris 11.3 SRU 2.4 OS부터 NVMe 저장소 디스크 유형을 전체 디스크 또는 단일 슬라이스 디스크로 사용할 수 있습니다.

Oracle Solaris 11.3 SRU 2.4 OS 이전에는 NVMe 저장소 디스크 유형을 단일 슬라이스 디스크로만 사용할 수 있습니다.



주의 - 단일 슬라이스 디스크는 장치 ID가 없습니다. 장치 ID가 필요하면 전체 물리적 디스크 백엔드를 사용하십시오.

전체 디스크

백엔드를 전체 디스크로 도메인에 내보낼 경우 해당 도메인에서 8개의 슬라이스($s_0 - s_7$)를 가진 일반 디스크로 나타납니다. 이 유형의 디스크는 `format(1M)` 명령으로 볼 수 있습니다. 디스크의 분할 영역 테이블은 `fmthard` 또는 `format` 명령을 사용하여 변경할 수 있습니다.

전체 디스크는 OS 설치 소프트웨어에도 표시되며 OS를 설치할 디스크로 선택할 수 있습니다.

단일 슬라이스 디스크로만 내보낼 수 있는 물리적 디스크 슬라이스를 제외하고 모든 백엔드는 전체 디스크로 내보낼 수 있습니다.

단일 슬라이스 디스크

백엔드를 단일 슬라이스 디스크로 도메인에 내보낼 경우 해당 도메인에서 8개의 슬라이스($s_0 - s_7$)를 가진 일반 디스크로 나타납니다. 하지만 첫번째 슬라이스(s_0)만 사용할 수 있습니다. 이 유형의 디스크는 `format(1M)` 명령으로 볼 수 있지만, 디스크의 분할 영역 테이블은 변경할 수 없습니다.

단일 슬라이스 디스크는 OS 설치 소프트웨어에도 표시되며 OS를 설치할 디스크로 선택할 수 있습니다. 이 때 UNIX 파일 시스템(UFS)을 사용하여 OS를 설치할 경우 루트 분할 영역($/$)만 정의해야 하며 이 분할 영역에서 모든 디스크 공간을 사용해야 합니다.

전체 디스크로만 내보낼 수 있는 물리적 디스크를 제외하고 모든 백엔드는 단일 슬라이스 디스크로 내보낼 수 있습니다.

주 - Oracle Solaris 10 10/08 OS 릴리스 이전에는 단일 슬라이스 디스크가 단일 분할 영역 (`sd`)을 가진 디스크로 나타났습니다. 이 유형의 디스크는 `format` 명령을 사용하여 볼 수 없었습니다. 또한 디스크는 OS 설치 소프트웨어에 표시되지 않았으며 OS를 설치할 디스크 장치로 선택할 수 없었습니다.

가상 디스크 백엔드 옵션

가상 디스크 백엔드를 내보낼 때 서로 다른 옵션을 지정할 수 있습니다. 해당 옵션은 `ldm add-vdsdev` 명령의 `options=` 인수에 쉼표로 구분된 목록으로 지정됩니다. 유효한 옵션은 `ro`, `slice` 및 `excl`입니다.

읽기 전용(`ro`) 옵션

읽기 전용(`ro`) 옵션은 백엔드를 읽기 전용 장치로 내보내도록 지정합니다. 이 경우 게스트 도메인에 지정되는 가상 디스크는 읽기 작업용으로만 액세스할 수 있으며, 가상 디스크에 대한 모든 쓰기 작업은 실패합니다.

배타적(`excl`) 옵션

배타적(`excl`) 옵션은 다른 도메인에 가상 디스크로 내보낼 때 서비스 도메인의 백엔드를 가상 디스크 서버에서 배타적으로 열도록 지정합니다. 백엔드가 배타적으로 열리면 서비스 도메인의 다른 응용 프로그램에서 액세스할 수 없게 됩니다. 이 제한 사항으로 인해 서비스 도메인에서 실행되는 응용 프로그램이 게스트 도메인에서도 사용되고 있는 백엔드를 실수로 사용하지 않도록 방지됩니다.

주 - 일부 드라이버는 `excl` 옵션을 준수하지 않으며 일부 가상 디스크 백엔드가 배타적으로 열리는 것을 허용하지 않습니다. `excl` 옵션은 물리적 디스크 및 슬라이스에서 작동하는 것으로 알려져 있지만, 이 옵션은 파일에서 작동하지 않습니다. 디스크 볼륨과 같은 의사 장치에서 작동할 수도 있습니다. 백엔드의 드라이버가 배타적 열기를 준수하지 않을 경우 백엔드 `excl` 옵션은 무시되고 백엔드가 배타적으로 열리지 않습니다.

`excl` 옵션은 서비스 도메인에서 실행 중인 응용 프로그램이 게스트 도메인으로 내보낸 백엔드에 액세스하지 못하도록 하므로 다음 상황에서는 `excl` 옵션을 설정하지 마십시오.

- 게스트 도메인이 실행 중일 때 `format`과 같은 명령을 사용하여 물리적 디스크를 관리하려는 경우 `excl` 옵션과 함께 이러한 디스크를 내보내지 마십시오.

- Solaris Volume Manager 볼륨(RAID 또는 미러링된 볼륨 등)을 내보낼 때 `excl` 옵션을 설정하지 마십시오. 그렇지 않으면 RAID 또는 미러링된 볼륨의 구성요소가 실패할 경우 Solaris Volume Manager에서 일부 복구 작업을 시작하지 못할 수 있습니다. 자세한 내용은 “Solaris Volume Manager에서 가상 디스크 사용” [191]을 참조하십시오.
- VxVM(Veritas Volume Manager)이 서비스 도메인에 설치되고 VxDMP(Veritas Dynamic Multipathing)가 물리적 디스크에 대해 사용으로 설정된 경우 물리적 디스크는 `excl` 옵션 없이(기본값 아님) 내보내야 합니다. 그렇지 않으면 가상 디스크 서버(`vds`)에서 물리적 디스크 장치를 열 수 없으므로 내보내기가 실패합니다. 자세한 내용은 “VxVM이 설치된 경우 가상 디스크 사용” [191]을 참조하십시오.
- 동일 가상 디스크 서비스에서 동일한 가상 디스크 백엔드를 여러 번 내보내는 경우 자세한 내용은 가상 디스크 백엔드를 여러 번 내보내는 방법 [165]을 참조하십시오.

기본적으로 백엔드는 비배타적으로 열립니다. 이에 따라 백엔드는 서비스 도메인에서 실행 중인 응용 프로그램이 사용할 수 있고 다른 도메인에도 내보낼 수 있습니다.

슬라이스(slice) 옵션

백엔드는 해당 유형에 따라 일반적으로 전체 디스크 또는 단일 슬라이스 디스크로 내보내집니다. `slice` 옵션이 지정되면 백엔드는 강제로 단일 슬라이스 디스크로 내보내집니다.

이 옵션은 백엔드의 원시 콘텐츠를 내보내고자 할 때 유용합니다. 예를 들어, 이미 데이터를 저장한 ZFS 또는 Solaris Volume Manager 볼륨이 있고 게스트 도메인에서 이 데이터에 액세스하도록 하려는 경우 `slice` 옵션을 사용하여 ZFS 또는 Solaris Volume Manager 볼륨을 내보내야 합니다.

이 옵션에 대한 자세한 내용은 “가상 디스크 백엔드” [169]를 참조하십시오.

가상 디스크 백엔드

가상 디스크 백엔드는 가상 디스크의 데이터가 저장되는 위치입니다. 백엔드는 디스크, 디스크 슬라이스, 파일 또는 볼륨(예: ZFS, Solaris Volume Manager 또는 VxVM)이 될 수 있습니다. 백엔드를 서비스 도메인에서 내보낼 때 `slice` 옵션이 설정되었는지 여부에 따라 백엔드는 게스트 도메인에서 전체 디스크 또는 단일 슬라이스 디스크로 나타납니다. 기본적으로 가상 디스크 백엔드는 읽기 가능-쓰기 가능 전체 디스크로 비배타적으로 내보냅니다.

물리적 디스크 또는 디스크 LUN

물리적 디스크 또는 디스크 LUN은 항상 전체 디스크로 내보냅니다. 이 경우 가상 디스크 드라이버(`vds` 및 `vdv`)가 가상 디스크에서 I/O를 전달하고 물리적 디스크 또는 디스크 LUN에 통로로 작동합니다.

`slice` 옵션을 설정하지 않고 해당 디스크의 슬라이스 2(`s2`)에 해당하는 장치를 내보내서 물리적 디스크 또는 디스크 LUN을 서비스 도메인에서 내보냅니다. `slice` 옵션을 사용하여 디스크의 슬라이스 2를 내보낼 경우 전체 디스크가 아닌 이 슬라이스만 내보내집니다.

▼ 물리적 디스크를 가상 디스크로 내보내는 방법



주의 - 가상 디스크를 구성할 때는 각 가상 디스크가 별개의 물리적(백엔드) 리소스(예: 물리적 디스크, 디스크 슬라이스, 파일, 볼륨)를 참조하도록 해야 합니다.

FibreChannel 및 SAS와 같은 일부 디스크는 "이중 포트"를 사용하므로 동일한 디스크를 두 개의 다른 경로에서 참조할 수 있습니다. 서로 다른 도메인에 지정한 경로가 동일한 물리적 디스크를 참조하지 않도록 하십시오.

1. 물리적 디스크를 가상 디스크로 내보냅니다.

예를 들어, 물리적 디스크 `c1t48d0`을 가상 디스크로 내보내려면 해당 디스크의 슬라이스 2(`c1t48d0s2`)를 내보내야 합니다.

```
primary# ldm add-vdsdev /dev/dsk/c1t48d0s2 c1t48d0@primary-vds0
```

2. 게스트 도메인에 디스크를 지정합니다.

예를 들어, 디스크(`pdisk`)를 게스트 도메인 `ldg1`에 지정합니다.

```
primary# ldm add-vdisk pdisk c1t48d0@primary-vds0 ldg1
```

3. 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크에 액세스 가능하고 전체 디스크인지 확인합니다.

전체 디스크는 8개의 슬라이스가 있는 일반 디스크입니다.

예를 들어, 확인할 디스크는 `c0d1`입니다.

```
ldg1# ls -l /dev/dsk/c0d1s*  
/dev/dsk/c0d1s0  
/dev/dsk/c0d1s1  
/dev/dsk/c0d1s2  
/dev/dsk/c0d1s3  
/dev/dsk/c0d1s4  
/dev/dsk/c0d1s5  
/dev/dsk/c0d1s6  
/dev/dsk/c0d1s7
```

물리적 디스크 슬라이스

물리적 디스크 슬라이스는 항상 단일 슬라이스 디스크로 내보냅니다. 이 경우 가상 디스크 드라이버(vds 및 vdc)가 가상 디스크에서 I/O를 전달하고 물리적 디스크 슬라이스에 통로로 작동합니다.

물리적 디스크 슬라이스는 해당하는 슬라이스 장치를 내보내서 서비스 도메인에서 내보냅니다. 장치가 슬라이스 2와 다른 경우 `slice` 옵션 지정 여부에 관계없이 자동으로 단일 슬라이스 디스크로 내보냅니다. 장치가 해당 디스크의 슬라이스 2인 경우 `slice` 옵션을 설정하여 슬라이스 2만 단일 슬라이스 디스크로 내보내야 합니다. 그렇지 않으면 모든 디스크가 전체 디스크로 내보내집니다.

주 - NVMe(Non-Volatile Memory Express) 저장소는 SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버부터 제공됩니다. 이 저장소는 디스크 드라이브 또는 Flash Accelerator F160 PCIe 카드일 수 있습니다.

Oracle Solaris 11.3 SRU 2.4 OS부터 NVMe 저장소 디스크 유형을 전체 디스크 또는 단일 슬라이스 디스크로 사용할 수 있습니다.

Oracle Solaris 11.3 SRU 2.4 OS 이전에는 NVMe 저장소 디스크 유형을 단일 슬라이스 디스크로만 사용할 수 있습니다.

▼ 물리적 디스크 슬라이스를 가상 디스크로 내보내는 방법

1. 물리적 디스크의 슬라이스를 가상 디스크로 내보냅니다.

예를 들어, 물리적 디스크(c1t57d0)의 슬라이스 0을 가상 디스크로 내보내려면 해당 슬라이스에 해당하는 장치(c1t57d0s0)를 다음과 같이 내보내야 합니다.

```
primary# ldm add-vdsdev /dev/dsk/c1t57d0s0 c1t57d0s0@primary-vds0
```

슬라이스는 항상 단일 슬라이스 디스크로 내보내지므로 `slice` 옵션을 지정할 필요가 없습니다.

2. 게스트 도메인에 디스크를 지정합니다.

예를 들어, 디스크(pslice)를 게스트 도메인 ldg1에 지정합니다.

```
primary# ldm add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

3. 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크(예: c0d13)를 나열하고 해당 디스크에 액세스할 수 있는지 확인합니다.

```
ldg1# ls -l /dev/dsk/c0d13s*
/dev/dsk/c0d13s0
/dev/dsk/c0d13s1
/dev/dsk/c0d13s2
```

```
/dev/dsk/c0d13s3  
/dev/dsk/c0d13s4  
/dev/dsk/c0d13s5  
/dev/dsk/c0d13s6  
/dev/dsk/c0d13s7
```

8개의 장치가 있지만 디스크는 단일 슬라이스 디스크이므로 첫번째 슬라이스(s0)만 사용할 수 있습니다.

▼ 슬라이스 2를 내보내는 방법

- 슬라이스 2(예: 디스크 c1t57d0s2)를 내보내려면 `slice` 옵션을 지정해야 합니다. 그렇지 않으면 전체 디스크가 내보내집니다.

```
primary# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

파일 및 볼륨 내보내기

파일 또는 볼륨(예: ZFS 또는 Solaris Volume Manager)은 `slice` 옵션 설정 여부에 따라 전체 디스크 또는 단일 슬라이스 디스크로 내보내집니다.

전체 디스크로 내보낸 파일 또는 볼륨

`slice` 옵션을 설정하지 않을 경우 파일 또는 볼륨은 전체 디스크로 내보내집니다. 이 경우 가상 디스크 드라이버(`vds` 및 `vdc`)가 가상 디스크에서 I/O를 전달하고 가상 디스크의 분할 영역을 관리합니다. 파일 또는 볼륨은 결국 가상 디스크의 모든 슬라이스 데이터와 분할 영역 및 디스크 구조를 관리하는 데 사용되는 메타데이터를 포함하는 디스크 이미지가 됩니다.

빈 파일 또는 볼륨을 전체 디스크로 내보낼 경우 게스트 도메인에서 포맷되지 않은 디스크(즉, 분할 영역이 없는 디스크)로 나타납니다. 이 경우 게스트 도메인에서 `format` 명령을 실행하여 사용 가능한 분할 영역을 정의하고 유효한 디스크 레이블을 작성해야 합니다. 디스크가 포맷되지 않은 동안에는 가상 디스크에 대한 모든 I/O가 실패합니다.

주 - 게스트 도메인에서 `format` 명령을 실행하여 분할 영역을 만들어야 합니다.

▼ 파일을 전체 디스크로 내보내는 방법

1. 서비스 도메인에서 가상 디스크로 사용할 파일(예: `fdisk0`)을 만듭니다.

```
service# mkfile 100m /ldoms/domain/test/fdisk0
```

파일의 크기로 가상 디스크의 크기가 정의됩니다. 이 예에서는 100MB의 가상 디스크를 얻기 위해 100MB의 빈 파일을 만듭니다.

2. 컨트롤 도메인에서 파일을 가상 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /ldoms/domain/test/fdisk0 fdisk0@primary-vds0
```

이 예에서는 `slice` 옵션이 설정되지 않았으므로 파일이 전체 디스크로 내보내집니다.

3. 컨트롤 도메인에서 디스크를 게스트 도메인에 지정합니다.

예를 들어, 디스크(`fdisk`)를 게스트 도메인 `ldg1`에 지정합니다.

```
primary# ldm add-vdisk fdisk fdisk0@primary-vds0 ldg1
```

4. 게스트 도메인이 시작되고 **Oracle Solaris OS**를 실행 중일 때 디스크에 액세스 가능하고 전체 디스크인지 확인합니다.

전체 디스크는 8개의 슬라이스가 있는 일반 디스크입니다.

다음 예에서는 `c0d5` 디스크를 나열하며, 액세스 가능하고 전체 디스크인지 확인하는 방법을 보여줍니다.

```
ldg1# ls -l /dev/dsk/c0d5s*
/dev/dsk/c0d5s0
/dev/dsk/c0d5s1
/dev/dsk/c0d5s2
/dev/dsk/c0d5s3
/dev/dsk/c0d5s4
/dev/dsk/c0d5s5
/dev/dsk/c0d5s6
/dev/dsk/c0d5s7
```

▼ ZFS 볼륨을 전체 디스크로 내보내는 방법

1. 전체 디스크로 사용할 ZFS 볼륨을 만듭니다.

다음 예에서는 전체 디스크로 사용할 ZFS 볼륨 `zdisk0`을 만드는 방법을 보여줍니다.

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

볼륨의 크기로 가상 디스크의 크기가 정의됩니다. 이 예에서는 100MB의 가상 디스크를 얻기 위해 100MB의 볼륨을 만듭니다.

2. 컨트롤 도메인에서 해당 장치를 ZFS 볼륨으로 내보냅니다.

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

이 예에서는 `slice` 옵션이 설정되지 않았으므로 파일이 전체 디스크로 내보내집니다.

3. 컨트롤 도메인에서 볼륨을 게스트 도메인에 지정합니다.

다음 예에서는 `zdisk0` 볼륨을 게스트 도메인 `ldg1`에 지정하는 방법을 보여줍니다.

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4. 게스트 도메인이 시작되고 **Oracle Solaris OS**를 실행 중일 때 디스크에 액세스 가능하고 전체 디스크인지 확인합니다.

전체 디스크는 8개의 슬라이스가 있는 일반 디스크입니다.

다음 예에서는 `c0d9` 디스크를 나열하며, 액세스 가능하고 전체 디스크인지 확인하는 방법을 보여줍니다.

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7
```

단일 슬라이스 디스크로 내보낸 파일 또는 볼륨

`slice` 옵션이 설정되면 파일 또는 볼륨이 단일 슬라이스 디스크로 내보내집니다. 이 경우 가상 디스크에는 직접 파일 또는 볼륨 백엔드로 매핑되는 하나의 분할 영역(`s0`)만 있습니다. 파일 또는 볼륨에는 분할 영역 정보나 디스크 구조와 같은 추가 데이터 없이 가상 디스크에 기록된 데이터만 포함됩니다.

파일 또는 볼륨을 단일 슬라이스 디스크로 내보낼 경우 시스템에서는 파일 또는 볼륨이 디스크 슬라이스로 나타나도록 가짜 디스크 분할을 시뮬레이션합니다. 디스크 분할이 시뮬레이션되므로 해당 디스크에 대한 분할 영역을 만들지 않습니다.

▼ ZFS 볼륨을 단일 슬라이스 디스크로 내보내는 방법

1. 단일 슬라이스 디스크로 사용할 ZFS 볼륨을 만듭니다.

다음 예에서는 단일 슬라이스 디스크로 사용할 ZFS 볼륨 `zdisk0`을 만드는 방법을 보여줍니다.

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

볼륨의 크기로 가상 디스크의 크기가 정의됩니다. 이 예에서는 100MB의 가상 디스크를 얻기 위해 100MB의 볼륨을 만듭니다.

2. 컨트롤 도메인에서 해당 장치를 ZFS 볼륨으로 내보내고 `slice` 옵션을 설정하여 볼륨이 단일 슬라이스 디스크로 내보내지도록 합니다.

```
primary# ldm add-vdsdev options=slice /dev/zvol/dsk/ldoms/domain/test/zdisk0 \
zdisk0@primary-vds0
```

3. 컨트롤 도메인에서 볼륨을 게스트 도메인에 지정합니다.

다음 예에서는 `zdisk0` 볼륨을 게스트 도메인 `ldg1`에 지정하는 방법을 보여줍니다.

```
primary# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```

4. 게스트 도메인이 시작되고 Oracle Solaris OS를 실행 중일 때 디스크(예: `c0d9`)를 나열한 후 해당 디스크에 액세스 가능하고 단일 슬라이스 디스크(`s0`)인지 확인합니다.

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
/dev/dsk/c0d9s1
/dev/dsk/c0d9s2
/dev/dsk/c0d9s3
/dev/dsk/c0d9s4
/dev/dsk/c0d9s5
/dev/dsk/c0d9s6
/dev/dsk/c0d9s7
```

볼륨 내보내기 및 역호환성

가상 디스크로 볼륨을 내보내는 구성이 있을 경우 단일 슬라이스 디스크 대신 전체 디스크로 볼륨이 내보내집니다. 이전 동작을 유지하고 볼륨을 단일 슬라이스 디스크로 내보내려면 다음 중 하나를 수행해야 합니다.

- Oracle VM Server for SPARC 3.4 소프트웨어에서 `ldm set-vdsdev` 명령을 사용하고 단일 슬라이스 디스크로 내보낼 모든 볼륨에 대해 `slice` 옵션을 설정합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- 서비스 도메인에서 `/etc/system` 파일에 다음 행을 추가합니다.

```
set vds:vd_volume_force_slice = 1
```

`/etc/system` 등록 정보 값을 정확히 만들거나 업데이트하는 방법은 ["/etc/system 파일에서 등록 정보 값 업데이트" \[352\]](#)를 참조하십시오.

주 - 이 조정 가능 변수를 설정하면 모든 볼륨이 단일 슬라이스 디스크로 내보내지고 전체 디스크로는 내보낼 수 없게 됩니다.

백엔드를 내보내는 여러 유형 요약

백엔드	슬라이스 옵션 없음	슬라이스 옵션 설정
디스크(디스크 슬라이스 2)	전체 디스크 [†]	단일 슬라이스 디스크 [‡]
디스크 슬라이스(슬라이스 2 아님)	단일 슬라이스 디스크 [*]	단일 슬라이스 디스크

백엔드	슬라이스 옵션 없음	슬라이스 옵션 설정
파일	전체 디스크	단일 슬라이스 디스크
볼륨(ZFS, Solaris Volume Manager, VxVM 등)	전체 디스크	단일 슬라이스 디스크

[†]전체 디스크로 내보냅니다.

[‡]슬라이스 2만 내보냅니다.

^{*}슬라이스는 항상 단일 슬라이스 디스크로 내보냅니다.

파일 및 디스크 슬라이스를 가상 디스크로 내보내기 위한 지침

이 절에는 파일 및 디스크 슬라이스를 가상 디스크로 내보내기 위한 지침이 포함되어 있습니다.

루프백 파일(lofi) 드라이버 사용

루프백 파일(lofi) 드라이버를 사용하여 파일을 가상 디스크로 내보내면 여분의 드라이버 계층이 추가되고 가상 디스크의 성능이 영향을 받습니다. 대신 파일을 전체 디스크나 단일 슬라이스 디스크로 직접 내보낼 수 있습니다. [“파일 및 볼륨 내보내기” \[172\]](#)를 참조하십시오.

직접 또는 간접적으로 디스크 슬라이스 내보내기

슬라이스를 직접 또는 간접적으로(예: Solaris Volume Manager 볼륨을 통해) 가상 디스크로 내보내려면 prtvtoc 명령을 사용하여 슬라이스가 물리적 디스크의 첫번째 블록(블록 0)에서 시작되지 않는지 확인하십시오.

물리적 디스크의 첫번째 블록에서 시작되는 디스크 슬라이스를 직접 또는 간접적으로 내보낼 경우 물리적 디스크의 분할 영역 테이블을 덮어쓰고 해당 디스크의 모든 분할 영역에 액세스하지 못하게 될 수 있습니다.

가상 디스크 다중 경로 구성

가상 디스크 다중 경로를 통해 여러 경로에서 백엔드 저장소에 액세스하도록 게스트 도메인의 가상 디스크를 구성할 수 있습니다. 경로는 동일 백엔드 저장소(디스크 LUN 등)에 대한 액세스를 제공하는 여러 서비스 도메인을 통해 연결됩니다. 이 기능을 사용하면 서비스 도메인 중 하나가 작동 중지되더라도 게스트 도메인의 가상 디스크에 액세스할 수 있습니다. 예를 들어, 네트워크 파일 시스템(NFS) 서버의 파일에 액세스하도록 가상 디스크 다중 경로 구성을 설정할 수 있습니다. 또는 이 구성을 사용하여 둘 이상의 서비스 도메인에 연결된 공유 저장소에서

LUN에 액세스할 수 있습니다. 따라서 게스트 도메인에서 가상 디스크에 액세스할 경우 가상 디스크 드라이버가 서비스 도메인 중 하나를 통과하여 백엔드 저장소에 액세스하게 됩니다. 가상 디스크 드라이버가 서비스 도메인에 연결할 수 없는 경우 가상 디스크는 다른 서비스 도메인을 통해 백엔드 저장소에 접근하려고 시도합니다.

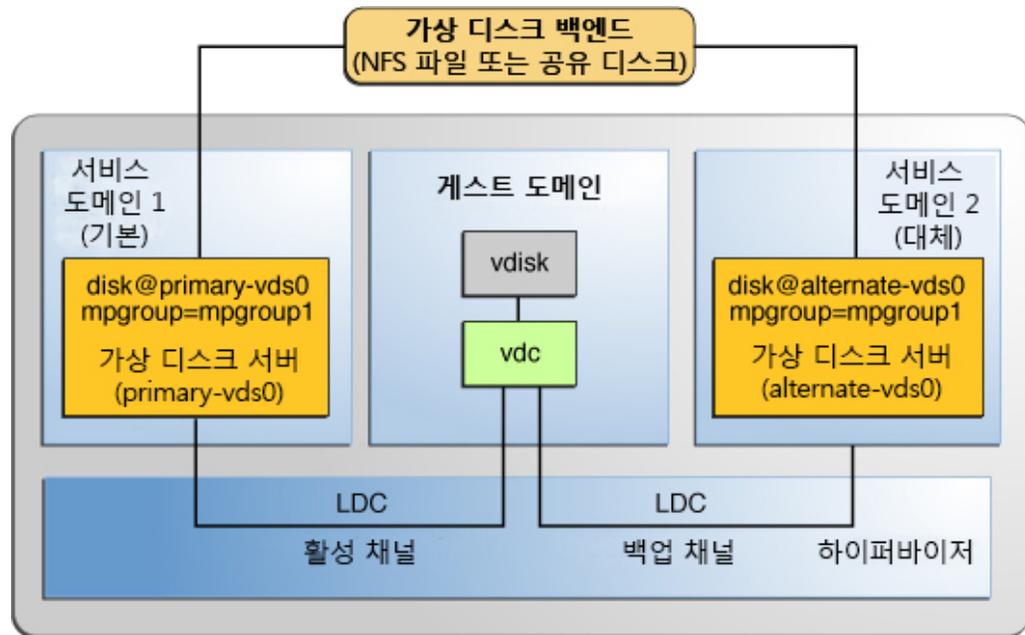
주 - mpgroup과 SCSI 예약을 함께 사용할 수 없습니다.

가상 디스크 다중 경로 기능에서는 서비스 도메인이 백엔드 저장소에 액세스할 수 없을 때 감지할 수 있습니다. 이러한 경우 가상 디스크 드라이버는 다른 경로로 백엔드 저장소에 액세스를 시도합니다.

가상 디스크 다중 경로를 사용으로 설정하려면 가상 디스크 백엔드를 각 서비스 도메인에서 내보내고 가상 디스크를 동일 다중 경로 그룹(mpgroup)에 추가해야 합니다. mpgroup은 이름으로 식별되고 가상 디스크 백엔드를 내보낼 때 구성됩니다.

다음 그림은 가상 디스크 다중 경로를 구성하는 방법 [178] 절차에서 예로 사용된 가상 디스크 다중 경로 구성을 보여줍니다. 이 예에서 mpgroup1 이름의 다중 경로 그룹이 가상 디스크를 만드는 데 사용되며, 백엔드는 primary 및 alternate의 두 서비스 도메인에서 액세스할 수 있습니다.

그림 7 가상 디스크 다중 경로 구성



가상 디스크 다중 경로 및 NFS

NFS 파일 시스템에서 `mpgroup` 페일오버를 설정하는 경우 `soft` NFS 마운트 옵션으로 NFS 파일 시스템을 마운트하는지 확인합니다.

`soft` 옵션을 사용하면 I/O 오류가 VDS 또는 VDC에 보고되고 VDC는 추가 메시지를 전송하여 이것이 I/O 오류인지 또는 전체 백엔드를 사용할 수 없는지 여부를 확인합니다. 페일오버 시간은 NFS 시간 초과 및 가능한 재전송에 비례합니다.

기본값 60초(`timeo=600`)에서 `timeo` NFS 마운트 옵션을 변경하지 마십시오. `timeo=40`과 같은 짧은 시간 초과 값은 허위 I/O 오류가 발생할 수 있음을 의미합니다. 예를 들어 NFS 서버 또는 네트워크를 5초 동안 사용할 수 없게 된 후 작업으로 복귀하면 중단이 페일오버가 발생할 만큼 길지 않았기 때문에 I/O 오류가 보고될 수 있습니다. 긴 시간 초과(예: 60초)는 몇 초 동안의 중단을 마스킹합니다.

가상 디스크 다중 경로 및 가상 디스크 시간 초과

가상 디스크 다중 경로를 사용하면 현재 활성 경로로 백엔드에 액세스할 수 없는 경우 백엔드에 액세스하는 데 사용되는 경로가 자동으로 변경됩니다. 이 경로 변경은 가상 디스크 `timeout` 등록 정보의 값과 무관하게 발생합니다.

가상 디스크 `timeout` 등록 정보는 I/O를 처리하는 데 사용할 수 있는 서비스 도메인이 없을 때 I/O가 실패하는 시간을 지정합니다. 이 시간 초과는 가상 디스크 다중 경로를 사용하는 가상 디스크를 포함한 모든 가상 디스크에 적용됩니다.

이에 따라 가상 디스크 다중 경로가 구성된 경우 가상 디스크 시간 초과를 설정하면, 특히 시간 초과 값을 작게 설정하면 다중 경로가 제대로 작동하지 않을 수 있습니다. 따라서 다중 경로 그룹의 일부인 가상 디스크에 대해서는 가상 디스크 시간 초과를 설정하지 않는 것이 좋습니다.

자세한 내용은 “[가상 디스크 시간 초과](#)” [184]를 참조하십시오.

▼ 가상 디스크 다중 경로를 구성하는 방법

그림 7을 참조하십시오.

1. `primary` 서비스 도메인에서 가상 디스크 백엔드를 내보냅니다.

```
primary# ldm add-vdsdev mpgroup=mpgroup1 backend-path1 volume@primary-vds0
```

`backend-path1`은 `primary` 도메인에서 가상 디스크 백엔드로 연결되는 경로입니다.

2. `alternate` 서비스 도메인에서 동일한 가상 디스크 백엔드를 내보냅니다.

```
primary# ldm add-vdsdev mpgroup=mpgroup1 backend-path2 volume@alternate-vds0
```

*backend-path2*는 *alternate* 도메인에서 가상 디스크 백엔드로 연결되는 경로입니다.

주 - *backend-path1* 및 *backend-path2*는 서로 다른 두 도메인(*primary* 및 *alternate*)에서 동일한 가상 디스크 백엔드에 대한 경로입니다. 이러한 경로는 *primary* 및 *alternate* 도메인의 구성에 따라 같거나 서로 다를 수 있습니다. *volume* 이름은 사용자 선택입니다. 두 명령에 대해 같거나 서로 다를 수 있습니다.

3. 가상 디스크를 게스트 도메인으로 내보냅니다.

```
primary# ldm add-vdisk disk-name volume@primary-vds0 domain-name
```

주 - 가상 디스크 백엔드는 여러 서비스 도메인을 통해 여러 번 내보낼 수 있지만, 하나의 가상 디스크만 게스트 도메인에 지정하고 서비스 도메인을 통해 가상 디스크 백엔드와 연결해야 합니다.

예 34 Mpgroup을 사용하여 기본 도메인과 대체 도메인 모두의 가상 디스크 서비스에 LUN 추가

다음은 LUN을 만들 때 동일한 *mpgroup*을 사용하여 기본 도메인과 대체 도메인 모두의 가상 디스크 서비스에 추가하는 방법을 보여줍니다.

LUN에 액세스할 때 첫번째로 사용할 도메인을 결정하려면 도메인에 디스크를 추가할 때 연관된 경로를 지정합니다.

- 가상 디스크 장치를 만듭니다.

```
primary# ldm add-vdsdev mpgroup=ha lun1@primary-vds0
primary# ldm add-vdsdev mpgroup=ha lun1@alternate-vds0
```

- *primary-vds0*의 LUN을 첫번째로 사용하려면 다음 명령을 수행하십시오.

```
primary# ldm add-vdisk disk1 lun1@primary-vds0 gd0
```

- *alternate-vds0*의 LUN을 첫번째로 사용하려면 다음 명령을 수행하십시오.

```
primary# ldm add-vdisk disk1 lun1@alternate-vds0 gd0
```

가상 디스크 다중 경로 결과

다중 경로로 가상 디스크를 구성하고 게스트 도메인을 시작하면 가상 디스크에서 연관된 서비스 도메인 중 하나를 통해 백엔드에 액세스합니다. 이 서비스 도메인을 사용할 수 없는 경우 가상 디스크는 동일 다중 경로 그룹의 일부인 다른 서비스 도메인을 통해 백엔드에 액세스를 시도합니다.



주의 - 다중 경로 그룹(*mpgroup*)을 정의할 때 동일 *mpgroup*의 일부인 가상 디스크 백엔드가 실제로 동일한 가상 디스크 백엔드인지 확인하십시오. 서로 다른 백엔드를 동일 *mpgroup*에 추가할 경우 예상치 않은 동작이 발생할 수 있으며, 백엔드에 저장된 데이터가 손실되거나 손상될 수 있습니다.

동적 경로 선택

최소 Oracle Solaris 11.2 SRU 1 OS를 실행하는 게스트 도메인에서 가상 디스크에 사용할 경로를 동적으로 선택할 수 있습니다.

`ldm set-vdisk` 명령에서 `volume` 등록 정보를 `volume-name@service-name` 형식의 값으로 설정하여 `mpgroup` 디스크의 첫번째 경로를 변경할 때 동적 경로 선택이 발생합니다. 동적 경로 선택을 지원하는 활성 도메인은 선택된 경로로만 전환할 수 있습니다. 업데이트된 드라이버가 실행 중이 아니면 Oracle Solaris OS가 디스크 인스턴스를 다시 로드할 때나 다음 도메인 재부트 시 이 경로가 선택됩니다.

동적 경로 선택 기능을 통해 디스크가 사용 중인 동안 다음 단계를 동적으로 수행할 수 있습니다.

- 디스크를 연결할 때 게스트 도메인에서 첫번째 시도할 디스크 경로를 지정합니다.
- 현재 활성 경로를 이미 연결된 다중 경로 디스크에 지정된 경로로 변경합니다.

이제 `ldm add-vdisk` 명령을 `mpgroup` 디스크와 함께 사용하여 `volume-name@service-name`으로 지정된 경로를 디스크 액세스를 위해 선택된 경로로 지정합니다.

연관된 `mpgroup`을 만들 때의 순위에 관계없이 선택된 디스크 경로가 게스트 도메인에 제공된 경로 세트에 첫번째로 나열됩니다.

바운드, 비활성, 활성 도메인에서 `ldm set-vdisk` 명령을 사용할 수 있습니다. 활성 도메인에서 사용할 때 이 명령으로 사용자는 `mpgroup` 디스크의 선택된 경로만 고를 수 있습니다.

`ldm list-bindings` 명령은 다음 정보를 보여줍니다.

- 각 `mpgroup` 경로의 `STATE` 열은 다음 값 중 하나를 나타냅니다.
 - `active` – 현재 활성 `mpgroup` 경로
 - `standby` – 경로가 현재 사용되지 않음
 - `unknown` – 도메인이 동적 경로 선택을 지원하지 않거나, 장치가 연결되지 않았거나, 오류 때문에 경로 상태가 검색되지 않음
- 디스크 경로는 활성 경로 선택에 사용되는 순서로 나열됩니다.
- 디스크와 연관된 볼륨은 `mpgroup`에 선택된 경로이며 첫번째로 나열됩니다.

다음 예제는 선택된 경로가 `vol-1dg2@opath-1dg2`이고 현재 사용된 활성 경로가 `1dg1` 도메인을 지나고 있음을 보여줍니다. 선택된 경로를 사용할 수 없어서 대신 두번째 가능한 경로를 사용했을 때 이 상황이 발생할 수 있습니다. 선택된 경로가 온라인으로 돌아와도 비선택 경로가 계속 사용됩니다. 첫번째 경로를 다시 활성으로 만들려면 `ldm set-vdisk` 명령을 다시 실행하여 `volume` 등록 정보를 원하는 경로 이름으로 설정합니다.

DISK

NAME	VOLUME	TOUT	ID	DEVICE	SERVER	MPGROUP
disk	disk-1dg4@primary-vds0	0		disk@0	primary	
tdiskgroup	vol-1dg2@opath-1dg2	1		disk@1	1dg2	testdiskgroup

PORT	MPGROUP	VOLUME	MPGROUP	SERVER	STATE
2	vol-ldg2@opath-ldg2	ldg2			standby
0	vol-ldg1@opath-vds	ldg1			active
1	vol-prim@primary-vds0	primary			standby

최소 Oracle Solaris 11.2 SRU 1 OS를 실행하지 않는 바운드 도메인의 mpgroup 디스크에 `ldm set-vdisk` 명령을 사용할 경우, 경로 우선순위 순서가 바뀌면서 OBP가 액세스할 때나 다음 디스크 연결/재부트 시 새 경로를 첫번째로 사용할 수 있습니다.

CD, DVD 및 ISO 이미지

가상 디스크를 내보내는 방법과 동일하게 CD(compact disc) 또는 DVD(digital versatile disc)를 내보낼 수 있습니다. CD 또는 DVD를 게스트 도메인으로 내보내려면 CD 또는 DVD 장치의 슬라이스 2를 전체 디스크로 내보냅니다(즉, `slice` 옵션 없이).

주 - CD 또는 DVD 드라이브 자체는 내보낼 수 없습니다. CD 또는 DVD 드라이브에 속한 CD 또는 DVD만 내보낼 수 있습니다. 따라서 CD 또는 DVD를 내보내려면 드라이브 안에 CD 또는 DVD가 있어야 합니다. 또한 CD 또는 DVD를 내보낼 수 있으려면 해당 CD 또는 DVD가 서비스 도메인에서 사용 중이 아니어야 합니다. 특히, 볼륨 관리 파일 시스템 `volfs` 서비스에서 CD 또는 DVD를 사용하지 않아야 합니다. `volfs`에서 사용 중인 장치를 제거하는 방법에 대한 자세한 내용은 [CD 또는 DVD를 서비스 도메인에서 게스트 도메인으로 내보내는 방법 \[182\]](#)을 참조하십시오.

파일이나 볼륨에 저장된 CD 또는 DVD의 ISO(International Organization for Standardization) 이미지가 있고 해당 파일이나 볼륨을 전체 디스크로 내보낼 경우 게스트 도메인에서는 CD 또는 DVD로 나타납니다.

CD, DVD 또는 ISO 이미지를 내보낼 경우 게스트 도메인에서는 자동으로 읽기 전용 장치로 나타납니다. 하지만 게스트 도메인에서 CD 제어 작업(즉, 게스트 도메인에서 CD를 시작, 중지 또는 꺼내기)을 수행할 수는 없습니다. 내보낸 CD, DVD 또는 ISO 이미지가 부트 가능한 경우 게스트 도메인을 해당 가상 디스크에서 부트할 수 있습니다.

예를 들어, Oracle Solaris OS 설치 DVD를 내보낼 경우 DVD에 해당하는 가상 디스크에서 게스트 도메인을 부트하고 해당 DVD에서 게스트 도메인을 설치할 수 있습니다. 이렇게 하려면 게스트 도메인에서 `ok` 프롬프트가 나타날 때 다음 명령을 사용해야 합니다.

```
ok boot /virtual-devices@100/channel-devices@200/disk@n:f
```

여기서 `n`은 내보낸 DVD를 나타내는 가상 디스크의 인덱스입니다.

주 - Oracle Solaris OS 설치 DVD를 내보내고 DVD에 해당하는 가상 디스크에서 게스트 도메인을 부트하여 게스트 도메인을 설치할 경우 설치 중에는 DVD를 변경할 수 없습니다. 따라서 다른 CD/DVD를 요청하는 설치 단계를 건너뛰거나 요청된 매체에 액세스하기 위한 대체 경로를 제공해야 할 수 있습니다.

▼ CD 또는 DVD를 서비스 도메인에서 게스트 도메인으로 내보내는 방법

1. 서비스 도메인에서 볼륨 관리 데몬 `volld`가 실행 중이고 온라인 상태인지 확인합니다.

```
service# svcs volfs
STATE          STIME      FMRI
online         12:28:12  svc:/system/filesystem/volfs:default
```

2. 볼륨 관리 데몬이 실행 중이고 온라인 상태인 경우 단계 1의 예에 나온 대로 다음을 수행합니다.

- a. `/etc/vold.conf` 파일에서 다음 단어로 시작하는 행을 주석 처리합니다.

```
use cdrom drive....
```

`vold.conf`(4) 매뉴얼 페이지를 참조하십시오.

- b. CD 또는 DVD 드라이브에 CD 또는 DVD를 넣습니다.

- c. 서비스 도메인에서 볼륨 관리 파일 시스템 서비스를 다시 시작합니다.

```
service# svcadm refresh volfs
service# svcadm restart volfs
```

3. 서비스 도메인에서 CD-ROM 장치에 대한 디스크 경로를 찾습니다.

```
service# cdrw -l
Looking for CD devices...
Node                               Connected Device                Device type
-----+-----
/dev/rdisk/c1t0d0s2 | MATSHITA CD-RW CW-8124 DZ13 | CD Reader/Writer
```

4. CD 또는 DVD 디스크 장치를 전체 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /dev/dsk/c1t0d0s2 cdrom@primary-vds0
```

5. 내보낸 CD 또는 DVD를 게스트 도메인에 지정합니다.

다음 명령은 내보낸 CD 또는 DVD를 도메인 `ldg1`에 지정하는 방법을 보여줍니다.

```
primary# ldm add-vdisk cdrom cdrom@primary-vds0 ldg1
```

CD 또는 DVD 여러 번 내보내기

CD 또는 DVD는 여러 번 내보내고 서로 다른 게스트 도메인에 지정할 수 있습니다. 자세한 내용은 [가상 디스크 백엔드를 여러 번 내보내는 방법 \[165\]](#)을 참조하십시오.

▼ 컨트롤 도메인에서 ISO 이미지를 내보내서 게스트 도메인을 설치하는 방법

시작하기 전에 이 절차에서는 primary 도메인과 게스트 도메인이 모두 구성되었다고 가정합니다.

예를 들어, 다음 ldm list는 primary 및 ldom1 도메인이 모두 구성되었음을 나타냅니다.

```
primary# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    8     8G     0.3%  15m
ldom1        active -t- - - 5000   4     1G     25%   8m
```

1. ISO 이미지를 내보낼 가상 디스크 서버 장치를 추가합니다.

이 예에서 ISO 이미지는 /export/images/sol-10-u8-ga-sparc-dvd.iso입니다.

```
primary# ldm add-vdsdev /export/images/sol-10-u8-ga-sparc-dvd.iso dvd-iso@primary-vds0
```

2. 게스트 도메인을 중지합니다.

이 예에서 논리적 도메인은 ldom1입니다.

```
primary# ldm stop-domain ldom1
LDom ldom1 stopped
```

3. ISO 이미지에 대한 가상 디스크를 논리적 도메인에 추가합니다.

이 예에서 논리적 도메인은 ldom1입니다.

```
primary# ldm add-vdisk s10-dvd dvd-iso@primary-vds0 ldom1
```

4. 게스트 도메인을 다시 시작합니다.

이 예에서 논리적 도메인은 ldom1입니다.

```
primary# ldm start-domain ldom1
LDom ldom1 started
# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active -n-cv  SP    8     8G     0.4%  25m
ldom1        active -t- - - 5000   4     1G     0.0%  0s
```

이 예에서 ldm list 명령은 ldom1 도메인이 방금 시작되었음을 보여줍니다.

5. 게스트 도메인에 연결합니다.

```
primary# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Connecting to console "ldom1" in group "ldom1" ....
Press ~? for control options ..
```

6. ISO 이미지가 가상 디스크로 존재하는지 확인합니다.

```
{0} ok show-disks
a) /virtual-devices@100/channel-devices@200/disk@1
```

```
b) /virtual-devices@100/channel-devices@200/disk@0
q) NO SELECTION
Enter Selection, q to quit: q
```

이 예에서 새로 추가된 장치는 /virtual-devices@100/channel-devices@200/disk@1입니다.

7. 게스트 도메인을 부트하여 ISO 이미지에서 설치합니다.

이 예에서는 /virtual-devices@100/channel-devices@200/disk@1 디스크의 f 슬라임에서 부트합니다.

```
{0} ok boot /virtual-devices@100/channel-devices@200/disk@1:f
```

가상 디스크 시간 초과

기본적으로 가상 디스크 백엔드에 대한 액세스를 제공하는 서비스 도메인이 작동 중지될 경우 게스트 도메인에서 해당 가상 디스크로의 모든 I/O가 차단됩니다. 서비스 도메인이 작동하고 가상 디스크 백엔드에 I/O 요청을 서비스하면 I/O는 자동으로 재개됩니다.

하지만 서비스 도메인이 오랫동안 작동 중지 상태일 경우 파일 시스템이나 응용 프로그램에서 I/O 작업이 차단되지 않고 실패하도록 하여 오류를 보고하려는 경우가 있습니다. 이제 각 가상 디스크에 대한 연결 시간 초과 기간을 설정하여 게스트 도메인의 가상 디스크 클라이언트와 서비스 도메인의 가상 디스크 서버 간에 연결을 설정하는 데 사용할 수 있습니다. 시간 초과 기간에 도달하면 서비스 도메인이 작동 중지되고 가상 디스크 클라이언트와 서버간 연결이 다시 설정되지 않는 동안 보류 중인 모든 I/O 및 새로운 I/O는 실패합니다.

다음 방법 중 하나를 사용하여 이 시간 초과를 설정하십시오.

- ldm add-vdisk 명령 사용

```
ldm add-vdisk timeout=seconds disk-name volume-name@service-name domain-name
```

- ldm set-vdisk 명령 사용

```
ldm set-vdisk timeout=seconds disk-name domain-name
```

- 게스트 도메인의 /etc/system 파일에 다음 행 추가

```
set vdc:vdc_timeout=seconds
```

/etc/system 등록 정보 값을 정확히 만들거나 업데이트하는 방법은 “[/etc/system 파일에서 등록 정보 값 업데이트](#)” [352]를 참조하십시오.

주 - 이 조정 가능 변수가 설정될 경우 ldm CLI를 사용하여 수행된 모든 시간 초과 설정을 덮어 씁니다. 또한 조정 가능 변수는 게스트 도메인의 모든 가상 디스크에 대한 시간 초과를 설정합니다.

시간 초과는 초 단위로 지정합니다. 시간 초과가 0으로 설정되면 시간 초과는 사용 안함으로 설정되고 서비스 도메인이 작동 중지된 동안 I/O가 차단됩니다(기본 설정 및 동작).

가상 디스크 및 SCSI

물리적 SCSI 디스크 또는 LUN을 전체 디스크로 내보낼 경우 해당하는 가상 디스크는 사용자 SCSI 명령 인터페이스 `uscsi` 및 다중 호스트 디스크 제어 작업 `mhd`를 지원합니다. 파일이나 볼륨을 백엔드로 가지는 가상 디스크와 같은 다른 가상 디스크는 이러한 인터페이스를 지원하지 않습니다.

주 - `mpgroup`과 SCSI 예약을 함께 사용할 수 없습니다.

결과적으로 SCSI 명령을 사용하는 응용 프로그램이나 제품 기능(예: Solaris Volume Manager `metaset` 또는 Oracle Solaris Cluster `shared devices`)은 물리적 SCSI 디스크를 백엔드로 가지는 가상 디스크가 있는 게스트 도메인에서만 사용할 수 있습니다.

주 - SCSI 작업은 사실상 가상 디스크 백엔드로 사용되는 물리적 SCSI 디스크 또는 LUN을 관리하는 서비스 도메인에 의해 실행됩니다. 특히 SCSI 예약은 서비스 도메인의 SCSI 명령으로 구현됩니다. 따라서 서비스 도메인 및 게스트 도메인에서 실행되는 응용 프로그램은 동일한 물리적 SCSI 디스크에 SCSI 명령을 실행하지 않아야 합니다. 실행할 경우 예상치 않은 디스크 상태가 발생할 수 있습니다.

가상 디스크 및 `format` 명령

`format` 명령은 도메인에 존재하는 모든 가상 디스크를 인식합니다. 하지만 단일 슬라이스 디스크로 내보낸 가상 디스크의 경우 `format` 명령은 가상 디스크의 분할 영역 테이블을 변경할 수 없습니다. 가상 디스크와 이미 연결된 디스크 레이블과 유사한 레이블을 작성하려고 시도하지 않으면 `label`과 같은 명령이 실패합니다.

백엔드가 SCSI 디스크인 가상 디스크는 모든 `format(1M)` 하위 명령을 지원합니다. 백엔드가 SCSI 디스크가 아닌 가상 디스크는 일부 `format(1M)` 하위 명령(`repair` 및 `defect` 등)을 지원하지 않습니다. 이 경우 `format(1M)`의 동작은 IDE(Integrated Drive Electronics) 디스크의 동작과 유사합니다.

가상 디스크에서 ZFS 사용

이 절에서는 ZFS(Zettabyte File System)를 사용하여 게스트 도메인으로 내보낸 가상 디스크 백엔드를 저장하는 방법을 설명합니다. ZFS는 가상 디스크 백엔드를 만들고 관리할 수 있는 편리하면서 강력한 솔루션을 제공합니다. ZFS를 통해 다음 작업을 수행할 수 있습니다.

- ZFS 볼륨 또는 ZFS 파일에 디스크 이미지 저장
- 스냅샷을 사용하여 디스크 이미지 백업
- 복제본을 사용하여 디스크 이미지 복제 및 추가 도메인 프로비전

ZFS 사용에 대한 자세한 내용은 [Oracle Solaris ZFS Administration Guide](#)를 참조하십시오.

다음 설명 및 예에서는 기본 도메인이 디스크 이미지가 저장되는 서비스 도메인이기도 합니다.

서비스 도메인에서 ZFS 풀 구성

디스크 이미지를 저장하려면 먼저 서비스 도메인에서 ZFS 저장소 풀을 만듭니다. 예를 들어, 이 명령은 `primary` 도메인에서 `c1t50d0` 디스크를 포함하는 ZFS 저장소 풀 `ldmpool`을 만듭니다.

```
primary# zpool create ldmpool c1t50d0
```

ZFS를 사용하여 디스크 이미지 저장

다음 명령은 게스트 도메인 `ldg1`에 대한 디스크 이미지를 만듭니다. 이 게스트 도메인에 대한 ZFS 파일 시스템이 만들어지고, 이 게스트 도메인의 모든 디스크 이미지가 해당 파일 시스템에 저장됩니다.

```
primary# zfs create ldmpool/ldg1
```

디스크 이미지는 ZFS 볼륨이나 ZFS 파일에 저장할 수 있습니다. `zfs create -v` 명령을 사용하면 크기에 상관없이 ZFS 볼륨이 빠르게 만들어집니다. 반면 ZFS 파일은 `mkfile` 명령을 사용하여 만들어야 합니다. 특히, 만들 파일이 큰 경우(대개 디스크 이미지를 만드는 경우) 이 명령을 완료하는 데 시간이 걸릴 수 있습니다.

ZFS 볼륨과 ZFS 파일은 스냅샷 및 복제본 기능과 같은 ZFS 기능을 활용하지만, ZFS 볼륨은 의사 장치이고 ZFS 파일은 일반 파일입니다.

디스크 이미지가 OS가 설치되는 가상 디스크로 사용될 경우 디스크 이미지는 OS 설치 요구 사항을 충족할 만큼 커야 합니다. 이 크기는 OS의 버전 및 수행되는 설치 유형에 따라 달라

집니다. Oracle Solaris OS를 설치하는 경우 20GB의 디스크 크기를 사용하면 모든 Oracle Solaris OS 버전의 모든 유형 설치가 가능합니다.

예: ZFS를 사용하여 디스크 이미지 저장

다음 예에서는 ZFS 볼륨 또는 ZFS 파일을 사용하여 디스크 이미지를 저장하는 방법을 보여 줍니다. ZFS 볼륨이나 파일을 내보내기 위한 구문은 동일하지만 백엔드에 대한 경로는 다릅니다.

게스트 도메인이 만들어지면 ZFS 볼륨이나 파일은 Oracle Solaris OS를 설치할 수 있는 가상 디스크로 나타납니다.

예 35 ZFS 볼륨을 사용하여 디스크 이미지 저장

먼저 ZFS 볼륨에 20GB의 이미지를 만듭니다.

```
primary# zfs create -V 20gb ldmpool/ldg1/disk0
```

그런 다음 ZFS 볼륨을 가상 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /dev/zvol/dsk/ldmpool/ldg1/disk0 ldg1_disk0@primary-vds0
```

ldg1 게스트 도메인에 ZFS 볼륨을 지정합니다.

```
primary# ldm add-vdisk disk0 ldg1_disk0@primary-vds0 ldg1
```

예 36 ZFS 파일을 사용하여 디스크 이미지 저장

먼저 ZFS 볼륨에 20GB의 디스크 이미지를 만들고 ZFS 파일을 만듭니다.

```
primary# zfs create ldmpool/ldg1/disk0
primary# mkfile 20g /ldmpool/ldg1/disk0/file
```

그런 다음 ZFS 파일을 가상 디스크로 내보냅니다.

```
primary# ldm add-vdsdev /ldmpool/ldg1/disk0/file ldg1_dis0@primary-vds0
```

ldg1 게스트 도메인에 ZFS 파일을 지정합니다.

```
primary# ldm add-vdisk disk0 ldg1_disk0@primary-vds0 ldg1
```

디스크 이미지의 스냅샷 만들기

디스크 이미지가 ZFS 볼륨이나 ZFS 파일에 저장된 경우 ZFS snapshot 명령을 사용하여 이 디스크 이미지의 스냅샷을 만들 수 있습니다.

디스크 이미지의 스냅샷을 만들기 전에 디스크가 현재 게스트 도메인에서 사용되고 있지 않은지 확인하여 디스크 이미지에 현재 저장된 데이터가 일관성을 유지하도록 합니다. 다음 방법 중 하나로 디스크가 게스트 도메인에서 사용 중이 아닌지 확인할 수 있습니다.

- 게스트 도메인을 중지하고 바인드 해제합니다. 이 방법은 가장 안전하며, 게스트 도메인의 부트 디스크로 사용되는 디스크 이미지의 스냅샷을 만들려는 경우 사용할 수 있는 유일한 방법입니다.
- 게스트 도메인에서 사용되는 스냅샷을 만들 디스크의 모든 슬라이스를 마운트 해제하고 게스트 도메인에서 사용 중인 슬라이스가 없는지 확인합니다.

이 예에서는 ZFS 레이아웃으로 인해 디스크 이미지가 ZFS 볼륨 또는 ZFS 파일에 저장되어 있는지 상관없이 디스크 이미지의 스냅샷을 만들기 위한 명령이 동일합니다.

예 37 디스크 이미지의 스냅샷 만들기

이 예에서는 `ldg1` 도메인에 대해 만들어진 디스크 이미지의 스냅샷을 만듭니다.

```
primary# zfs snapshot ldmpool/ldg1/disk0@version_1
```

복제본을 사용하여 새 도메인 프로비전

디스크 이미지의 스냅샷을 만들었으면 ZFS `clone` 명령을 사용하여 이 디스크 이미지를 복제할 수 있습니다. 그런 다음 복제된 이미지를 다른 도메인에 지정할 수 있습니다. 부트 디스크 이미지 복제는 전체 Oracle Solaris OS 설치 프로세스를 수행할 필요 없이 새로운 게스트 도메인에 대한 부트 디스크를 빠르게 만듭니다.

예를 들어, 만들어진 `disk0`이 `ldg1` 도메인의 부트 디스크인 경우 다음을 수행하여 해당 디스크를 복제하고 `ldg2` 도메인에 대한 부트 디스크를 만듭니다.

```
primary# zfs create ldmpool/ldg2
primary# zfs clone ldmpool/ldg1/disk0@version_1 ldmpool/ldg2/disk0
```

그런 다음 `ldmpool/ldg2/disk0`을 가상 디스크로 내보내고 새로운 `ldg2` 도메인에 지정할 수 있습니다. `ldg2` 도메인은 OS 설치 프로세스를 거칠 필요 없이 가상 디스크에서 직접 부트할 수 있습니다.

부트 디스크 이미지 복제

부트 디스크 이미지가 복제되면 새로운 이미지는 원본 부트 디스크와 완전히 동일하며, 이미지가 복제되기 전 부트 디스크에 저장된 모든 정보(호스트 이름, IP 주소, 마운트된 파일 시스템 테이블, 모든 시스템 구성 또는 조정 등)가 포함됩니다.

마운트된 파일 시스템 테이블은 원본 부트 디스크 이미지와 복제된 디스크 이미지에서 동일하므로 복제된 디스크 이미지를 원래 도메인에 있었던 같은 순서로 새로운 도메인에 지정해야 합니다. 예를 들어, 부트 디스크 이미지가 원래 도메인의 첫번째 디스크로 지정된 경우 복제된 디스크 이미지도 새로운 도메인의 첫번째 디스크로 지정해야 합니다. 그렇지 않으면 새로운 도메인이 부트되지 않습니다.

원래 도메인이 정적 IP 주소로 구성된 경우 복제된 이미지를 사용하는 새로운 도메인도 동일한 IP 주소로 시작됩니다. 이 경우 Oracle Solaris 11 `sysconfig unconfigure` 명령 또는 Oracle Solaris 10 `sys-unconfig` 명령을 사용하여 새 도메인의 네트워크 구성을 변경할 수 있습니다. 이 문제가 발생하지 않도록 하기 위해 구성이 해제된 시스템의 디스크 이미지 스냅샷을 만들 수도 있습니다.

원래 도메인이 DHCP(Dynamic Host Configuration Protocol)로 구성된 경우 복제된 이미지를 사용하는 새로운 도메인도 DHCP를 사용합니다. 이 경우 새로운 도메인이 부트될 때 IP 주소 및 네트워크 구성을 자동으로 가져오므로 새로운 도메인의 네트워크 구성을 변경할 필요가 없습니다.

주 - 도메인의 호스트 ID는 부트 디스크에 저장되지 않지만, 도메인을 만들 때 Logical Domains Manager에 의해 지정됩니다. 따라서 디스크 이미지를 복제할 경우 새로운 도메인은 원래 도메인의 호스트 ID를 유지하지 않습니다.

▼ 구성되지 않은 시스템의 디스크 이미지 스냅샷을 만드는 방법

1. 원래 도메인을 바인드하고 시작합니다.
2. 시스템의 구성을 해제합니다.
 - **Oracle Solaris 11 OS:** `sysconfig unconfigure` 명령을 실행합니다.
 - **Oracle Solaris 10 OS:** `sys-unconfig` 명령을 실행합니다.

이 작업이 완료되면 도메인이 정지됩니다.
3. 도메인을 중지하고 바인드를 해제합니다. 재부트하지 마십시오.
4. 도메인 부트 디스크 이미지의 스냅샷을 만듭니다.
예를 들어, 다음과 같습니다.

```
primary# zfs snapshot ldmpool/ldg1/disk0@unconfigured
```


이 단계에서 구성이 해제된 시스템의 부트 디스크 이미지 스냅샷이 생성됩니다.
5. 이 이미지를 복제하여 새로운 도메인을 만듭니다. 그러면 처음 부트할 때 시스템의 구성을 물어봅니다.

Oracle VM Server for SPARC 환경에서 볼륨 관리자 사용

이 절에서는 Oracle VM Server for SPARC 환경에서 볼륨 관리자를 사용하는 방법에 대해 설명합니다.

Volume Manager에서 가상 디스크 사용

ZFS, Solaris Volume Manager 또는 VxVM(Veritas Volume Manager) 볼륨은 서비스 도메인에서 가상 디스크로 게스트 도메인에 내보낼 수 있습니다. 볼륨은 단일 슬라이스 디스크 (slice 옵션이 `ldm add-vdsdev` 명령과 함께 지정된 경우) 또는 전체 디스크로 내보낼 수 있습니다.

주 - 이 절의 나머지 부분에서는 Solaris Volume Manager 볼륨을 예로 사용합니다. 하지만 설명은 ZFS 및 VxVM 볼륨에도 해당됩니다.

다음 예에서는 볼륨을 단일 슬라이스 디스크로 내보내는 방법을 보여줍니다.

게스트 도메인의 가상 디스크(예: `/dev/dsk/c0d2s0`)는 연결된 볼륨(예: `/dev/md/dsk/d0`)에 직접 매핑되고, 게스트 도메인에서 가상 디스크에 저장되는 데이터는 추가 메타데이터 없이 연결된 볼륨에 직접 저장됩니다. 게스트 도메인에서 가상 디스크에 저장된 데이터는 연관된 볼륨을 통해 서비스 도메인에서 직접 액세스할 수도 있습니다.

예

- Solaris Volume Manager 볼륨 `d0`을 `primary` 도메인에서 `domain1`로 내보낸 경우 `domain1`의 구성을 위해서는 일부 추가 단계가 필요합니다.

```
primary# metainit d0 3 1 c2t70d0s6 1 c2t80d0s6 1 c2t90d0s6
primary# ldm add-vdsdev options=slice /dev/md/dsk/d0 vol13@primary-vds0
primary# ldm add-vdisk vdisk3 vol13@primary-vds0 domain1
```

- `domain1`이 바인드되고 시작된 후 내보낸 볼륨은 `/dev/dsk/c0d2s0`(예)으로 나타나고 사용할 수 있습니다.

```
domain1# newfs /dev/rdisk/c0d2s0
domain1# mount /dev/dsk/c0d2s0 /mnt
domain1# echo test-domain1 > /mnt/file
```

- `domain1`이 중지되고 바인드 해제된 후 `domain1`에서 가상 디스크에 저장된 데이터는 Solaris Volume Manager 볼륨 `d0`을 통해 기본 도메인에서 직접 액세스할 수 있습니다.

```
primary# mount /dev/md/dsk/d0 /mnt
primary# cat /mnt/file
test-domain1
```

Solaris Volume Manager에서 가상 디스크 사용

RAID 또는 미리 Solaris Volume Manager 볼륨이 다른 도메인에서 가상 디스크로 사용되는 경우 배타적(*excl*) 옵션 설정 없이 내보내야 합니다. 그렇지 않으면 Solaris Volume Manager 볼륨 구성요소 중 하나에서 오류가 발생할 경우 *metareplace* 명령 또는 핫 스페어를 사용한 Solaris Volume Manager 볼륨 복구가 시작되지 않습니다. *metastat* 명령은 볼륨을 재동기화 중으로 인식하지만 재동기화가 진행되지 않습니다.

예를 들어, */dev/md/dsk/d0*은 *excl* 옵션을 사용하여 다른 도메인에 가상 디스크로 내보낸 RAID Solaris Volume Manager 볼륨이고, *d0*은 몇 가지 핫 스페어 장치로 구성되어 있습니다. *d0*의 구성요소가 실패할 경우 Solaris Volume Manager는 실패한 구성요소를 핫 스페어로 교체하고 Solaris Volume Manager 볼륨을 재동기화합니다. 하지만 재동기화가 시작되지 않습니다. 볼륨이 재동기화 중으로 보고되지만 재동기화가 진행되지 않습니다.

```
primary# metastat d0
d0: RAID
  State: Resyncing
  Hot spare pool: hsp000
  Interlace: 32 blocks
  Size: 20097600 blocks (9.6 GB)
Original device:
  Size: 20100992 blocks (9.6 GB)
Device                               Start Block  Dbase   State Reloc
c2t2d0s1                               330         No      Okay   Yes
c4t12d0s1                               330         No      Okay   Yes
/dev/dsk/c10t600C0FF0000000000015153295A4B100d0s1 330         No      Resyncing Yes
```

이러한 경우 Solaris Volume Manager 볼륨을 가상 디스크로 사용하는 도메인을 중지하고 바인드 해제하여 재동기화를 완료해야 합니다. 그런 다음 *metasync* 명령을 사용하여 Solaris Volume Manager 볼륨을 재동기화할 수 있습니다.

```
# metasync d0
```

VxVM이 설치된 경우 가상 디스크 사용

VxVM이 시스템에 설치되어 있고 DMP(Veritas Dynamic Multipathing)가 가상 디스크로 내보낼 물리적 디스크 또는 분할 영역에서 사용으로 설정된 경우 *excl* 옵션 설정 없이(기본값 아님) 해당 디스크 또는 분할 영역을 내보내야 합니다. 그렇지 않으면 해당 디스크를 사용하는 도메인을 바인드할 때 */var/adm/messages*에 오류가 나타납니다.

```
vd_setup_vd(): ldi_open_by_name(/dev/dsk/c4t12d0s2) = errno 16
vds_add_vd(): Failed to add vdisk ID 0
```

vxdisk list 출력에서 다중 경로 정보를 확인하여 Veritas DMP가 사용으로 설정되었는지 여부를 확인할 수 있습니다. 예를 들어, 다음과 같습니다.

```
# vxdisk list Disk_3
Device:      Disk_3
```

```

devicetag: Disk_3
type:      auto
info:      format=none
flags:     online ready private autoconfig invalid
pubpaths:  block=/dev/vx/dmp/Disk_3s2 char=/dev/vx/rdmp/Disk_3s2
guid:      -
udid:      SEAGATE%5FST336753LSUN36G%5FDISKS%5F3032333948303144304E0000
site:      -
Multipathing information:
numpaths:  1
c4t12d0s2  state=enabled

```

또는 `excl` 옵션이 설정된 상태에서 가상 디스크로 내보낼 디스크 또는 슬라이스에서 Veritas DMP가 사용으로 설정된 경우 `vxdmpadm` 명령을 사용하여 DMP를 사용 안함으로 설정할 수 있습니다. 예를 들어, 다음과 같습니다.

```
# vxdmpadm -f disable path=/dev/dsk/c4t12d0s2
```

가상 디스크에서 Volume Manager 사용

이 절에서는 가상 디스크에서 Volume Manager를 사용하는 방법에 대해 설명합니다.

가상 디스크에서 ZFS 사용

모든 가상 디스크는 ZFS로 사용할 수 있습니다. ZFS 저장소 풀(`zpool`)은 이 `zpool`의 일부인 저장소 장치를 인식(도메인에서 이러한 장치를 가상 장치 또는 실제 장치로 인식하는지 상관없이)하는 모든 도메인에서 가져올 수 있습니다.

가상 디스크에서 Solaris Volume Manager 사용

모든 가상 디스크는 Solaris Volume Manager 로컬 디스크 세트에서 사용할 수 있습니다. 예를 들어, 가상 디스크는 로컬 디스크 세트의 Solaris Volume Manager 메타 장치 상태 데이터베이스 `metadb`를 저장하고 로컬 디스크 세트에서 Solaris Volume Manager 볼륨을 만드는 데 사용할 수 있습니다.

백엔드가 SCSI 디스크인 모든 가상 디스크는 Solaris Volume Manager 공유 디스크 세트 `metaset`에서 사용할 수 있습니다. 백엔드가 SCSI 디스크가 아닌 가상 디스크는 Solaris Volume Manager 공유 디스크 세트에 추가할 수 없습니다. 백엔드가 SCSI 디스크가 아닌 가상 디스크를 Solaris Volume Manager 공유 디스크 세트에 추가하려고 시도하면 다음과 같은 오류와 함께 실패합니다.

```
# metaset -s test -a c2d2
metaset: domain1: test: failed to reserve any drives
```

가상 디스크에서 VxVM 사용

게스트 도메인에서 VxVM 지원은 Symantec의 VxVM 설명서를 참조하십시오.

가상 디스크 문제

다음 절에서는 가상 디스크를 사용할 때 발생할 수 있는 문제를 설명합니다.

특정 조건에서 게스트 도메인의 Solaris Volume Manager 구성 또는 메타 장치가 손실될 수 있음

서비스 도메인이 Oracle Solaris 10 1/13 OS 이전의 Oracle Solaris 10 OS 버전을 실행 중이고 게스트 도메인에 물리적 디스크 슬라이스를 가상 디스크로 내보내는 경우 이 가상 디스크는 게스트 도메인에서 잘못된 장치 ID로 표시됩니다. 해당 서비스 도메인을 다시 Oracle Solaris 10 1/13 OS로 업그레이드하면 가상 디스크로 내보낸 물리적 디스크 슬라이스가 게스트 도메인에 장치 ID 없이 표시됩니다.

이러한 가상 디스크의 장치 ID 제거로 인해 가상 디스크의 장치 ID를 참조하려고 시도하는 응용 프로그램에 문제가 발생할 수 있습니다. 특히, Solaris Volume Manager가 해당 구성을 찾을 수 없거나 해당 메타 장치에 액세스하지 못할 수 있습니다.

서비스 도메인을 Oracle Solaris 10 1/13 OS로 업그레이드한 후 게스트 도메인이 해당 Solaris Volume Manager 구성 또는 해당 메타 장치를 찾을 수 없는 경우 다음 절차를 수행하십시오.

▼ 게스트 도메인의 Solaris Volume Manager 구성 또는 메타 장치를 찾는 방법

1. 게스트 도메인을 부트합니다.
2. `/kernel/drv/md.conf` 파일에 다음 라인을 추가하여 Solaris Volume Manager의 `devid` 기능을 사용 안함으로 설정합니다.

```
md_devid_destroy=1;
md_keep_repl_state=1;
```

3. 게스트 도메인을 재부트합니다.
도메인이 부트되면 Solaris Volume Manager 구성 및 메타 장치를 사용할 수 있어야 합니다.

4. **Solaris Volume Manager** 구성이 올바른지 확인합니다.
5. `/kernel/drv/md.conf` 파일에서 2단계에서 추가한 두 행을 제거하여 **Solaris Volume Manager** `devid` 기능을 다시 사용으로 설정합니다.
6. 게스트 도메인을 재부트합니다.
재부트 중에 다음과 비슷한 메시지가 표시됩니다.

```
NOTICE: mddb: unable to get devid for 'vdc', 0x10
```

이러한 메시지는 정상이며 문제를 보고하지 않습니다.

Oracle Solaris 부트 디스크 호환성

지금까지 Oracle Solaris OS는 SMI VTOC 디스크 레이블로 구성된 부트 디스크에 설치되었습니다. Oracle Solaris 11.1 OS부터는 기본적으로 EFI(Extensible Firmware Interface) GPT(GUID 분할 영역 테이블) 디스크 레이블로 구성된 부트 디스크에 OS가 설치됩니다. UltraSPARC T2, UltraSPARC T2 Plus 및 SPARC T3 서버를 제외하고 지원되는 모든 서버의 현재 시스템 펌웨어 버전은 EFI 레이블을 지원합니다.

다음 서버의 경우 EFI GPT 디스크 레이블이 있는 디스크에서 부트할 수 없습니다.

- 모든 시스템 펌웨어 버전의 UltraSPARC T2, UltraSPARC T2 Plus 및 SPARC T3 서버
- 8.4.0 이전 버전의 시스템 펌웨어를 실행하는 SPARC T4 서버
- 9.1.0 이전 버전의 시스템 펌웨어를 실행하는 SPARC T5 서버, SPARC M5 서버 및 SPARC M6 서버
- 9.4.3 이전 버전의 시스템 펌웨어를 실행하는 SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버

따라서 최신 SPARC T4 서버, SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버 또는 SPARC M7 시리즈 서버에 만들어진 Oracle Solaris 11.1 부트 디스크는 이전 서버 또는 이전 펌웨어를 실행하는 서버에서 사용할 수 없습니다.

이러한 제한 사항으로 인해 콜드 또는 라이브 마이그레이션을 사용하여 도메인을 최신 서버에서 이전 서버로 이동할 수 없습니다. 또한 이전 서버에서 EFI GPT 부트 디스크 이미지를 사용할 수도 없습니다.

Oracle Solaris 11.1 부트 디스크가 서버 및 해당 펌웨어와 호환되는지 확인하려면 Oracle Solaris 11.1 OS가 SMI VTOC 디스크 레이블로 구성된 디스크에 설치되었는지 확인하십시오.

이전 펌웨어를 실행하는 시스템과의 역호환성을 유지하려면 다음 절차 중 하나를 사용하십시오. 그렇지 않으면 기본적으로 부트 디스크에 EFI GPT 디스크 레이블이 사용됩니다. 다음 절차는 시스템 펌웨어 버전 8.4.0 이상을 실행하는 SPARC T4 서버, 시스템 펌웨어 버전 9.1.0 이상을 실행하는 SPARC T5, SPARC M5, SPARC M6 서버, 시스템 펌웨어 버전 9.4.3 이상

을 실행하는 SPARC T7 시리즈 서버 또는 SPARC M7 시리즈 서버에서 SMI VTOC 디스크 레이블이 있는 부트 디스크에 Oracle Solaris 11.1 OS가 설치되었는지 확인하는 방법을 보여줍니다.

- **해결책 1:** 펌웨어에서 EFI를 지원하지 않도록 gpt 등록 정보를 제거합니다.

1. OpenBoot PROM 프롬프트에서 자동 부트를 사용 안함으로 설정하고 설치할 시스템을 재설정합니다.

```
ok setenv auto-boot? false
ok reset-all
```

시스템이 재설정되면 ok 프롬프트로 돌아옵니다.

2. /packages/disk-label 디렉토리로 변경하고 gpt 등록 정보를 제거합니다.

```
ok cd /packages/disk-label
ok " gpt" delete-property
```

3. Oracle Solaris 11.1 OS 설치를 시작합니다.

예를 들어 네트워크 설치를 수행합니다.

```
ok boot net - install
```

- **해결책 2:** format -e 명령을 사용하여 Oracle Solaris 11.1 OS와 함께 설치할 디스크에 SMI VTOC 레이블을 씁니다.

1. 디스크에 SMI VTOC 레이블을 씁니다.

예를 들어 label 옵션을 선택하고 SMI 레이블을 지정합니다.

```
# format -e c1d0
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
```

2. 전체 디스크를 포함하는 슬라이스 0과 슬라이스 2로 디스크를 구성합니다.

디스크에 다른 분할 영역이 없어야 합니다. 예를 들어, 다음과 같습니다.

```
format> partition
```

```
partition> print
```

```
Current partition table (unnamed):
```

```
Total disk cylinders available: 14087 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 14086	136.71GB	(14087/0/0) 286698624
1	unassigned	wu	0	0	(0/0/0) 0
2	backup	wu	0 - 14086	136.71GB	(14087/0/0) 286698624
3	unassigned	wm	0	0	(0/0/0) 0

```
4 unassigned wm 0 0 (0/0/0) 0
5 unassigned wm 0 0 (0/0/0) 0
6 unassigned wm 0 0 (0/0/0) 0
7 unassigned wm 0 0 (0/0/0) 0
```

3. SMI VTOC 디스크 레이블을 다시 씁니다.

```
partition> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 0
Ready to label disk, continue? y
```

4. 부트 디스크의 슬라이스 0에 Oracle Solaris OS를 설치하도록 Oracle Solaris AI(자동 설치 프로그램)를 구성합니다.

AI 매니페스트의 <disk> 발췌 부분을 다음과 같이 변경합니다.

```
<target>
  <disk whole_disk="true">
    <disk_keyword key="boot_disk"/>
    <slice name="0" in_zpool="rpool"/>
  </disk>
[...]
```

5. Oracle Solaris 11.1 OS 설치를 수행합니다.

◆◆◆ 12 장

가상 SCSI 호스트 버스 어댑터 사용

이 장에서는 Oracle VM Server for SPARC 소프트웨어에서 가상 SCSI HBA(호스트 버스 어댑터)를 사용하는 방법에 대해 설명합니다.

이 장에서는 다음 주제를 다룹니다.

- “가상 SCSI 호스트 버스 어댑터 소개” [197]
- “가상 SCSI HBA의 작동 모델” [199]
- “가상 SCSI HBA 식별자 및 장치 이름” [200]
- “가상 SCSI HBA 관리” [201]
- “게스트 도메인에서 가상 LUN의 모양” [204]
- “가상 SCSI HBA 및 가상 SAN 구성” [205]
- “가상 SCSI HBA 다중 경로 구성” [205]
- “SCSI 장치에서 부트” [212]
- “가상 LUN 설치” [213]
- “가상 SCSI HBA 시간 초과” [213]
- “가상 SCSI HBA 및 SCSI” [214]
- “게스트 도메인에서 단편화가 심한 I/O 버퍼 지원” [214]

가상 SCSI 호스트 버스 어댑터 소개

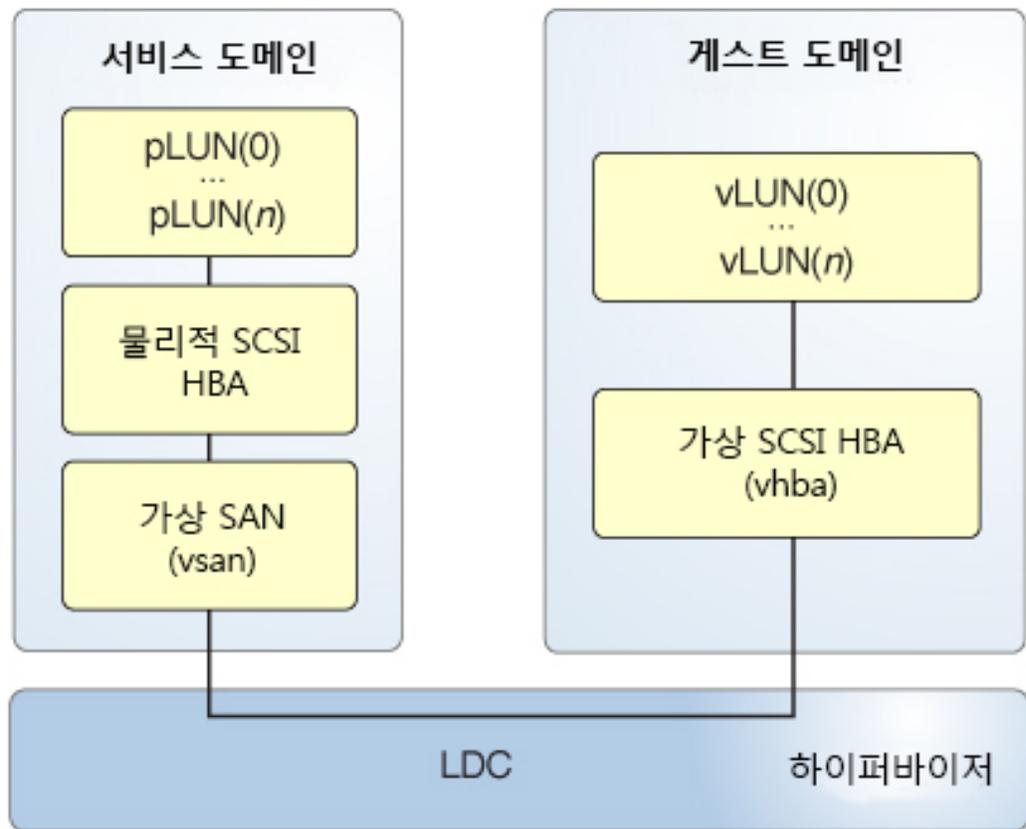
가상 SCSI HBA(호스트 버스 어댑터)는 게스트 도메인의 가상 HBA와 서비스 도메인의 가상 SAN(저장 영역 네트워크)이라는 두 가지 구성요소로 이루어집니다. 가상 HBA와 가상 SAN 인스턴스는 상호 협력하여 게스트 도메인에서 실행하는 SCSI 대상 드라이버의 SCSI HBA 인터페이스를 구현합니다. vSAN 서비스는 v_{san} 드라이버로 구현되며, 서비스 도메인에서 실행하는 물리적 SCSI HBA 드라이버로 SCSI I/O 요청을 전달합니다. v_{hba} 드라이버는 하이퍼바이저로 관리되는 LDC(논리적 도메인 채널)를 사용하여 v_{san}으로 I/O 요청을 보냅니다.

vHBA 인스턴스는 특정 vSAN 인스턴스로 연결할 수 있는 모든 SCSI 장치에 대한 액세스를 제공합니다. vHBA는 디스크, CD, DVD, 테이프와 같은 SCSI 장치 유형을 인식할 수 있습니다. 연결 가능한 SCSI 장치 세트는 현재 가상 SAN과 연관된 물리적 HBA 드라이버에 알려진 물리적 SCSI 장치 세트에 따라 달라집니다. 특정 vHBA에 알려진 SCSI 장치의 ID와 번호는 런타임까지 알 수 없으며, 물리적 HBA 드라이버 역시 마찬가지입니다.

vHBA는 하위 장치로 가상 LUN(vLUN)을 포함하며 물리적 LUN과 같은 방법으로 작동합니다. 예를 들어, vHBA 인스턴스 및 해당 vLUN과 함께 Oracle Solaris I/O 다중 경로 솔루션을 사용할 수 있습니다. vLUN의 장치 경로는 전체 cXtYdZsN 표기법: /dev/[r]disk/cXtYdZsN을 사용합니다. 장치 이름의 tY 부분은 SCSI 대상 장치를 나타냅니다.

가상 SAN과 가상 SCSI HBA를 구성한 후에 OpenBoot 프롬프트에서 가상 LUN을 부트하거나 format 명령을 사용하여 모든 가상 LUN을 보는 등의 작업을 수행할 수 있습니다.

그림 8 Oracle VM Server for SPARC의 가상 SCSI HBA



가상 SAN은 서비스 도메인에 존재하고 vsan 커널 모듈로 구현되며, 가상 SCSI HBA는 게스트 도메인에 존재하고 vhba 모듈로 구현됩니다. 가상 SAN은 특정 물리적 SCSI HBA 개시자 포트와 연관되고, 가상 SCSI HBA는 특정 가상 SAN과 연관됩니다.

vhba 모듈은 SCSSA 준수 인터페이스를 내보내어 SCSSA 준수 SCSI 대상 드라이버에서 I/O 요청을 받습니다. vhba 모듈은 I/O 요청을 가상 I/O 프로토콜 메시지로 번역하여 LDC를 통해 서비스 도메인으로 전송합니다.

vsan 모듈은 vhba가 보낸 가상 I/O 메시지를 I/O 요청으로 번역합니다. 이 요청은 SCSSA 준수 물리적 SCSI HBA 드라이버로 보내집니다. vsan 모듈은 I/O 페이로드 및 상태를 LDC를 통해 vhba로 반환합니다. 마지막으로, vhba는 이 I/O 응답을 I/O 요청 개시자에게 전달합니다.

가상 SCSI HBA의 작동 모델

가상 SCSI HBA와 가상 SAN 인스턴스만 Logical Domains Manager에 알려지기 때문에 가상 SCSI HBA 사용을 위한 작동 모델은 다른 유형의 Oracle VM Server for SPARC 가상 장치와 질적으로 다릅니다. 게스트 도메인에 나타나는 가상 LUN과 서비스 도메인에 나타나는 물리적 LUN은 런타임에 검색할 때까지 알 수 없습니다. 가상 LUN과 물리적 LUN은 연관된 LDC 연결을 재설정할 때 암시적으로 검색되고, `ldm rescana-vhba` 명령을 사용하여 명시적으로 검색됩니다.

`ldm` 명령을 사용하여 가상 디스크 이름을 명시적으로 지정하더라도, 게스트 도메인의 가상 LUN은 서비스 도메인의 연관된 물리적 LUN ID로부터 해당 ID를 도출합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예를 들어, 물리적 LUN과 가상 LUN은 다음 장치 경로에서 굵게 표시된 텍스트를 공유합니다.

- 서비스 도메인의 물리적 LUN:

```
/pci@0/pci@0/pci@8/pci@0/pci@2/SUNW,q1c@0/fp@0,0/ssd@w216000c0ff8089d5,0
```

- 게스트 도메인의 가상 LUN:

```
/virtual-devices@100/channel-devices@200/scsi@1/iproto@0/disk@w216000c0ff8089d5,0
```

주 - 게스트 도메인 장치 경로는 Oracle Solaris I/O 다중 경로가 게스트 도메인에서 사용 안 함으로 설정되었을 때만 존재합니다. Oracle Solaris I/O 다중 경로가 사용으로 설정되면 `scsi_vhci` 모듈은 게스트 도메인에 장치 경로를 만들며 여기에는 다른 구문이 사용됩니다.

가상 LUN의 장치 경로에 있는 `scsi@1` 구성요소는 이 가상 LUN이 멤버로 속한 가상 SCSI HBA 인스턴스를 나타냅니다.

가상 SCSI HBA의 가상 LUN 세트가 런타임에 서비스 도메인에서 파생되기 때문에 가상 LUN을 게스트 도메인에서 명시적으로 추가/제거할 수 없습니다. 대신, 게스트 도메인의 가상 LUN 멤버십을 변경할 수 있도록 원래 물리적 LUN을 추가/제거해야 합니다. 도메인 재부트나 도메인 마이그레이션과 같은 이벤트가 발생하면 게스트 도메인의 가상 LUN 멤버십이 변경될 수 있습니다. 이러한 변경은 가상 SCSI HBA의 LDC 연결을 재설정할 때마다 자동으로 가상 LUN이 재검색되기 때문에 발생합니다. 추후 검색 시 가상 LUN의 원래 물리적 LUN을 찾을 수 없으면 가상 LUN이 사용 불가능으로 표시되고 가상 LUN에 액세스하면 다음과 비슷한 오류가 반환됩니다.

```
WARNING: .../scsi@1/iport@0/disk@w216000c0ff8089d5,0 (sd6): ... Command failed to complete...Device is gone
```

가상 SCSI HBA 인스턴스는 `vhba` 드라이버로 관리되지만, 가상 LUN은 원래 물리적 LUN의 장치 유형에 준하여 SCSI 대상 드라이버로 관리됩니다. 다음 출력은 `vhba` 드라이버가 가상 SCSI HBA 인스턴스를 관리하고 `sd` SCSI 디스크 드라이버가 가상 LUN을 관리함을 보여줍니다.

```
# prtconf -a -D /dev/dsk/c2t216000C0FF8089D5d0
SUNW,SPARC-Enterprise-T5220 (driver name: rootnex)
  virtual-devices, instance #0 (driver name: vnex)
    channel-devices, instance #0 (driver name: cnex)
      scsi, instance #0 (driver name: vhba)
        iport, instance #3 (driver name: vhba)
          disk, instance #30 (driver name: sd)
```

가상 SCSI HBA 식별자 및 장치 이름

`ldm add-vhba` 명령을 사용하여 가상 SCSI HBA를 도메인에 추가하는 경우 `id` 등록 정보를 설정하여 해당 장치 번호를 지정할 수 있습니다.

```
ldm add-vhba [id=vHBA-ID] vHBA-name vSAN-name domain-name
```

도메인의 각 가상 SCSI HBA는 도메인이 바인드될 때 지정되는 고유한 장치 번호를 가집니다. `id` 등록 정보를 십진수 값으로 설정하여 명시적 장치 번호로 가상 SCSI HBA를 추가한 경우 지정된 장치 번호가 사용됩니다. 그렇지 않은 경우 시스템에서 사용 가능한 가장 낮은 장치 번호를 자동으로 지정합니다. 이 경우 지정되는 장치 번호는 도메인에 가상 SCSI HBA가 추가된 방법에 따라 달라집니다. 결국 가상 SCSI HBA에 지정되는 장치 번호는 도메인이 바인드될 때 `ldm list-bindings` 및 `ldm list -o hba` 명령의 출력에 표시됩니다.

`ldm list-bindings`, `ldm list -o hba`, `ldm add-vhba id=id` 명령은 모두 `id` 등록 정보 값을 십진수 값으로 표시하고 지정합니다. Oracle Solaris OS는 가상 SCSI HBA `id` 값을 16진수로 표시합니다.

다음 예제는 `gdom` 도메인에서 `vh1` 가상 SCSI HBA의 장치 이름이 `vhba@0` 장치임을 보여줍니다.

```
primary# ldm list -o hba gdom
NAME
gdom

VHBA
  NAME          VSAN          DEVICE TOUT SERVER
  vh1           vs1           vhba@0  0      svcdom
```



주의 - 장치 번호가 가상 SCSI HBA에 명시적으로 지정되지 않은 경우 도메인이 바인드 해제되고 나중에 재바인드될 때 해당 장치 번호를 변경할 수 있습니다. 이 경우 도메인에서 실행 중인 OS에서 지정한 장치 이름도 변경할 수 있으며 시스템의 기존 구성이 달라집니다. 예를 들어, 가상 SCSI HBA가 도메인의 구성에서 제거될 경우 이러한 상황이 발생할 수 있습니다.

가상 SCSI HBA 관리

이 절에서는 다음 작업을 다룹니다.

- “물리적 SCSI HBA 정보 얻기” [201]
- “가상 저장 영역 네트워크 만들기” [202]
- “가상 SCSI 호스트 버스 어댑터 만들기” [202]
- “가상 SCSI HBA의 존재 여부 확인” [203]
- “가상 SCSI HBA 시간 초과 옵션 설정” [203]
- “가상 SCSI 호스트 버스 어댑터 제거” [203]
- “가상 저장 영역 네트워크 제거” [204]
- “LUN 추가 또는 제거” [204]

이 절에 표시된 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

물리적 SCSI HBA 정보 얻기

가상 SCSI HBA를 구성하기 전에 서비스 도메인에 연결된 물리적 SCSI HBA에 대한 정보를 얻어야 합니다. I/O 도메인에서 HBA 카드 구성에 대한 자세한 내용은 [6장. I/O 도메인 구성](#)을 참조하십시오.

주 - 최소 Oracle Solaris 11.3 OS가 `primary` 도메인에 설치된 경우 서비스 도메인이 컨트롤 도메인이 될 수 있습니다.

`ldm list-hba` 명령은 지정된 활성 도메인에 대한 물리적 SCSI HBA 개시자 포트를 나열합니다. 논리적 도메인의 SCSI HBA 개시자 포트를 식별한 후에 `ldm add-vsan` 명령줄에 특정 개시자 포트를 지정하여 가상 SAN을 만들 수 있습니다.

```
ldm list-hba [-d] [-l] [-p] [-t] domain-name
```

다음 예제는 `svcdom` 서비스 도메인에 연결된 SCSI HBA의 개시자 포트를 보여줍니다. `-l` 옵션은 자세한 정보를 보여줍니다.

```
primary# ldm list-hba -l svcdom
NAME                               VSAN
----                               ----
/SYS/MB/SASHBA0/HBA0/PORT1
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@1]
/SYS/MB/SASHBA0/HBA0/PORT2
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@2]
/SYS/MB/SASHBA0/HBA0/PORT4
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@4]
/SYS/MB/SASHBA0/HBA0/PORT8
[/pci@300/pci@1/pci@0/pci@2/scsi@0/iport@8]
/SYS/MB/PCIE1/HBA0/PORT0,0
[/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0/fp@0,0]
/SYS/MB/PCIE1/HBA0,1/PORT0,0
```

```
[/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0,1/fp@0,0]
```

개시자 포트에 대해 예상한 LUN이 `ldm list-hba` 출력에 나타나지 않으면 참조된 개시자 포트에 대해 참조된 서비스 도메인에서 다중 경로가 사용 안함으로 설정되었는지 확인합니다. [Managing SAN Devices and Multipathing in Oracle Solaris 11.3](#)를 참조하십시오.

가상 저장 영역 네트워크 만들기

물리적 SCSI HBA의 개시자 포트를 얻은 후에 서비스 도메인에 가상 SAN(저장 영역 네트워크)을 만들어야 합니다. 가상 SAN은 지정된 SCSI HBA 개시자 포트에서 연결할 수 있는 모든 SCSI 장치를 관리합니다.

```
ldm add-vsan [-q] iport-path vSAN-name domain-name
```

vSAN 이름은 시스템에 고유하며 지정된 도메인 이름에는 고유하지 않습니다. 도메인 이름은 SCSI HBA 개시자 포트가 구성된 도메인을 식별합니다. 동일한 개시자 포트 경로를 참조하는 가상 SAN을 여러 개 만들 수 있습니다.

동일한 개시자 포트 경로에서 가상 SAN을 여러 개 만들 수 있습니다. 이 작업은 여러 게스트 도메인이 동일한 개시자 포트를 사용할 수 있도록 해줍니다.

주 - Oracle Solaris 11.3 OS가 서비스 도메인에서 실행 중인 경우 `ldm add-vsan` 명령은 개시자 포트 경로가 유효한 장치 경로인지 확인합니다. `ldm add-vsan` 명령을 실행할 때 지정된 서비스 도메인이 활성이 아니면 지정된 개시자 포트 경로를 서비스 도메인에서 확인할 수 없습니다. 개시자 포트 경로가 서비스 도메인에 설치된 물리적 SCSI HBA 개시자 포트와 일치하지 않으면 서비스 도메인이 활성화된 후 서비스 도메인의 시스템 로그에 경고 메시지가 기록됩니다.

이 예제에서 `svcdom` 서비스 도메인의 `/SYS/MB/PCIE1/HBA0,1/PORT0,0` 개시자 포트를 가상 SAN과 연관시킵니다. 가상 SAN의 이름을 선택할 수 있습니다. 이 예제에서 가상 SAN의 이름은 `port0`입니다.

```
primary# ldm add-vsan /SYS/MB/PCIE1/HBA0,1/PORT0,0 port0 svcdom
/SYS/MB/PCIE1/HBA0,1/PORT0,0 resolved to device:
/pci@300/pci@1/pci@0/pci@4/SUNW,emlxs@0,1/fp@0,0
```

가상 SCSI 호스트 버스 어댑터 만들기

가상 SAN이 정의된 후에 `ldm add-vhba` 명령을 사용하여 게스트 도메인에 가상 SCSI HBA를 만들 수 있습니다. 가상 SCSI HBA는 가상 SAN의 물리적 SCSI 장치로 I/O 요청을 보냅니다.

```
ldm add-vhba [id=vHBA-ID] vHBA-name vSAN-name domain-name
```

이 예제에서 `gdom` 게스트 도메인에 `port0` 가상 SAN과 통신하는 `port0_vhba` 가상 SCSI HBA를 만듭니다.

```
primary# ldm add-vhba port0_vhba port0 gdom
```

가상 SCSI HBA의 존재 여부 확인

`ldm list` 명령을 사용하여 서비스 도메인과 게스트 도메인에서 새로 만든 가상 SCSI HBA와 가상 SAN 장치의 존재 여부를 확인합니다.

```
ldm list -o san,hba [domain-name ...]
```

이 예제에서 가상 SAN이 있는 서비스 도메인은 `svcdom`이고 가상 SCSI HBA가 있는 게스트 도메인은 `gdom`입니다. `gdom` 도메인은 아직 바인드되지 않았기 때문에 이 예제에서 가상 HBA 식별자는 할당되지 않습니다.

```
primary# ldm list -o san,hba svcdom gdom
NAME
svcdom

VSAN
NAME          TYPE  DEVICE IPOR
port0         VSAN  [/pci@300/pci@1/pci@0/pci@4/SUNw,emlxs@0,1/fp@0,0]

-----
NAME
gdom

VHBA
NAME          VSAN          DEVICE TOUT SERVER
port0_vhba    port0         0      svcdom
```

가상 SCSI HBA 시간 초과 옵션 설정

`ldm set-vhba` 명령으로 지정된 논리적 도메인에서 가상 SCSI HBA의 시간 초과 값을 지정할 수 있습니다. `timeout` 등록 정보는 가상 SAN과 LDC 연결을 만들 수 없다고 선언하기 전에 지정된 가상 SCSI HBA 인스턴스의 대기 시간(초)을 지정합니다. [“가상 SCSI HBA 시간 초과” \[213\]](#)를 참조하십시오.

기본 시간 초과 값 0은 가상 SAN과 LDC 연결을 만들기 위해 `vhba`가 무기한 기다립니다.

```
ldm set-vhba [timeout=seconds] vHBA-name domain-name
```

이 예에서는 `gdom` 게스트 도메인에서 `port0_vhba` 가상 SCSI HBA에 대해 시간 초과를 90초로 설정합니다.

```
primary# ldm set-vhba timeout=90 port0_vhba gdom
```

가상 SCSI 호스트 버스 어댑터 제거

`ldm remove-vhba` 명령을 사용하여 지정된 게스트 도메인에서 가상 SCSI HBA를 제거할 수 있습니다.

가상 SCSI HBA를 제거하기 전에 어떤 OS나 응용 프로그램에서도 이를 활발히 사용 중이 아니어야 합니다. 가상 SCSI HBA가 사용 중인 경우 `ldm remove-vhba` 명령을 실패합니다.

```
ldm remove-vhba vHBA-name domain-name
```

이 예제에서 `gdom` 게스트 도메인에서 `port0_vhba` 가상 SCSI HBA를 제거합니다.

```
primary# ldm remove-vhba port0_vhba gdom
```

가상 저장 영역 네트워크 제거

`ldm remove-vsan` 명령을 사용하여 가상 SAN을 제거할 수 있습니다.

먼저, 가상 SAN과 연관된 가상 SCSI HBA를 제거합니다. 그 다음, `ldm remove-vsan` 명령을 사용하여 가상 SAN을 제거합니다.

```
ldm remove-vsan vSAN-name
```

이 예제에서 `port0` 가상 SAN을 제거합니다.

```
primary# ldm remove-vsan port0
```

LUN 추가 또는 제거

가상 SCSI HBA에서 직접 가상 LUN을 추가/제거할 수 없습니다. 먼저 물리적 LUN을 추가/제거한 다음, `ldm rescan-vhba` 명령을 실행하여 가상 SCSI HBA와 가상 SAN에 보이는 SCSI 장치 세트를 동기화해야 합니다. 물리적 LUN을 추가/제거하는 명령은 가상 SAN과 연관된 개시자 포트의 토폴로지에 따라 다릅니다. 예를 들어, 개시자 포트가 물리적 SAN과 통신하는 경우 SAN 관리 명령을 사용하여 SAN 요소에서 LUN을 추가하거나 LUN을 제거해야 합니다.

```
ldm rescan-vhba vHBA-name domain-name
```

예를 들어, 다음 명령은 `gdom` 도메인에서 `port0_vhba` 가상 SCSI HBA의 SCSI 장치를 동기화합니다.

```
primary# ldm rescan-vhba port0_vhba gdom
```

게스트 도메인에서 가상 LUN의 모양

가상 LUN이나 가상 SCSI HBA와 연관된 LUN은 물리적 LUN처럼 작동합니다.

예를 들어, SCSI 디스크를 표현하는 가상 LUN은 게스트 도메인에서 `/dev/[r]dsk` 아래에 일반 디스크로 나타납니다. 원래 연관된 물리적 LUN이 `disk` 유형이기 때문에 `format` 명령의 출력에서 가상 LUN을 볼 수 있습니다. 가상 LUN의 장치 경로는 `prtpic1` 및 `prtconf` 등의 다른 명령으로 작동할 수 있습니다.

가상 LUN이 CD/DVD를 표현하는 경우 해당 장치 경로는 `/dev/[r]dsk`에 정의됩니다. 가상 LUN이 테이프 장치를 표현하는 경우 해당 장치 경로는 `/dev/rmt`에 정의됩니다. 가상 LUN에서 작동할 수 있는 명령은 [Managing Devices in Oracle Solaris 11.3](#)를 참조하십시오.

가상 SCSI HBA 및 가상 SAN 구성

가상 SCSI HBA와 가상 SAN 구성은 매우 유연성이 있습니다. `ldm add-vsan` 명령에서 사용되는 물리적 SCSI HBA 개시자 포트는 광 섬유 채널, SAS, SATA 등 SCSI를 지원하는 모든 유형의 버스를 구동할 수 있습니다. 가상 SCSI HBA와 가상 SAN은 동일한 도메인에서 실행될 수 있습니다. 보다 일반적인 구성에서는 가상 SAN이 물리적 HBA 카드에 대한 직접 액세스 권한을 가지는 서비스 도메인에서 실행되는 경우 가상 SCSI HBA와 가상 SAN이 다른 도메인에서 실행됩니다.

가상 SAN은 개념상 물리적 SAN과 연관되지만 꼭 그럴 필요는 없습니다. 예를 들어, 마더보드 HBA에서 연결 가능한 일련의 로컬 저장 장치로 구성된 가상 SAN을 만들 수 있습니다.

가상 HBA 부속 시스템은 검색된 각 물리적 LUN에 대해 무조건 가상 LUN을 만듭니다. 따라서 가상 디스크와 마찬가지로 충돌하는 작업 부하가 동일한 가상 LUN에 액세스하지 않도록 해야 합니다.

예를 들어, 개시자 포트가 10개의 물리적 SCSI 장치에 연결되는 경우 가상 HBA 부속 시스템이 게스트 도메인에 10개의 가상 LUN을 만듭니다. 게스트 운영체제가 가상 LUN 중 하나에서 부트되는 경우 다른 게스트 도메인이 가상 LUN에 액세스하지 않고 물리적 SCSI 장치를 소유하는 도메인이 물리적 LUN에 액세스하지 않도록 해야 합니다.

게스트 도메인이 사용 중일 수 있는 가상 LUN에 대해 유사한 경고가 제공됩니다. 액세스가 충돌하지 않도록 하려면 다른 게스트 도메인의 해당 가상 LUN에 대한 액세스와 서비스 도메인의 기본 물리적 LUN에 대한 액세스를 엄격하게 제어해야 합니다. 액세스 충돌이 발생하면 데이터가 손상될 수 있습니다.

가상 SAN을 구성할 때 LUN 0의 SCSI 대상 장치만 해당 물리적 LUN이 게스트 도메인에 표시됩니다. 이 제약 조건에 따라 Oracle Solaris OS 구현에서 대상의 LUN 0이 SCSI REPORT LUNS 명령에 응답해야 합니다.

가상 SCSI HBA 다중 경로 구성

가상 SCSI HBA 부속 시스템은 Oracle Solaris I/O 다중 경로 구현을 활용하여 게스트 도메인 및 서비스 도메인에서 다중 경로를 지원합니다. 자세한 내용은 [Managing SAN Devices and Multipathing in Oracle Solaris 11.3](#)를 참조하십시오.

Oracle Solaris I/O 다중 경로에서와 마찬가지로, 하나 이상의 경로를 통해 특정 백엔드 SCSI 장치에 액세스할 수 있습니다. 가상 SCSI HBA 부속 시스템의 경우 각 경로는 하나의 가상 LUN과 연관됩니다. `scsi_vhci` 모듈은 Oracle Solaris I/O 다중 경로 동작을 구현합니다. 연관된 `mpathadm` 관리 명령에 전달된 인수를 기반으로 가상 LUN 세트로 I/O 요청을 보냅니다. 자세한 내용은 [scsi_vhci\(7D\)](#) 및 [mpathadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

그림 10에 나와 있는 것처럼 다중 경로가 서비스 도메인에서 사용으로 설정된 경우 `ldm add-vsan` 명령을 사용하면 지정된 개시자 포트를 통해 연결할 수 있는 SCSI 장치를 참조하는 모든 경로를 나타내는 `vsan` 인스턴스를 만들 수 있습니다. 하지만 다중 경로가 서비스 도메인에서 사용 안함으로 설정된 경우 `vsan` 인스턴스는 지정된 개시자 포트에서 발생하고 SCSI 장치를 참조하는 해당 경로만 나타냅니다.

다중 경로를 구성하려면 게스트 도메인 또는 서비스 도메인에서 동일한 백엔드 장치로 향하는 고유 경로를 두 개 이상 구성해야 합니다. 다중 경로는 여전히 하나의 구성 경로로 작동합니다. 그러나 예상 구성에는 고유의 서비스 도메인에 상주하는 고유의 물리적 SCSI HBA 개시자 포트를 통해 I/O 요청을 보내는 경로가 두 개 이상 있습니다.

1. 백엔드 저장소로 가는 개별 경로마다 `ldm add-vhba`와 `ldm add-vsan` 명령 쌍을 실행합니다.
2. `vhba` 가상 HBA 모듈로 관리되는 개시자 포트에 대해 게스트 도메인에서 Oracle Solaris I/O 다중 경로를 사용으로 설정합니다.

다음 그림은 게스트 도메인의 다중 경로 구성에 대한 예입니다. 여기서는 Oracle Solaris I/O 다중 경로에서 관리하는 경로 두 개로 액세스되는 SAN의 물리적 LUN 하나를 보여 줍니다. 이 그림에 표시된 구성을 만드는 방법을 설명하는 절차는 [가상 SCSI HBA 다중 경로를 구성하는 방법 \[208\]](#)을 참조하십시오.

그림 9 게스트 도메인에서 가상 SCSI HBA 다중 경로 구성

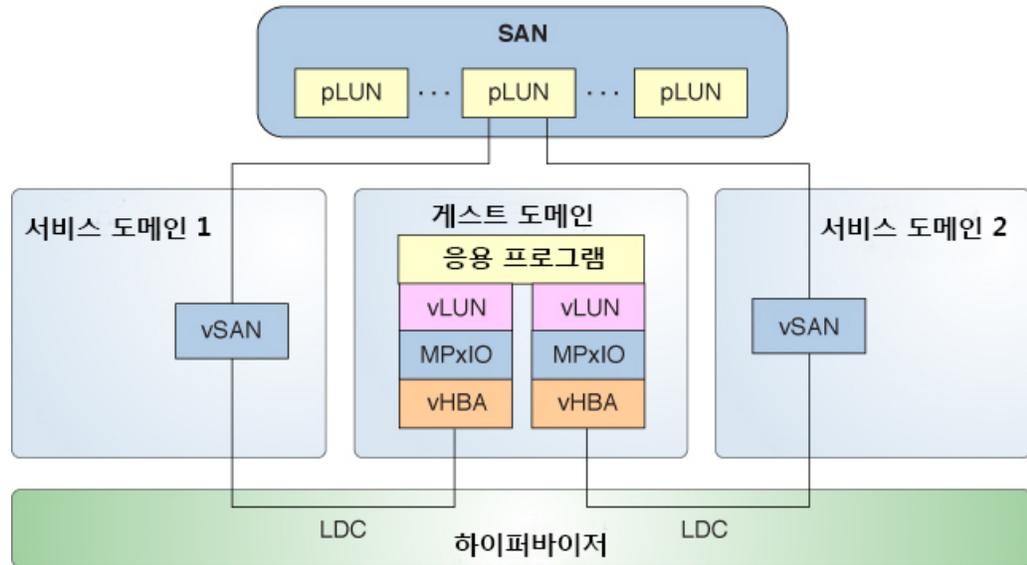
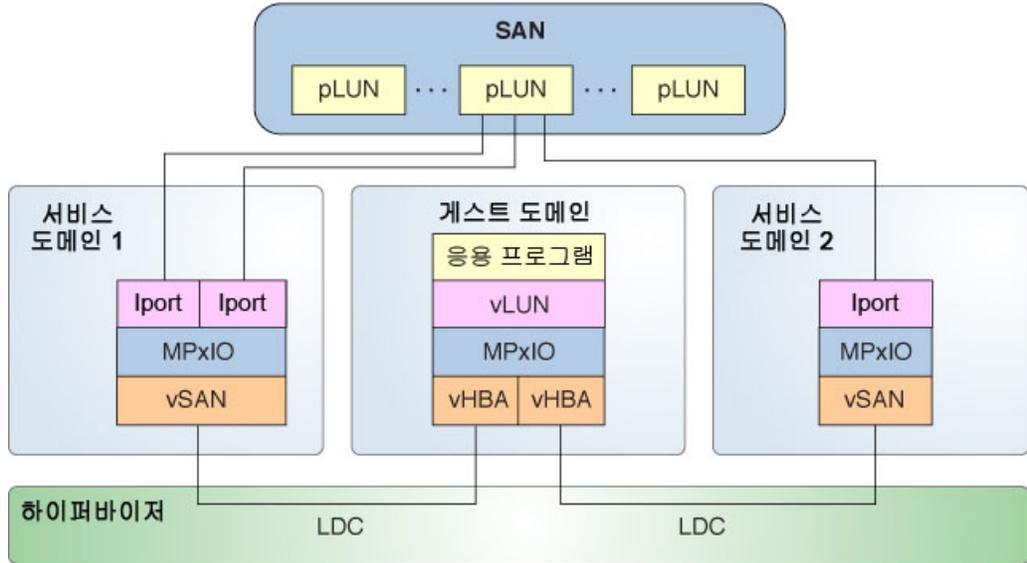


그림 10 서비스 도메인에서 가상 SCSI HBA 다중 경로 구성



▼ 가상 SCSI HBA 다중 경로를 구성하는 방법

이 절차는 그림 9에 나와 있는 가상 SCSI HBA 다중 경로 구성을 만드는 방법을 설명합니다. 이 예제는 여러 가지 가능한 다중 경로 구성 중 하나일 뿐입니다.

1. 지정된 물리적 SCSI HBA로 I/O 도메인을 만듭니다.
6장. I/O 도메인 구성을 참조하십시오.
2. 첫번째 서비스 도메인에서 첫번째 개시자 포트 경로의 가상 SAN을 내보냅니다.
`ldm add-vsan vSAN-path1 vSAN-name domain-name`
vSAN-path1은 SAN의 첫번째 개시자 포트 경로입니다.
예를 들어, 다음 명령은 svcdom1 도메인에서 vsan-mpxio1 가상 SAN을 내보냅니다.
`primary# ldm add-vsan /SYS/MB/RISER0/PCIE1/HBA0/PORT0 vsan-mpxio1 svcdom1`
3. 두번째 서비스 도메인에서 두번째 개시자 포트 경로의 가상 SAN을 내보냅니다.

```
ldm add-vsan vSAN-path2 vSAN-name domain-name
```

vSAN-path2는 SAN의 두번째 개시자 포트 경로입니다.

예를 들어, 다음 명령은 svcdom2 도메인에서 vsan-mpxio2 가상 SAN을 내보냅니다.

```
primary# ldm add-vsan /SYS/MB/RISER0/PCIE3/HBA0/PORT0 vsan-mpxio2 svcdom2
```

4. 가상 SCSI HBA를 게스트 도메인으로 내보냅니다.

```
ldm add-vhba vHBA-name vSAN-name domain-name
```

예를 들어, 다음 명령은 vhba-mpxio1 및 vhba-mpxio2 가상 SCSI HBA를 gdom 도메인으로 내보냅니다.

```
primary# ldm add-vhba vhba-mpxio1 vsan-mpxio1 gdom
primary# ldm add-vhba vhba-mpxio2 vsan-mpxio2 gdom
```

5. 게스트 도메인에서 가상 SCSI HBA의 timeout 등록 정보 값을 지정합니다.

```
ldm set-vhba timeout=seconds vHBA-name domain-name
```

예를 들어, 다음 명령은 gdom 도메인에서 vsan-mpxio1 및 vsan-mpxio2 가상 SCSI HBA의 timeout 등록 정보 값을 30으로 설정합니다.

```
primary# ldm set-vhba timeout=30 vhba-mpxio1 gdom
primary# ldm set-vhba timeout=30 vhba-mpxio2 gdom
```

6. 게스트 도메인을 재부트합니다.

▼ 게스트 도메인에서 가상 SCSI HBA에 대한 다중 경로를 사용으로 설정하는 방법

1. /platform/sun4v/kernel/drv/vhba.conf 파일을 /etc/driver/drv 디렉토리로 복사합니다.

```
# cp /platform/sun4v/kernel/drv/vhba.conf /etc/driver/drv
```

2. mpxio-disable 등록 정보를 no로 설정하도록 /etc/driver/drv/vhba.conf 파일을 편집하여 Oracle Solaris I/O 다중 경로를 사용으로 설정합니다.

3. 게스트 도메인을 재부트합니다.

▼ 게스트 도메인에서 가상 SCSI HBA에 대해 다중 경로를 사용 안함으로 설정하는 방법

1. /platform/sun4v/kernel/drv/vhba.conf 파일을 /etc/driver/drv 디렉토리로 복사합니다.

```
# cp /platform/sun4v/kernel/drv/vhba.conf /etc/driver/drv
```

2. `mpxio-disable` 등록 정보를 `yes`로 설정하도록 `/etc/driver/drv/vhba.conf` 파일을 편집하여 **Oracle Solaris I/O 다중 경로를 사용 안함으로 설정합니다.**
3. 게스트 도메인을 재부트합니다.

▼ 서비스 도메인에서 가상 SCSI HBA에 대한 다중 경로를 사용으로 설정하는 방법

1. 서비스 도메인의 모든 개시자 포트에 대해 **Oracle Solaris I/O** 경로 지정을 사용으로 설정합니다.

```
svcdom# stmsboot -e
```

자세한 내용은 [Oracle Solaris SAN Configuration and Multipathing Guide](#) 의 “Enabling and Disabling Multipathing”을 참조하십시오.

2. 서비스 도메인의 각 개시자 포트에서 연결할 수 있는 **SCSI** 장치를 나열합니다.

예를 들어 `ldm list-hba` 명령으로 [그림 10](#)에 나와 있는 것과 같이 서비스 도메인 1에 대한 다음 정보가 표시될 수 있습니다.

```
primary# ldm list-hba -d svcdom
DOMAIN
svcdom

IPORT                                VSAN
----                                -
/SYS/MB/PCIE0/HBA0/PORT1
  c0t600110D00021150101090001061ADB4d0
  c0t600110D0002115010109000146489D34d0
/SYS/MB/PCIE1/HBA0/PORT1
  c0t600110D00021150101090001061ADB4d0
  c0t600110D0002115010109000146489D34d0
```

3. 특정 개시자 포트를 참조하는 가상 **SAN** 인스턴스를 만듭니다.

다음 명령에서 개시자 포트는 `PCIE0`에서도 참조하는 SCSI 장치 두 개를 참조합니다. 동일한 LUN에 대한 물리적 경로가 있는 개시자 포트 중 하나를 `ldm add-vsant` 명령 중 하나로 사용하여 다중 경로가 사용으로 설정된 경우 가상 SAN을 구성할 수 있습니다.

```
primary# ldm add-vsant /SYS/MB/PCIE1/HBA0/PORT1 my_mpxio_vsant svcdom
```

4. 가상 **SCSI HBA**에서 게스트 도메인에 가상 **SAN**을 추가합니다.

```
primary# ldm add-vsant my_vhba my_mpxio_vsant gdom
```

5. 서비스 도메인에서 `format` 명령을 실행하여 물리적 장치를 봅니다.

다음 출력에는 각각 하나 이상의 경로가 있을 수 있는 물리적 SCSI 장치 두 개가 나와 있습니다.

```
svcdom# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c0t600110D00021150101090001061ADBF4d0 <SANBlaze-VLUN P0T1L7-V7.3-1.00GB>
     /scsi_vhci/ssd@g600110d00021150101090001061adbf4
  1. c0t600110D0002115010109000146489D34d0 <SANBlaze-VLUN P0T1L6-V7.3-1.00GB>
     /scsi_vhci/ssd@g600110d0002115010109000146489d34
  2. cid0 <SUN-DiskImage-10GB cyl 282 alt 2 hd 96 sec 768>
     /virtual-devices@100/channel-devices@200/disk@0
Specify disk (enter its number):
```

이 명령은 서비스 도메인 구성에 각 물리적 장치에 대해 경로 두 개가 있음을 보여 줍니다.

```
svcdom# mpathadm list lu
/dev/rdisk/c0t600110D00021150101090001061ADBF4d0s2
    Total Path Count: 2
    Operational Path Count: 2
/dev/rdisk/c0t600110D0002115010109000146489D34d0s2
    Total Path Count: 2
    Operational Path Count: 2
```

Oracle Solaris I/O 다중 경로 구현이 게스트 도메인과 서비스 도메인에서 실행되기 때문에 게스트 도메인의 format 출력은 실제로 동일합니다. Oracle Solaris I/O 다중 경로도 다음 출력의 g600110d0002115010109000146489d34와 같은 논리적 장치의 전체 번호를 사용하는 장치 경로를 만듭니다.

```
gdom# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c0t600110D0002115010109000146489D34d0 <SANBlaze-VLUN P0T1L6-V7.3-1.00GB>
     /scsi_vhci/disk@g600110d0002115010109000146489d34
  1. c0t600110D00021150101090001061ADBF4d0 <SANBlaze-VLUN P0T1L7-V7.3-1.00GB>
     /scsi_vhci/disk@g600110d00021150101090001061adbf4
  2. cid0 <SUN-DiskImage-10GB cyl 282 alt 2 hd 96 sec 768>
     /virtual-devices@100/channel-devices@200/disk@0
Specify disk (enter its number):
```

▼ 서비스 도메인에서 가상 SCSI HBA에 대한 다중 경로를 사용 안함으로 설정하는 방법

- 서비스 도메인의 모든 개시자 포트에 대해 Oracle Solaris I/O 다중 경로를 사용 안함으로 설정합니다.

```
svcdom# stmsboot -d
```

서비스 도메인의 개시자 포트 장치별로 Oracle Solaris I/O 다중 경로를 사용 또는 사용 안함으로 설정할 수도 있습니다. 자세한 내용은 [Oracle Solaris SAN Configuration and Multipathing Guide](#)의 “Enabling or Disabling Multipathing on a Per-Port Basis”를 참조하십시오.

SCSI 장치에서 부트

다음 절에서는 SCSI 장치에서 부트하는 방법을 설명합니다.

- “가상 LUN에서 부트” [212]
- “SCSI DVD 장치에서 부트” [213]

가상 LUN에서 부트

연관된 물리적 LUN이 OBP로 부트할 수 있는 SCSI 장치 유형(예: CD, DVD, 디스크)을 참조하는 경우 해당 가상 LUN을 부트할 수 있습니다.

주 - UltraSPARC T2, UltraSPARC T2 Plus 및 SPARC T3 서버의 시스템 펌웨어는 가상 SCSI HBA에서 부트를 지원하지 않습니다.

OpenBoot PROM 프롬프트에서 `boot` 명령을 실행하기 전에 `probe-scsi-all` 명령을 실행하여 게스트 도메인의 가상 SCSI HBA와 연관된 가상 LUN을 찾습니다.

다음 주석 예제는 출력에서 관련 부분을 강조 표시합니다.

```
{0} ok probe-scsi-all

/virtual-devices@100/channel-devices@200/scsi@0           Line 1
vHBA
TPORT-PHYS: w200200110d214900                             Line 2
  LUN: 1   Disk   VLUN           2097152 Blocks, 1073 MB
  LUN: 0   Disk   VLUN           32768000 Blocks, 16 GB   Line 3
```

이 예제의 `probe-scsi-all` 출력은 하나의 가상 SCSI HBA 인스턴스(`scsi@0`)에 `disk` 유형의 LUN 두 개가 있음을 보여줍니다.

특정 가상 LUN에서 부트하려면 `boot` 명령에 전달할 장치 경로를 수동으로 구성합니다. 장치 경로는 다음 구문을 따릅니다.

```
vhba-device-path/disk@target-port,lun:slice
```

Line 3의 LUN에서 부트하려면 다음과 같이 장치 경로를 구성해야 합니다.

- Line 2에서 `target-port` 값을 가져옵니다.
- Line 1에서 `vhba-device-path` 값을 가져옵니다.

다음은 결과 장치 경로입니다.

```
/virtual-devices@100/channel-devices@200/scsi@0/disk@w200200110d214900,0
```

이 장치 경로를 다음과 같이 OBP `boot` 명령에 전달할 수 있습니다.

```
{0} ok boot /virtual-devices@100/channel-devices@200/scsi@0/disk@w200200110d214900,0
```

SCSI DVD 장치에서 부트

SCSI DVD(디지털 다기능 디스크) 드라이브에서 부트하여 해당 DVD에서 게스트 도메인을 설치할 수 있습니다.

다음 예제는 `primary` 도메인에 연결된 SCSI DVD 장치가 있는 가상 SCSI HBA 구성을 보여줍니다.

```
primary# ldm list-hba -t -d primary
IPORT                                VSAN
----                                ----
[...]
/SYS/MB/SASHBA1/HBA0/PORT40
    init-port w50800200008f4329
    Transport Protocol SAS
    c2t3d0s0
        lun 0
        removable media 1

primary# ldm add-vsan /SYS/MB/SASHBA1/HBA0/PORT40 dvd_vsan primary
/SYS/MB/SASHBA1/HBA0/PORT40 resolved to device: /pci@400/pci@2/pci@0/pci@4/scsi@0/ipro@40
primary# ldm add-vhba dvd_vhba dvd_vsan gdom
```

`gdom` 도메인 콘솔에서 SCSI 장치를 프로브하고 DVD에서 부트합니다.

```
{0} ok probe-scsi-all
/virtual-devices@100/channel-devices@200/scsi@0

vHBA

TPORT-PHYS: p3
  LUN: 0 Removable Read Only device   TEAC   DV-W28SS-V 1.0B

{0} ok boot /virtual-devices@100/channel-devices@200/scsi@0/disk@p3
...
```

가상 LUN 설치

연관된 물리적 LUN이 설치 프로그램에서 지원되는 유형의 SCSI 장치를 참조하는 경우 해당 가상 LUN에 OS를 설치할 수 있습니다. 그러면 지정된 가상 LUN에서 부트할 수 있습니다.

가상 SCSI HBA 시간 초과

기본적으로 가상 SAN에 대한 액세스를 제공하는 서비스 도메인을 사용할 수 없는 경우 게스트 도메인에서 해당 가상 SCSI HBA로의 모든 I/O가 차단됩니다. 서비스 도메인이 작동하고 가상 SAN에 서비스를 복원하면 I/O는 자동으로 재개됩니다.

때때로 서비스 도메인을 오랫동안 사용할 수 없는 경우 파일 시스템이나 응용 프로그램에서 I/O 작업을 실패하고 오류를 보고할 수 있습니다. 각 가상 SCSI HBA에 대한 연결 시간 초과 기간을 설정하여 게스트 도메인의 가상 SCSI HBA와 서비스 도메인의 가상 SAN 간에 연결을 설정할 수 있습니다. 시간 초과 기간에 도달하면 서비스 도메인을 사용할 수 없고 가상 SCSI HBA와 가상 SAN 간에 연결이 다시 설정되지 않는 동안 보류 중인 모든 I/O 및 새로운 I/O 작업을 실패합니다.

그 밖에 시간 초과 값을 지정하려는 상황은 다음과 같습니다.

- Oracle Solaris I/O 다중 경로가 구성된 다른 경로로 페일오버되도록 하려면 관련된 각 가상 SCSI HBA에 대한 시간 초과를 설정해야 합니다.
- 라이브 마이그레이션을 수행하는 경우 마이그레이션할 게스트 도메인의 가상 SCSI HBA에 대해 `timeout` 등록 정보 값을 0으로 설정합니다. 마이그레이션을 완료한 후 각 가상 SCSI HBA에 대해 시간 초과 등록 정보를 원래 설정으로 되돌립니다.

시간 초과 값을 설정하는 방법을 알아보려면 “[가상 SCSI HBA 시간 초과 옵션 설정](#)” [203]을 참조하십시오.

가상 SCSI HBA 및 SCSI

`vhba` 모듈은 가상 SAN의 SCSI 개시자 포트와 연관된 물리적 SCSI HBA 드라이버로 SCSI 명령을 프록시합니다.

Oracle Solaris I/O 다중 경로를 구현하는 `scsi_vhci` 드라이버는 SCSI-2 예약과 SCSI-3 예약에 대한 경로 페일오버 동안 예약 지속성을 처리합니다. `vhba` 모듈은 Oracle Solaris I/O 프레임워크에 플러그인해서 `scsi_vhci` 지원을 활용하여 SCSI 예약을 지원합니다.

게스트 도메인에서 단편화가 심한 I/O 버퍼 지원

다른 `sun4v` 가상 장치와 마찬가지로, `vhba` 모듈은 소프트웨어 스택에서 상위 계층에 만들어진 I/O 버퍼에서 작동합니다. I/O 버퍼에 물리적 메모리 단편이 너무 많이 누적되면 `vhba` 모듈은 I/O 요청을 처리할 때 다음과 같은 치명적 경고 메시지를 실행합니다.

```
WARNING: ... ldc_mem_bind_hdl: ncookies(max, actual) = (8, 9)
```

각 물리적 메모리 단편은 하나의 쿠키와 연관됩니다. 실제 쿠키 수가 최대 쿠키 수를 초과할 경우 I/O 요청을 실패합니다.

오류 메시지는 필요한 실제 쿠키 수를 보여줍니다. 오류를 없애려면 `/etc/system` 파일에서 `vhba_desc_ncookies` 값을 변경합니다. I/O 버퍼당 사용할 쿠키 수가 최소한 실제 값이 되도록 지정합니다. 또한 허용 가능한 최대 쿠키 수를 지정하는 `vhba_desc_max_ncookies` 등록 정보 값을 늘립니다.

`/etc/system` 등록 정보 값을 정확히 만들거나 업데이트하는 방법은 “[/etc/system 파일에서 등록 정보 값 업데이트](#)” [352]를 참조하십시오.

그런 다음 `ldm remove-vhba`와 `ldm add-vhba` 명령을 연속 실행하거나 게스트 도메인을 재부트하여 가상 SCSI HBA 연결을 다시 만듭니다.

예를 들어, `vhba_desc_max_ncookies` 등록 정보 값을 12로 설정하려면 다음 행을 `/etc/system` 파일에 추가합니다.

```
set vhba:vhba_desc_ncookies = 12
```


◆◆◆ 13 장

가상 네트워크 사용

이 장은 Oracle VM Server for SPARC 소프트웨어로 가상 네트워크를 사용하는 방법에 대해 설명하며, 다음과 같은 항목으로 구성됩니다.

- “가상 네트워크에 대한 소개” [218]
- “Oracle Solaris 11 네트워킹 개요” [218]
- “Oracle Solaris 10 네트워킹 개요” [221]
- “가상 네트워크 성능 최대화” [222]
- “가상 스위치” [223]
- “가상 네트워크 장치” [225]
- “네트워크 장치 구성 및 통계 보기” [228]
- “가상 네트워크 장치에서 소비되는 물리적 네트워크 대역폭의 양 제어” [231]
- “가상 장치 식별자 및 네트워크 인터페이스 이름” [234]
- “자동 또는 수동 MAC 주소 지정” [237]
- “Oracle Solaris 10을 실행하는 도메인에서 네트워크 어댑터 사용” [239]
- “NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성” [240]
- “Oracle VM Server for SPARC 환경에서 IPMP 구성” [243]
- “VLAN 태그 지정 사용” [252]
- “전용 VLAN 사용” [256]
- “패킷 처리량 성능 조정” [261]
- “가상 스위치에서 링크 통합 사용” [262]
- “정보 프레임 구성” [264]
- “가상 네트워크에서 가상 NIC 사용” [268]
- “신뢰할 수 있는 가상 네트워크 사용” [271]
- “Oracle Solaris 11 네트워킹 관련 기능 차이점” [277]

Oracle Solaris OS 네트워킹은 Oracle Solaris 10 OS와 Oracle Solaris 11 OS 간에 상당히 다릅니다. 고려할 문제에 대한 자세한 내용은 “Oracle Solaris 10 네트워킹 개요” [221], “Oracle Solaris 11 네트워킹 개요” [218] 및 “Oracle Solaris 11 네트워킹 관련 기능 차이점” [277]을 참조하십시오.

가상 네트워크에 대한 소개

가상 네트워크는 외부 물리적 네트워크를 사용하지 않고도 도메인이 서로 통신할 수 있도록 해 줍니다. 또한 도메인이 동일한 물리적 네트워크 인터페이스를 사용하여 물리적 네트워크에 액세스하고 원격 시스템과 통신할 수 있도록 해줍니다. 가상 네트워크 장치를 연결할 수 있는 가상 스위치를 생성하면 가상 네트워크가 만들어집니다.

Oracle Solaris 네트워킹은 Oracle Solaris 10 OS와 Oracle Solaris 11 OS 간에 상당히 다릅니다. 다음 두 단원에서는 OS별 네트워킹에 대한 개요 정보를 다룹니다.

주 - Oracle Solaris 10 네트워킹은 도메인 또는 시스템에서와 동일하게 작동합니다. Oracle Solaris 11 네트워킹의 경우도 마찬가지입니다. Oracle Solaris OS 네트워킹에 대한 자세한 내용은 [Oracle Solaris 10 Documentation](#) 및 [Oracle Solaris 11.3 Documentation](#)을 참조하십시오.

Oracle Solaris 10 및 Oracle Solaris 11 네트워킹 간 기능 차이점은 "[Oracle Solaris 11 네트워킹 개요](#)" [218]에 설명되어 있습니다.

Oracle Solaris 11 네트워킹 개요

Oracle Solaris 11 OS에서는 [Oracle Solaris 11.3 Documentation](#)의 Oracle Solaris 11 네트워크 설명서에 설명된 여러 가지 새로운 네트워킹 기능이 도입되었습니다.

Oracle VM Server for SPARC 소프트웨어를 사용할 때 다음과 같은 Oracle Solaris 11 네트워크 기능을 잘 알고 있는 것이 중요합니다.

- 모든 네트워크 구성은 `ipadm` 및 `dladm` 명령으로 수행됩니다.
- “기본 베니티 이름” 기능은 모든 물리적 네트워크 어댑터에 대해 일반 링크 이름(예: `net0`)을 생성합니다. 이 기능은 또한 OS에서 물리적 네트워크 어댑터처럼 인식되는 가상 스위치(`vswn`) 및 가상 네트워크 장치(`vnetn`)에 대해서도 일반 이름을 생성합니다. 물리적 네트워크 장치와 연관된 일반 링크 이름을 식별하려면 `dladm show-phys` 명령을 사용합니다.

기본적으로 Oracle Solaris 11에서 물리적 네트워크 장치 이름은 일반 “베니티” 이름을 사용합니다. Oracle Solaris 10에서 사용되던 장치 드라이버 이름(예: `nxge0`) 대신 일반 이름(예: `net0`)이 사용됩니다.

다음 명령은 드라이버 이름(예: `nxge0`) 대신 일반 이름(예: `net0`)을 지정하여 `primary` 도메인에 대한 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

- Oracle Solaris 11 OS에서는 VNIC(가상 네트워크 인터페이스 카드)를 사용하여 내부 가상 네트워크를 만듭니다.

VNIC는 물리적 네트워크 장치에서 만들어 영역에 지정할 수 있는 물리적 네트워크 장치에 대한 가상 인스턴스화입니다.

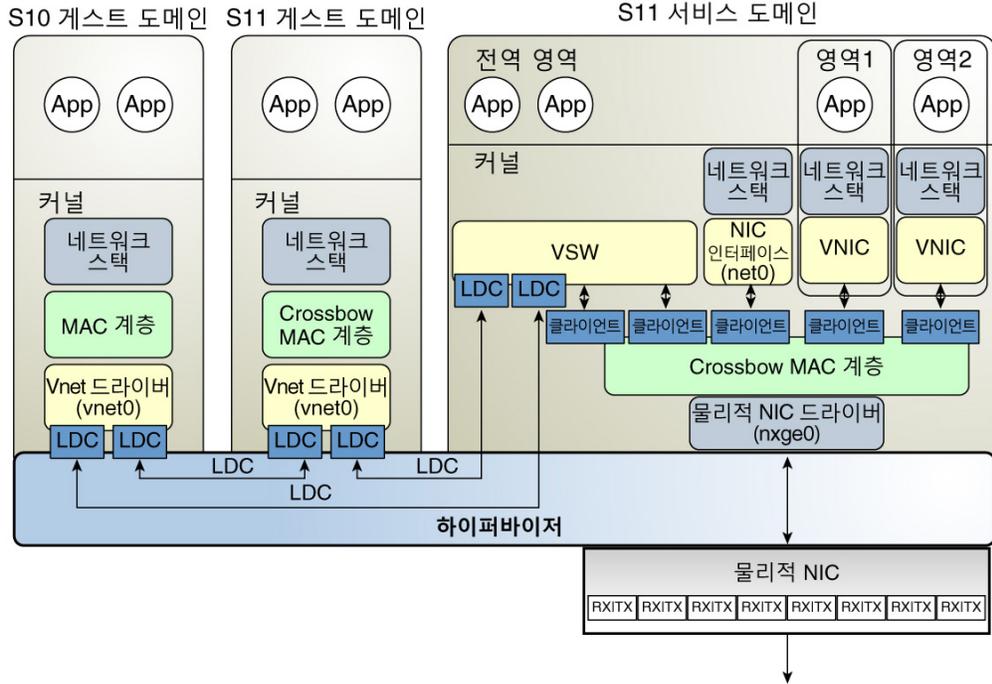
- Oracle VM Server for SPARC 소프트웨어를 구성할 경우 Oracle Solaris 11 `DefaultFixed` NCP(네트워크 구성 프로파일)를 사용합니다.
Oracle Solaris 11 도메인의 경우, `DefaultFixed` NCP를 사용합니다. 설치 중이나 설치 후에 이 프로파일을 사용하여 설정할 수 있습니다. Oracle Solaris 11 설치 중 수동 네트워크 구성을 선택합니다.
- 기본 네트워크 인터페이스를 가상 스위치(`vsw`) 인터페이스로 바꾸지 마십시오. 서비스 도메인이 기존의 기본 네트워크 인터페이스를 사용하여 동일한 가상 스위치에 가상 네트워크 장치가 연결되어 있는 게스트 도메인과 통신할 수 있습니다.
- 가상 스위치에 물리적 어댑터의 MAC 주소를 사용할 경우 기본 네트워크 인터페이스와 충돌하므로 가상 스위치에 물리적 네트워크 어댑터의 MAC 주소를 사용하지 마십시오.

주 - 이 릴리스에서는 `dladm` 또는 `ipadm` 명령을 사용하여 `DefaultFixed` NCP로 Oracle Solaris 11 시스템에서 데이터 링크와 네트워크 인터페이스를 구성하십시오.

`netadm list` 명령을 사용하여 `DefaultFixed` NCP가 사용으로 설정되어 있는지 확인하십시오. [Oracle Solaris Administration: Network Interfaces and Network Virtualization](#) 의 7 장, "Using Datalink and Interface Configuration Commands on Profiles"을 참조하십시오.

다음 다이어그램은 Oracle Solaris 10 OS를 실행하는 게스트 도메인이 Oracle Solaris 11 서비스 도메인과 완전하게 호환됨을 보여줍니다. 단, Oracle Solaris 11 OS에서 기능이 추가되었거나 향상되었다는 점이 다릅니다.

그림 11 Oracle Solaris 11 OS에 대한 Oracle VM Server for SPARC 네트워크 개요



다이어그램은 Oracle Solaris 11 도메인에서 네트워크 장치 이름(예: `nxge0` 및 `vnet0`)이 일반 링크 이름(예: `netn`)으로 표시될 수 있음을 보여줍니다. 또한 다음 사항에 유의하십시오.

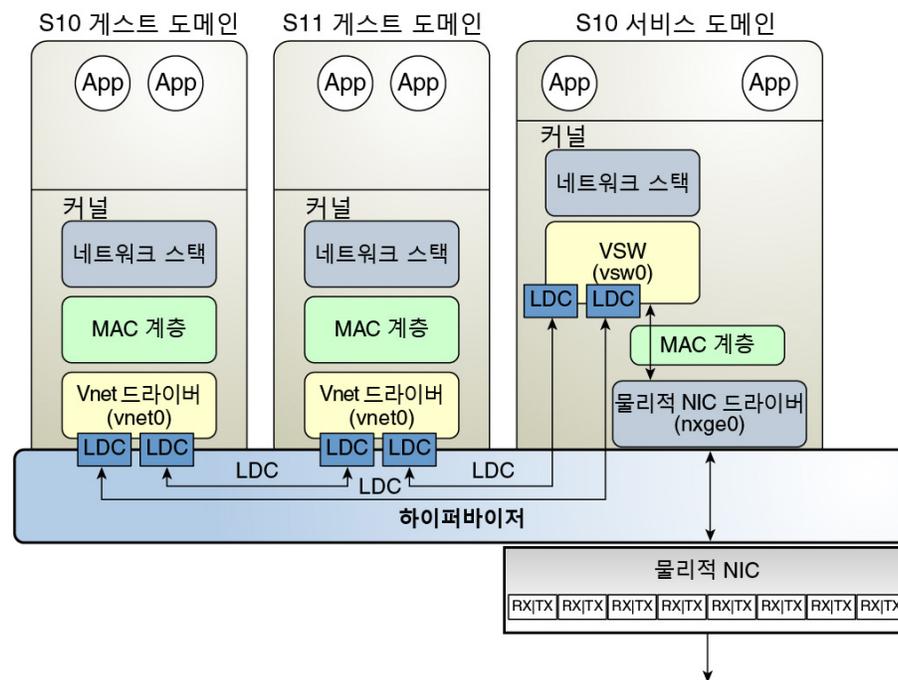
- 서비스 도메인의 가상 스위치는 게스트 도메인이 서로 통신할 수 있도록 게스트 도메인에 연결됩니다.
- 가상 스위치는 게스트 도메인이 물리적 네트워크와 통신할 수 있도록 물리적 네트워크 장치 `nxge0`에도 연결됩니다.
가상 스위치는 또한 게스트 도메인을 통해 서비스 도메인 네트워크 인터페이스 `net0` 및 `nxge0`과 동일한 물리적 네트워크 장치에 있는 VNIC와 통신할 수 있습니다. 여기에는 게스트 도메인과 Oracle Solaris 11 서비스 도메인 간의 통신이 포함됩니다. 가상 스위치 자체 (`vsw` 장치)를 네트워크 장치로 구성하지 마십시오. 이 기능은 Oracle Solaris 11에서 제거되었으며 더 이상 지원되지 않습니다.
- Oracle Solaris 10 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `ifconfig` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.
- Oracle Solaris 11 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `net0`과 같은 일반 링크 이름으로 표시될 수 있습니다. 또한 `ipadm` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.

가상 스위치는 정규 물리적 네트워크 스위치처럼 작동하며, 연결되어 있는 여러 시스템 간에 네트워크 패킷을 전환합니다. 시스템은 게스트 도메인, 서비스 도메인 또는 물리적 네트워크 중 하나일 수 있습니다.

Oracle Solaris 10 네트워크 개요

다음 다이어그램은 Oracle Solaris 11 OS를 실행하는 게스트 도메인이 Oracle Solaris 10 서비스 도메인과 완전하게 호환됨을 보여줍니다. 단, Oracle Solaris 11 OS에서 기능이 추가되었거나 향상되었다는 점이 다릅니다.

그림 12 Oracle Solaris 10 OS에 대한 Oracle VM Server for SPARC 네트워크 개요



이전 다이어그램은 Oracle Solaris 10 OS에만 적용되는 인터페이스 이름(예: nxge0, vsw0 및 vnet0)을 보여줍니다. 또한 다음 사항에 유의하십시오.

- 서비스 도메인의 가상 스위치는 게스트 도메인이 서로 통신할 수 있도록 게스트 도메인에 연결됩니다.

- 가상 스위치는 또한 게스트 도메인이 물리적 네트워크와 통신할 수 있도록 물리적 네트워크 인터페이스 `nxge0`에도 연결됩니다.
- 가상 스위치 네트워크 인터페이스 `vsw0`은 두 게스트 도메인이 서비스 도메인과 통신할 수 있도록 서비스 도메인에 만들어집니다.
- 서비스 도메인의 가상 스위치 네트워크 인터페이스 `vsw0`은 Oracle Solaris 10 `ifconfig` 명령을 사용하여 구성할 수 있습니다.
- Oracle Solaris 10 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `ifconfig` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.
- Oracle Solaris 11 게스트 도메인의 가상 네트워크 장치 `vnet0`은 `net0`과 같은 일반 링크 이름으로 표시될 수 있습니다. 또한 `ipadm` 명령을 사용하여 네트워크 인터페이스로 구성할 수 있습니다.

가상 스위치는 정규 물리적 네트워크 스위치처럼 작동하며, 게스트 도메인, 서비스 도메인, 물리적 네트워크 등 연결되어 있는 여러 시스템 간에 네트워크 패킷을 전환합니다. `vsw` 드라이버는 가상 스위치를 네트워크 인터페이스로 구성할 수 있도록 해주는 네트워크 장치 기능을 제공합니다.

가상 네트워크 성능 최대화

이 절에 설명된 대로 플랫폼과 도메인을 구성하면 게스트와 외부 네트워크 그리고 게스트 간 통신을 위한 높은 전송 속도를 달성할 수 있습니다. 가상 네트워크 스택은 점보 프레임을 사용할 필요 없이 높은 TCP 성능을 제공하는 LSO(large segment offload) 지원을 시작합니다.

하드웨어 및 소프트웨어 요구사항

도메인에 대한 네트워크 성능을 최대화하려면 다음 요구사항을 충족해야 합니다.

- 하드웨어 요구사항. 이러한 성능 향상은 SPARC T4 서버부터 제공됩니다.
- 시스템 펌웨어 요구사항. 이 SPARC 시스템에서 최신 시스템 펌웨어를 실행해야 합니다. [Oracle VM Server for SPARC 3.4 설치 설명서](#)의 “정규화된 시스템 펌웨어 버전”을 참조하십시오.
- Oracle Solaris OS 요구사항. 서비스 도메인 및 게스트 도메인이 다음 Oracle Solaris OS 버전을 실행하는지 확인하십시오.

주 - 정규화된 Oracle Solaris OS 버전을 실행하면 새로운 기능에 액세스할 수 있습니다. [Oracle VM Server for SPARC 3.4 설치 설명서](#)의 “정규화된 Oracle Solaris OS 버전”을 참조하십시오.

- 서비스 도메인. 최소 Oracle Solaris 11.1 SRU 9 OS 또는 Oracle Solaris 10 OS와 150031-03 패치.
- 게스트 도메인. 최소 Oracle Solaris 11.1 SRU 9 OS 또는 Oracle Solaris 10 OS와 150031-03 패치.
- CPU 및 메모리 요구사항. 서비스 도메인과 게스트 도메인에 충분한 CPU 및 메모리 리소스를 지정하십시오.
 - 서비스 도메인. 서비스 도메인은 게스트 도메인에 대한 데이터 프록시로 작동하므로 최소 2 CPU 코어 및 최소 16GB의 메모리를 서비스 도메인에 지정하십시오.
 - 게스트 도메인. 각 게스트 도메인이 최소 10Gbps 성능을 발휘할 수 있도록 구성하십시오. 최소 2 CPU 코어 및 최소 4GB의 메모리를 각 게스트 도메인에 지정하십시오.

가상 네트워크의 성능 최대화를 위한 도메인 구성

Oracle VM Server for SPARC 및 Oracle Solaris OS의 이전 버전에서는 점보 프레임 구성하여 네트워크 성능을 향상시킬 수 있었습니다. 이 구성은 더 이상 필요하지 않으며 다른 이유로 필요하지 않다면 서비스 및 게스트 도메인에 대해 표준 MTU 값인 1500이 가장 좋습니다.

향상된 네트워킹 성능을 달성하려면 서비스 도메인 및 게스트 도메인에 대해 `extended-mapin-space` 등록 정보를 `on`으로 설정합니다. 이 옵션이 기본 동작입니다.

```
primary# ldm set-domain extended-mapin-space=on domain-name
```

`extended-mapin-space` 등록 정보 값을 확인하려면 다음 명령을 실행합니다.

```
primary# ldm list -l domain-name |grep extended-mapin
extended-mapin-space=on
```

주 - `extended-mapin-space` 등록 정보 값을 변경하면 `primary` 도메인에 대한 지연된 재구성이 트리거됩니다. 이 상황에서는 `primary` 도메인을 재부트해야 합니다. 또한 이 등록 정보 값을 변경하기 전에 먼저 게스트 도메인을 중지해야 합니다.

가상 스위치

가상 스위치(`vsw`)는 서비스 도메인에서 실행되고 가상 스위치 드라이버에 의해 관리되는 구성 요소로, 일부 게스트 도메인에 연결되어 게스트 도메인 간 네트워크 통신이 가능하도록 해줍니다. 또한 가상 스위치가 물리적 네트워크 인터페이스와도 연관된 경우 물리적 네트워크 인터페이스를 통해 게스트 도메인과 물리적 네트워크 간의 네트워크 통신이 허용됩니다. Oracle Solaris 10 서비스 도메인에서 실행할 때 가상 스위치에는 또한 서비스 도메인이 가상 스위치에 연결된 다른 도메인과 통신할 수 있도록 해주는 네트워크 인터페이스 `vswn`이 있습니다. 가상

스위치는 정규 네트워크 인터페이스처럼 사용할 수 있으며, Oracle Solaris 10 `ifconfig` 명령으로 구성할 수 있습니다.

가상 네트워크 장치를 도메인에 지정하면 가상 스위치를 제공하는 도메인에 암시적 종속성이 생깁니다. `ldm list-dependencies` 명령을 사용하여 해당 종속성을 보거나 이 가상 스위치에 종속된 도메인을 볼 수 있습니다. [“도메인 I/O 종속성 나열” \[371\]](#)을 참조하십시오.

Oracle Solaris 11 서비스 도메인에서 가상 스위치를 정규 네트워크 인터페이스로 사용할 수 없습니다. 가상 스위치가 물리적 네트워크 인터페이스에 연결된 경우 이 물리적 인터페이스를 사용하여 서비스 도메인과 통신할 수 있습니다. 물리적 인터페이스 없이 구성된 경우 `etherstub`를 네트워크 장치(`net-dev`)로 사용하여 VNIC와 연결하면 서비스 도메인과 통신할 수 있습니다.

가상 스위치에 대한 백엔드 장치로 사용할 네트워크 장치를 확인하려면 `dladm show-phys` 출력에서 물리적 네트워크 장치를 검색하거나 `ldm list-netdev` 명령을 사용하여 논리적 도메인의 네트워크 장치를 나열합니다.

주 - Oracle Solaris 10 서비스 도메인에 가상 스위치를 추가하면 네트워크 인터페이스가 생성되지 않습니다. 따라서 기본적으로 서비스 도메인은 가상 스위치에 연결된 게스트 도메인과 통신할 수 없습니다. 게스트 도메인과 서비스 도메인 간 네트워크 통신이 가능하게 하려면 연관된 가상 스위치의 네트워크 인터페이스를 서비스 도메인에 생성하고 구성해야 합니다. 지침은 [“Oracle Solaris 10 서비스 도메인과 다른 도메인 사이의 네트워킹 사용으로 설정” \[38\]](#)을 참조하십시오.

이러한 상황은 Oracle Solaris 10 OS의 경우에만 발생하며 Oracle Solaris 11 OS의 경우에는 발생하지 않습니다.

`ldm add-vsw`, `ldm set-vsw` 및 `ldm rm-vsw` 명령을 각각 사용하여 도메인에 가상 스위치를 추가하고, 가상 스위치 옵션을 설정하며, 가상 스위치를 제거할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

NIC 또는 통합의 VLAN 태그 지정 인스턴스에 가상 스위치를 만들 때 `ldm add-vsw` 또는 `ldm set-vsw` 명령을 사용하는 경우 NIC(`nxge0`), 통합(`aggr3`) 또는 베니티 이름(`net0`)을 `net-dev` 등록 정보의 값으로 지정해야 합니다.

주 - Oracle Solaris 11.2 SRU 1 OS부터 `ldm set-vsw` 명령을 사용하여 `net-dev` 등록 정보 값을 동적으로 업데이트할 수 있습니다. 이전 Oracle Solaris OS 릴리스에서는 `primary` 도메인에서 `ldm set-vsw` 명령을 사용하여 `net-dev` 등록 정보 값을 업데이트하면 `primary` 도메인이 지연된 재구성 모드로 진입했습니다.

IPoIB(InfiniBand IP-over-InfiniBand) 네트워크 장치 위에 가상 스위치를 추가할 수 없습니다. `ldm add-vsw` 및 `ldm add-vnet` 명령이 성공하는 것처럼 보이더라도 이 장치는 InfiniBand 전송 계층을 통해 IP 패킷을 전송하므로 데이터 플로우가 발생하지 않습니다. 가상 스위치는 전송 계층으로 이더넷만 지원합니다. IPoIB 및 EoIB(Ethernet-over-InfiniBand)는 가상 스위치에 대한 백엔드를 지원하지 않습니다.

다음 명령은 물리적 네트워크 어댑터에 net0이라는 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

다음 예제는 ldm list-netdev -b 명령을 사용하여 svcdom 서비스 도메인에 유효한 가상 스위치 백엔드 장치만 표시합니다.

```
primary# ldm list-netdev -b svcdom
DOMAIN
svcdom

NAME          CLASS MEDIA STATE   SPEED OVER  LOC
-----
net0          PHYS  ETHER up      10000 ixgbe0 /SYS/MB/RISER1/PCIE
net1          PHYS  ETHER unknown 0      ixgbe1 /SYS/MB/RISER1/PCIE4
net2          ESTUB ETHER unknown 0      --      --
net3          ESTUB ETHER unknown 0      --      --
ldoms-estub.vsw0 ESTUB ETHER unknown 0      --      --
```

가상 네트워크 장치

가상 네트워크 장치는 가상 스위치에 연결된 도메인에 정의된 가상 장치입니다. 가상 네트워크 드라이버에 의해 관리되며, LDC(논리적 도메인 채널)를 사용하여 하이퍼바이저를 통해 가상 네트워크에 연결됩니다.

주 - 게스트 도메인은 가상 네트워크 장치를 최대 999개까지 지원합니다.

가상 네트워크 장치는 vnet*n*이라는 이름의 네트워크 인터페이스로 사용할 수 있으며, 정규 네트워크 인터페이스처럼 사용되어 Oracle Solaris 10 ifconfig 명령 또는 Oracle Solaris 11 ipadm 명령으로 구성할 수 있습니다.

주 - Oracle Solaris 11에서는 장치에 일반 이름이 지정되므로, vnet*n*에는 net0과 같은 일반 이름이 사용됩니다.

ldm add-vnet, ldm set-vnet 및 ldm rm-vnet 명령을 각각 사용하여 도메인에 가상 네트워크 장치를 추가하고, 기존의 가상 네트워크 장치에 대한 옵션을 설정하며, 가상 네트워크 장치를 제거할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Oracle Solaris 10 및 Oracle Solaris 11용 Oracle VM Server for SPARC 네트워킹에 대한 정보는 [그림 12](#) 및 [그림 11](#) 각각에서 참조하십시오.

Vnet 간 LDC 채널

기본적으로 Logical Domains Manager는 다음과 같은 방식으로 LDC 채널을 지정합니다.

- 가상 네트워크 장치와 가상 스위치 장치 간에 LDC 채널을 지정합니다.

- 동일한 가상 스위치에 연결된 가상 네트워크 장치 쌍 간에 LDC 채널을 지정합니다(Vnet 간).

Vnet 간 LDC 채널은 뛰어난 게스트 간 통신 성능을 얻기 위해 가상 네트워크 장치가 직접 통신할 수 있도록 구성됩니다. 그러나 가상 스위치 장치에 있는 가상 네트워크 장치의 수가 늘어나면 Vnet 간 통신에 필요한 LDC 채널 수도 2차식으로 늘어납니다.

지정된 가상 스위치 장치에 연결된 모든 가상 네트워크 장치에 대해 Vnet 간 LDC 채널 할당을 사용 또는 사용 안함으로 설정할 수 있습니다. 이 할당을 사용 안함으로 설정하면 LDC 채널의 개수가 제한되어 소비가 줄어들 수 있습니다.

다음과 같은 경우에 이 할당을 사용 안함으로 설정하는 것이 좋습니다.

- 게스트 간 통신 성능이 중요하지 않은 경우
- 가상 스위치 장치에 많은 수의 가상 네트워크 장치가 필요한 경우

Vnet 간 채널을 지정하지 않으면 게스트 도메인에 가상 I/O 장치를 추가할 때 더 많은 LDC 채널을 사용할 수 있습니다.

주 - 게스트 간 성능이 시스템에서 가상 네트워크 장치 수를 늘리는 것보다 더 중요한 경우 Vnet 간 LDC 채널 할당을 사용 안함으로 설정하지 마십시오.

`ldm add-vsw` 및 `ldm set-vsw` 명령을 사용하여 `inter-vnet-link` 등록 정보의 `on`, `off` 또는 `auto` 값을 지정할 수 있습니다.

기본적으로 `inter-vnet-link` 등록 정보는 `auto`로 설정됩니다. 즉, 특정 가상 스위치의 가상 네트워크 수가 `ldmd/ldm/inter_vnet_link_limit` SMF 등록 정보에서 지정한 기본 최대 제한을 초과하여 증가하지 않는 한 `inter-vnet` LDC 채널이 할당됩니다. 기본 `ldmd/ldm/inter_vnet_link_limit` 값은 8입니다. 최대 가상 네트워크 수보다 많은 수가 가상 스위치에 존재하는 경우 `inter-vnet` LDC가 사용 안함으로 설정됩니다. [“논리적 도메인에 있는 네트워크 확인” \[228\]](#)을 참조하십시오.

게스트 도메인을 바인딩하거나 가상 네트워크를 바인드된 도메인에 추가하여 가상 스위치에서 가상 네트워크 수가 제한을 초과하는 경우 `inter-vnet` LDC가 자동으로 사용 안함으로 설정됩니다. 그 반대의 경우도 마찬가지입니다. 게스트 도메인을 바인딩 해제하거나 바인드된 도메인에서 가상 네트워크를 제거하여 가상 스위치의 가상 네트워크 수가 제한보다 작아지는 경우 `inter-vnet` LDC가 자동으로 사용으로 설정됩니다.

`inter-vnet-link=auto`일 때 가상 스위치에 대한 `inter-vnet` 링크의 활성 상태에 따라 `ldm list` 출력에 값이 `on/auto` 또는 `off/auto`로 표시됩니다.

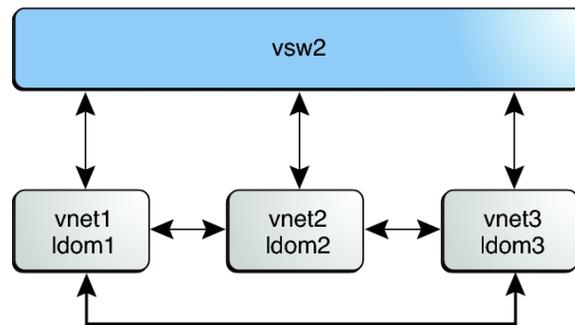
Oracle VM Server for SPARC 3.4로 시스템을 업그레이드하는 경우 `inter-vnet-link` 값이 보 존됩니다.

다음 그림에는 각각 `inter-vnet-link=on` 및 `inter-vnet-link=off`일 때 일반적인 가상 스위치를 보여 줍니다.

다음 그림은 가상 네트워크 장치를 세 개 포함하는 일반적인 가상 스위치를 보여줍니다. `inter-vnet-link` 등록 정보가 `on`으로 설정되어 있어 Vnet 간 LDC 채널이 할당되었음을 나타냅니다. `vnet1`과 `vnet2` 사이에서 이루어지는 게스트 간 통신은 가상 스위치를 거치지 않고 직접 수행됩니다.

이 그림에는 `inter-vnet-link=auto` 및 동일한 가상 스위치에 연결된 가상 네트워크 수가 `ldmd/auto_inter_vnet_link_limit` SMF 등록 정보에서 설정한 최대 값 이하인 경우도 나와 있습니다.

그림 13 Vnet 간 채널을 사용하는 가상 스위치 구성

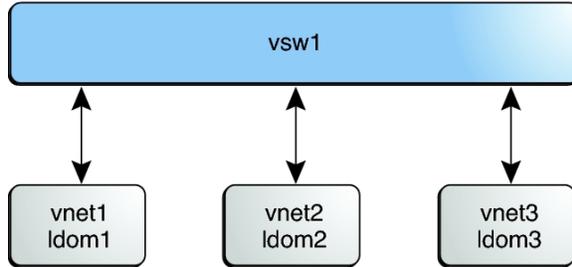


다음 그림은 `inter-vnet-link` 등록 정보가 `off`로 설정된 동일한 가상 스위치 구성을 보여줍니다. Vnet 간 LDC 채널은 할당되지 않습니다. `inter-vnet-link` 등록 정보가 `on`으로 설정된 경우보다 더 적은 수의 LDC 채널이 사용됩니다. 이 구성에서는 `vnet1`과 `vnet2` 사이에서 이루어지는 게스트 간 통신이 `vsw1`을 통해 수행되어야 합니다.

이 그림에는 `inter-vnet-link=auto` 및 동일한 가상 스위치에 연결된 가상 네트워크 수가 `ldmd/auto_inter_vnet_link_limit` SMF 등록 정보에서 설정한 최대값을 초과하는 경우도 나와 있습니다.

주 - Vnet 간 LDC 채널 지정을 사용 안함으로 설정해도 게스트 간 통신이 수행됩니다. 대신, 모든 게스트 간 통신 트래픽이 한 게스트 도메인에서 다른 게스트 도메인으로 직접 이동하지 않고 가상 스위치를 거칩니다.

그림 14 Vnet 간 채널을 사용하지 않는 가상 스위치 구성



LDC 채널에 대한 자세한 내용은 “논리적 도메인 채널 사용” [363]을 참조하십시오.

논리적 도메인에 있는 네트워크 확인

OBP(OpenBoot PROM) 프롬프트에서 명령을 실행하여 논리적 도메인에 대한 네트워크 정보를 나열할 수 있습니다.

도메인에서 사용 가능한 네트워크를 나열하려면 해당 도메인에서 `show-nets` 명령을 실행하십시오.

```
OK show-nets
```

사용 가능한 네트워크를 나열하고 네트워크 트래픽을 표시하려면 컨트롤 도메인에서 `watch-net-all` 명령을 실행하십시오.

```
OK watch-net-all
```

네트워크 장치 구성 및 통계 보기

`ldm list-netdev` 및 `ldm list-netstat` 명령으로 각각 시스템의 네트워크 장치와 네트워크 통계에 대한 정보를 볼 수 있습니다. 결과적으로, 주어진 물리적 도메인의 네트워크 장치와 통계를 중앙 집중식으로 볼 수 있습니다.

이 명령을 사용하려면 게스트 도메인에서 최소 Oracle Solaris 11.2 SRU 1 OS를 실행해야 합니다.

예 38 네트워크 장치 구성 정보 나열

다음 예제는 `ldm list-netdev` 명령을 사용하여 `ldg1` 도메인의 네트워크 장치를 간단한 목록으로 표시합니다.

```
primary# ldm list-netdev ldg1

DOMAIN
ldg1

NAME          CLASS MEDIA STATE  SPEED OVER  LOC
-----
net0          VNET  ETHER up      0      --  primary-vsw0/vne t0_ldg1
net3          PHYS  ETHER up    10000 --  /SYS/MB/RISER1/PCIE4
net4          VSW   ETHER up    10000 --  ldg1-vsw1
net1          PHYS  ETHER up    10000 --  /SYS/MB/RISER1/PCIE4
net5          VNET  ETHER up      0      --  ldg1-vsw1/vnet1_ldg1
net6          VNET  ETHER up      0      --  ldg1-vsw1/vnet2_ldg1
aggr2        AGGR  ETHER unknown 0      net1,net3 --
ldoms-vsw0.vport3 VNIC  ETHER unknown 0      --  ldg1-vsw1/vnet2_ldg1
ldoms-vsw0.vport2 VNIC  ETHER unknown 0      --  ldg1-vsw1/vnet1_ldg1
ldoms-vsw0.vport1 VNIC  ETHER unknown 0      --  ldg1-vsw1/vnet2_ldg3
ldoms-vsw0.vport0 VNIC  ETHER unknown 0      --  ldg1-vsw1/vnet2_ldg2
```

예 39 상세한 네트워크 장치 구성 정보 나열

다음 예제는 `ldm list-netdev -l` 명령을 사용하여 `ldg1` 도메인의 네트워크 장치를 상세한 목록으로 표시합니다.

```
primary# ldm list-netdev -l ldg1

-----
DOMAIN
ldg1

NAME          CLASS MEDIA STATE  SPEED OVER  LOC
-----
net0          VNET  ETHER up      0      --  primary-vsw0/vnet0_ldg1
[/virtual-devices@100/channel-devices@200/network@0]
MTU           : 1500 [1500-1500]
IPADDR        : 10.129.241.200/255.255.255.0
MAC_ADDRS     : 00:14:4f:fb:9c:df

net3          PHYS  ETHER up    10000 --  /SYS/MB/RISER1/PCIE4
[/pci@400/pci@1/pci@0/pci@0/network@0]
MTU           : 1500 [576-15500]
MAC_ADDRS     : a0:36:9f:0a:c5:d2

net4          VSW   ETHER up    10000 --  ldg1-vsw1
[/virtual-devices@100/channel-devices@200/virtual-network-switch@0]
MTU           : 1500 [1500-1500]
IPADDR        : 192.168.1.2/255.255.255.0
MAC_ADDRS     : 00:14:4f:fb:61:6e

net1          PHYS  ETHER up    10000 --  /SYS/MB/RISER1/PCIE4
[/pci@400/pci@1/pci@0/pci@0/network@0,1]
MTU           : 1500 [576-15500]
MAC_ADDRS     : a0:36:9f:0a:c5:d2

net5          VNET  ETHER up      0      --  ldg1-vsw1/vnet1_ldg1
[/virtual-devices@100/channel-devices@200/network@1]
```

```

MTU      : 1500 [1500-1500]
IPADDR   : 0.0.0.0 /255.0.0.0
          : fe80::214:4fff:fef8:5062/ffc0::
MAC_ADDRS : 00:14:4f:f8:50:62

net6          VNET   ETHER   up      0    --  ldg1-vsw1/vnet2_ldg1
[/virtual-devices@100/channel-devices@200/network@2]
MTU          : 1500 [1500-1500]
IPADDR       : 0.0.0.0 /255.0.0.0
          : fe80::214:4fff:fef8:af92/ffc0::
MAC_ADDRS    : 00:14:4f:f8:af:92

aggr2         AGGR    ETHER   unknown 0    net1,net3 --
MODE         : TRUNK
POLICY       : L2,L3
LACP_MODE    : ACTIVE
MEMBER       : net1 [PORTSTATE = attached]
MEMBER       : net3 [PORTSTATE = attached]
MAC_ADDRS    : a0:36:9f:0a:c5:d2

ldoms-vsw0.vport3 VNIC   ETHER   unknown 0    --  ldg1-vsw1/vnet2_ldg1
MTU          : 1500 [576-1500]
MAC_ADDRS    : 00:14:4f:f8:af:92

ldoms-vsw0.vport2 VNIC   ETHER   unknown 0    --  ldg1-vsw1/vnet1_ldg1
MTU          : 1500 [576-1500]
MAC_ADDRS    : 00:14:4f:f8:50:62

ldoms-vsw0.vport1 VNIC   ETHER   unknown 0    --  ldg1-vsw1/vnet2_ldg3
MTU          : 1500 [576-1500]
MAC_ADDRS    : 00:14:4f:f9:d3:88

ldoms-vsw0.vport0 VNIC   ETHER   unknown 0    --  ldg1-vsw1/vnet2_ldg2
MTU          : 1500 [576-1500]
MAC_ADDRS    : 00:14:4f:fa:47:f4
              : 00:14:4f:f9:65:b5
              : 00:14:4f:f9:60:3f
    
```

예 40 네트워크 장치 통계 나열

ldm list-netstat 명령은 시스템에 있는 하나 이상의 도메인에 대한 네트워크 통계를 보여줍니다.

다음 예제는 시스템의 모든 도메인에 대한 기본 네트워크 통계를 보여줍니다.

```

primary# ldm list-netstat

DOMAIN
primary

NAME          IPACKETS      RBYTES      OPACKETS      OBYTES
----          -
net3          0             0           0             0
net0          2.72M        778.27M    76.32K        6.01M
net4          2.72M        778.27M    76.32K        6.01M
net6          2            140         1.30K         18.17K
net7          0             0           0             0
net2          0             0           0             0
net1          0             0           0             0
aggr1         0             0           0             0
ldoms-vsw0.vport0 935.40K     74.59M     13.15K        984.43K
    
```

```

ldoms-vsw0.vport1 933.26K 74.37M 11.42K 745.15K
ldoms-vsw0.vport2 933.24K 74.37M 11.46K 747.66K
ldoms-vsw1.vport1 202.26K 17.99M 179.75K 15.69M
ldoms-vsw1.vport0 202.37K 18.00M 189.00K 16.24M
-----
DOMAIN
ldg1

NAME          IPACKETS  RBYTES  OPACKETS  OBYTES
-----
net0          5.19K    421.57K 68         4.70K
net3          0         0        2.07K     256.93K
net4          0         0        4.37K     560.17K
net1          0         0        2.29K     303.24K
net5          149      31.19K  78         17.00K
net6          147      30.51K  78         17.29K
aggr2         0         0        0          0
ldoms-vsw0.vport3 162      31.69K  52         14.11K
ldoms-vsw0.vport2 163      31.74K  51         13.76K
ldoms-vsw0.vport1 176      42.99K  25         1.50K
ldoms-vsw0.vport0 158      40.19K  45         4.42K
-----
DOMAIN
ldg2

NAME          IPACKETS  RBYTES  OPACKETS  OBYTES
-----
net0          5.17K    418.90K 71         4.88K
net1          2.70K    201.67K 2.63K     187.01K
net2          132      36.40K  1.51K     95.07K
-----
DOMAIN
ldg3

NAME          IPACKETS  RBYTES  OPACKETS  OBYTES
-----
net0          5.16K    417.43K 72         4.90K
net1          2.80K    206.12K 2.67K     190.36K
net2          118      35.00K  1.46K     87.78K

```

가상 네트워크 장치에서 소비되는 물리적 네트워크 대역폭의 양 제어

대역폭 리소스 제어 기능을 통해 가상 네트워크 장치에서 소비되는 물리적 네트워크 대역폭을 제한할 수 있습니다. 이 기능은 최소한 Oracle Solaris 11 OS를 실행하고 가상 스위치로 구성된 서비스 도메인에서 지원됩니다. Oracle Solaris 10 서비스 도메인은 네트워크 대역폭 설정을 무시합니다. 이 기능은 하나의 게스트 도메인이 사용 가능한 물리적 네트워크 대역폭을 남김 없이 점유하지 않도록 합니다.

ldm add-vnet 및 ldm set-vnet 명령을 maxbw 등록 정보에 대한 값과 함께 사용하여 대역폭 제한을 지정합니다. ldm list-bindings 또는 ldm list-domain -o network 명령을 사용하여 기존 가상 네트워크 장치에 대한 maxbw 등록 정보 값을 봅니다. 최소 대역폭 제한은 10Mbps입니다.

네트워크 대역폭 제한

대역폭 리소스 제어는 가상 스위치를 통과하는 트래픽에만 적용됩니다. 따라서 Vnet 간 트래픽은 이 제한에 적용되지 않습니다. 구성된 물리적 백엔드 장치가 없는 경우 대역폭 리소스 제어를 무시할 수 있습니다.

최소 지원되는 대역폭 제한은 서비스 도메인의 Oracle Solaris 네트워크 스택에 따라 달라집니다. 대역폭 제한은 원하는 모든 높은 값으로 구성할 수 있습니다. 상한은 없습니다. 대역폭 제한은 대역폭이 구성된 값을 초과하지 않도록만 합니다. 따라서 가상 스위치에 지정된 물리적 네트워크 장치의 링크 속도보다 큰 값으로 대역폭 제한을 구성할 수 있습니다.

네트워크 대역폭 제한 설정

`ldm add-vnet` 명령을 사용하여 가상 네트워크 장치를 만들고 `maxbw` 등록 정보에 대한 값을 제공하여 대역폭 제한을 지정합니다.

```
primary# ldm add-vnet maxbw=limit if-name vswitch-name domain-name
```

`ldm set-vnet` 명령을 사용하여 기존 가상 네트워크 장치에 대한 대역폭 제한을 지정합니다.

```
primary# ldm set-vnet maxbw=limit if-name domain-name
```

`maxbw` 등록 정보에 대해 빈 값을 지정하여 대역폭 제한을 지울 수도 있습니다.

```
primary# ldm set-vnet maxbw= if-name domain-name
```

다음 예에서는 `ldm` 명령을 사용하여 대역폭 제한을 지정하는 방법을 보여줍니다. 대역폭은 단위와 함께 정수로 지정됩니다. 단위는 초당 메가비트의 경우 `M` 또는 초당 기가비트의 경우 `G`입니다. 단위를 지정하지 않을 경우 단위는 초당 메가비트입니다.

예 41 가상 네트워크 장치를 만들 때 대역폭 제한 설정

다음 명령은 대역폭 제한이 100Mbps인 가상 네트워크 장치(`vnet0`)를 만듭니다.

```
primary# ldm add-vnet maxbw=100M vnet0 primary-vsw0 ldg1
```

대역폭 제한을 최소값인 10Mbps보다 낮게 설정하려고 시도하면 다음 명령은 오류 메시지를 표시합니다.

```
primary# ldm add-vnet maxbw=1M vnet0 primary-vsw0 ldg1
```

예 42 기존 가상 네트워크 장치에 대한 대역폭 제한 설정

다음 명령은 기존 `vnet0` 장치에 대한 대역폭 제한을 200Mbps로 설정합니다.

실시간 네트워크 트래픽 패턴에 따라 대역폭의 양이 지정된 200Mbps 제한에 도달하지 못할 수 있습니다. 예를 들어, 대역폭은 200Mbps 제한을 초과하지 않는 95Mbps가 될 수 있습니다.

```
primary# ldm set-vnet maxbw=200M vnet0 ldg1
```

다음 명령은 기존 vnet0 장치에 대한 대역폭 제한을 2Gbps로 설정합니다.

MAC 레이어에서 대역폭에 대한 상한이 없으므로 기본 물리적 네트워크 속도가 2Gbps 미만이라도 제한을 2Gbps로 설정할 수 있습니다. 이러한 경우 대역폭 제한 효과가 없습니다.

```
primary# ldm set-vnet maxbw=2G vnet0 ldg1
```

예 43 기존 가상 네트워크 장치에 대한 대역폭 제한 지우기

다음 명령은 지정된 네트워크 장치(vnet0)에 대한 대역폭 제한을 지웁니다. 이 값을 지우면 가상 네트워크 장치는 기본 물리적 장치에서 제공하는 사용 가능한 최대 대역폭을 사용합니다.

```
primary# ldm set-vnet maxbw= vnet0 ldg1
```

예 44 기존 가상 네트워크 장치의 대역폭 제한 보기

ldm list-bindings 명령은 지정된 가상 네트워크 장치에 대한 maxbw 등록 정보의 값을 표시합니다(정의된 경우).

다음 명령은 vnet3 가상 네트워크 장치에 15Mbps의 대역폭 제한이 있음을 보여 줍니다. 설정된 대역폭 제한이 없을 경우 MAXBW 필드는 비어 있습니다.

```
primary# ldm ls-bindings -e -o network ldg3
```

```
NAME
ldg3

MAC
00:14:4f:f8:5b:12

NETWORK
NAME      SERVICE      MACADDRESS PVID|PVLAN|VIDs
----      -
vnet3    primary-vsw0@primary 00:14:4f:fa:ba:b9 1|--|--
  DEVICE :network@0      ID   :0
  LINKPROP  :--      MTU  :1500
  MAXBW    :15M      MODE  :--
  CUSTOM   :disable
  PRIORITY :--      COS   :--
  PROTECTION :--

PEER      MACADDRESS      PVID|PVLAN|VIDs
----      -
primary-vsw0@primary 00:14:4f:f9:08:28 1|--|--
  LINKPROP  :--      MTU  :1500
  MAXBW    :--      LDC  :0x0
  MODE     :--
```

다음과 같이 dladm show-linkprop 명령을 사용하여 maxbw 등록 정보 값을 볼 수도 있습니다.

```
# dladm show-linkprop -p maxbw
LINK          PROPERTY PERM VALUE EFFECTIVE DEFAULT POSSIBLE
...
```

```
ldoms-vsw0.vport0 maxbw rw 15 15 -- --
```

가상 장치 식별자 및 네트워크 인터페이스 이름

가상 스위치 또는 가상 네트워크 장치를 도메인에 추가하는 경우 `id` 등록 정보를 설정하여 장치 번호를 지정할 수 있습니다.

```
primary# ldm add-vsw [id=switch-id] vswitch-name domain-name
primary# ldm add-vnet [id=network-id] if-name vswitch-name domain-name
```

도메인의 가상 스위치 및 가상 네트워크 장치마다 고유한 장치 번호가 지정되는데, 이 번호는 도메인이 바인드될 때 지정됩니다. `id` 등록 정보를 설정하여 명시적 장치 번호로 가상 스위치 또는 가상 네트워크 장치를 추가한 경우 지정된 장치 번호가 사용됩니다. 그렇지 않은 경우 시스템에서 사용 가능한 가장 낮은 장치 번호를 자동으로 지정합니다. 이 경우 가상 스위치 또는 가상 네트워크 장치가 시스템에 추가된 방식에 따라 지정된 장치 번호가 달라집니다. 가상 스위치 또는 가상 네트워크 장치에 지정된 장치 번호는 도메인이 바인드될 때 `ldm list-bindings` 명령의 출력에 표시됩니다.

다음 예는 `primary` 도메인에 `primary-vsw0`라는 한 개의 가상 스위치가 있음을 보여줍니다. 이 가상 스위치의 장치 번호는 `0(switch@0)`입니다.

```
primary# ldm list-bindings primary
...
VSW
  NAME          MAC                NET-DEV DEVICE  DEFAULT-VLAN-ID PVID VID MTU MODE
  primary-vsw0  00:14:4f:fb:54:f2 net0    switch@0 1          1 5,6 1500
  ...
```

다음 예는 `ldg1` 도메인에 `vnet` 및 `vnet1`이라는 두 개의 가상 네트워크 장치가 있음을 보여줍니다. `vnet` 장치의 장치 번호는 `0(network@0)`이고, `vnet1` 장치의 장치 번호는 `1(network@1)`입니다.

```
primary# ldm list-bindings ldg1
...
NETWORK
  NAME SERVICE          DEVICE  MAC                MODE PVID VID MTU
  vnet primary-vsw0@primary network@0 00:14:4f:fb:e0:4b hybrid 1      1500
  ...
  vnet1 primary-vsw0@primary network@1 00:14:4f:f8:e1:ea      1      1500
  ...
```

마찬가지로, 가상 네트워크 장치가 있는 도메인에서 Oracle Solaris OS가 실행 중인 경우, 가상 네트워크 장치는 네트워크 인터페이스 `vnetN`을 사용합니다. 그러나 가상 네트워크 장치의 네트워크 인터페이스 번호 `N`이 반드시 가상 네트워크 장치의 장치 번호 `n`과 같을 필요는 없습니다.

주 - Oracle Solaris 11 시스템에서는 `netN` 형식의 일반 링크 이름이 `vswn`과 `vnetn`에 지정됩니다. `dladm show-phys` 명령을 사용하면 어떤 `netn` 이름이 `vswn` 및 `vnetn` 장치에 매핑되는지 확인할 수 있습니다.



주의 - Oracle Solaris OS에서는 네트워크 인터페이스의 이름과 장치 번호 기반의 가상 스위치 또는 가상 네트워크 장치 간 매핑이 유지됩니다. 가상 스위치 또는 가상 네트워크 장치에 장치 번호가 명시적으로 지정되지 않은 경우, 도메인의 바인드가 취소되었다가 다시 바인드될 때 장치 번호가 변경될 수 있습니다. 이 경우 도메인에서 실행되는 OS에 의해 지정된 네트워크 인터페이스 이름도 변경되고 기존 시스템 구성이 사용할 수 없는 상태로 설정될 수 있습니다. 이러한 상황은 가상 스위치 또는 가상 네트워크 인터페이스가 도메인 구성에서 제거된 경우 발생할 수 있습니다.

`ldm list-*` 명령을 사용하여 가상 스위치 또는 가상 네트워크 장치에 해당하는 Oracle Solaris OS 네트워크 인터페이스 이름을 직접 확인할 수는 없습니다. 그러나 `ldm list -l` 명령의 결과와 Oracle Solaris OS의 `/devices` 아래에 있는 항목의 결과를 조합하여 이 정보를 확인할 수는 있습니다.

Oracle Solaris 11 네트워크 인터페이스 이름 찾기

Oracle Solaris 11 시스템에서 `ldm list-netdev` 명령을 사용하여 Oracle Solaris OS 네트워크 인터페이스 이름을 찾을 수 있습니다. 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 예제는 `ldm list-netdev` 및 `ldm list -o network` 명령을 보여줍니다. `ldm list -o network` 명령은 `NAME` 필드에 가상 네트워크 장치를 보여줍니다. `ldm list-netdev` 출력은 `NAME` 열에 해당하는 OS 인터페이스 이름을 보여줍니다.

```
primary# ldm list -o network ldg1
....
NETWORK
  NAME          SERVICE          ID DEVICE      MAC          MODE
  PVID VID MTU  MAXBW LINKPROP
  vnet0-ldg1    primary-vsw0@primary 0  network@0  00:14:4f:fa:eb:4e 1
  1500
  vnet1-ldg1    svcdom-vsw0@svcdom  1  network@1  00:14:4f:f8:53:45 4
  1500
  PVLAN :400,community

primary# ldm list-netdev ldg1
DOMAIN
ldg1

NAME CLASS MEDIA STATE  SPEED OVER  LOC
-----
net0  VNET  ETHER up     0     vnet0 primary-vsw0/vnet0-ldg1
net1  VNET  ETHER up     0     vnet1 svcdom-vsw0/vnet1-ldg1
net2  VNET  ETHER unknown 0     vnet2 svcdom-vsw1/vnet2-ldg1
```

`ldm list-netdev` 출력이 정확한지 확인하려면 `ldg1`에서 `dladm show-phys` 및 `dladm show-linkprop -p mac-address` 명령을 실행합니다.

```
ldg1# dladm show-phys
LINK      MEDIA  STATE  SPEED  DUPLEX  DEVICE
net0      Ethernet up      0      unknown vnet0
net1      Ethernet up      0      unknown vnet1
net2      Ethernet unknown 0      unknown vnet2
```

```
ldg1# dladm show-linkprop -p mac-address
LINK PROPERTY PERM VALUE EFFECTIVE DEFAULT POSSIBLE
net0 mac-address rw 0:14:4f:fa:eb:4e 0:14:4f:fa:eb:4e 0:14:4f:fa:eb:4e --
net1 mac-address rw 0:14:4f:f8:53:45 0:14:4f:f8:53:45 0:14:4f:f8:53:45 --
```

▼ Oracle Solaris OS 네트워크 인터페이스 이름을 찾는 방법

이 절차에서는 ldg1에서 net-c에 해당하는 Oracle Solaris OS 네트워크 인터페이스 이름을 찾는 방법에 대해 설명합니다. 이 예는 또한 가상 네트워크 장치 대신 가상 스위치의 네트워크 인터페이스 이름을 찾을 경우 차이점도 보여줍니다. 이 예제 절차에서는 게스트 도메인 ldg1에 net-a 및 net-c라는 두 개의 가상 네트워크 장치가 있습니다.

1. **ldm** 명령을 사용하여 net-c에 대한 가상 네트워크 장치 번호를 찾습니다.

```
primary# ldm list -l ldg1
...
NETWORK
NAME SERVICE DEVICE MAC
net-a primary-vsw0@primary network@00 00:14:4f:f8:91:4f
net-c primary-vsw0@primary network@2 00:14:4f:f8:dd:68
...
```

net-c에 대한 가상 네트워크 장치 번호는 2(network@2)입니다.

가상 스위치의 네트워크 인터페이스 이름을 확인하려면 가상 스위치 장치 번호 *n*(예: switch@*n*)을 찾습니다.

2. ldg1에 로그인한 다음 /devices에서 이 장치 번호에 해당하는 항목을 찾아 ldg1에서 해당 네트워크 인터페이스를 찾습니다.

```
ldg1# uname -n
ldg1
ldg1# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
```

네트워크 인터페이스 이름은 콜론 뒤에 오는 항목(vnet1)의 일부입니다.

가상 스위치의 네트워크 인터페이스 이름을 확인하려면 -name 옵션에 대한 인수를 virtual-network-switch@*n**로 바꿉니다. 그런 다음 이름이 vsw*N*인 네트워크 인터페이스를 찾습니다.

3. 1단계의 net-c에 대한 ldm list -l 출력에 표시된 것과 같이, vnet1의 MAC 주소가 00:14:4f:f8:dd:68인지 확인합니다.

■ Oracle Solaris 11 OS.

- a. vnet1에 대해 지정할 인터페이스의 이름을 확인합니다.

```
ldg1# dladm show-phys |grep vnet1
net2 Ethernet up 0 unknown vnet1
```

- b. net2의 MAC 주소를 확인합니다.

```
# dladm show-linkprop -p mac-address net2
LINK PROPERTY   PERM VALUE           EFFECTIVE             DEFAULT POSSIBLE
net2 mac-address rw  00:14:4f:f8:dd:68  00:14:4f:f8:dd:68  --      --
```

이 예의 MAC 주소는 1단계의 `ldm list -l` 명령 출력에 표시된 `net-c` 주소와 일치합니다.

■ Oracle Solaris 10 OS.

```
ldg1# ifconfig vnet1
vnet1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 0.0.0.0 netmask 0
        ether 0:14:4f:f8:dd:68
```

자동 또는 수동 MAC 주소 지정

사용하려는 논리적 도메인, 가상 스위치 및 가상 네트워크 수에 지정할 MAC(매체 액세스 제어) 주소가 충분해야 합니다. Logical Domains Manager가 자동으로 논리적 도메인, 가상 네트워크 및 가상 스위치에 MAC 주소를 지정하도록 할 수도 있고, 지정된 MAC 주소의 사용자 고유 풀에서 수동으로 MAC 주소를 지정할 수도 있습니다. MAC 주소를 설정하는 `ldm` 하위 명령은 `add-domain`, `add-vsw`, `set-vsw`, `add-vnet` 및 `set-vnet`입니다. 이러한 하위 명령에 MAC 주소를 지정하지 않은 경우 Logical Domains Manager에서 자동으로 지정합니다.

Logical Domains Manager가 MAC 주소를 지정하도록 할 경우 논리적 도메인에만 사용할 수 있는 MAC 주소 블록이 사용된다는 이점이 있습니다. 또한 Logical Domains Manager는 같은 서버넷에 있는 다른 Logical Domains Manager 인스턴스와 MAC 주소가 충돌하는 것을 찾아내어 방지합니다. 따라서 MAC 주소 풀을 수동으로 관리할 필요가 없습니다.

MAC 주소는 논리적 도메인이 만들어지거나 네트워크 장치가 도메인에 구성되는 즉시 지정됩니다. 또한 장치 또는 논리적 도메인 자체가 제거될 때까지 지정된 주소가 유지됩니다.

도메인에 지정된 MAC 주소의 범위

도메인에는 다음과 같은 512K MAC 주소 블록이 지정됩니다.

```
00:14:4F:F8:00:00 ~ 00:14:4F:FF:FF:FF
```

이보다 낮은 256K 주소는 자동 MAC 주소 할당을 위해 Logical Domains Manager에서 사용하므로, 이 범위의 주소는 수동으로 요청할 수 없습니다.

```
00:14:4F:F8:00:00 - 00:14:4F:FB:FF:FF
```

이 범위의 위쪽 절반은 수동 MAC 주소 할당에 사용할 수 있습니다.

00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF

주 - Oracle Solaris 11에서 VNIC에 대한 MAC 할당은 이 범위 밖에 있는 주소를 사용합니다.

자동 지정 알고리즘

논리적 도메인이나 네트워크 장치를 만들 때 MAC 주소를 지정하지 않은 경우 Logical Domains Manager가 자동으로 해당 논리적 도메인이나 네트워크 장치에 MAC 주소를 할당합니다.

이 MAC 주소를 확인하기 위해 Logical Domains Manager에서는 반복적으로 주소를 선택한 다음 충돌 가능성이 있는지 확인합니다. 이 목적을 위해 설정된 256K 주소 범위에서 MAC 주소가 무작위로 선택됩니다. 중복된 MAC 주소를 후보로 선택할 가능성을 줄이기 위해 MAC 주소가 무작위로 선택됩니다.

그런 다음 중복된 MAC 주소가 실제로 지정되는 것을 방지하기 위해 선택된 주소는 다른 시스템에 있는 다른 Logical Domains Managers에 대해 검사됩니다. 사용된 알고리즘은 “[중복된 MAC 주소 감지](#)” [238]에 설명되어 있습니다. 주소가 이미 지정된 경우 Logical Domains Manager가 반복적으로 다른 주소를 선택하여 충돌 여부를 다시 검사합니다. 이 프로세스는 아직 할당되지 않은 MAC 주소를 찾을 때까지 또는 30초 시간 제한이 경과할 때까지 계속됩니다. 이 시간 제한에 도달하면 장치 만들기가 실패하고 다음과 비슷한 오류 메시지가 표시됩니다.

```
Automatic MAC allocation failed. Please set the vnet MAC address manually.
```

중복된 MAC 주소 감지

동일한 MAC 주소가 다른 장치에 할당되지 않도록 하기 위해 Logical Domains Managers가 장치에 지정하려고 하는 주소를 비롯한 멀티캐스트 메시지를 컨트롤 도메인의 기본 네트워크 인터페이스를 통해 보내는 방식으로 Logical Domains Manager가 다른 시스템에 있는 다른 Logical Domains Manager와 함께 검사를 수행합니다. MAC 주소를 지정하려고 시도하는 Logical Domains Manager는 1초간 응답을 기다립니다. 다른 Oracle VM Server for SPARC 사용 시스템에 있는 다른 장치에 이미 해당 MAC 주소가 지정된 경우 이 시스템의 Logical Domains Manager는 해당 MAC 주소가 포함된 응답을 보냅니다. 요청하는 Logical Domains Manager가 응답을 받으면 선택한 MAC 주소가 이미 할당되었음을 알게 되고 다른 주소를 반복적으로 선택합니다.

기본적으로 해당 멀티캐스트 메시지는 동일한 서브넷의 다른 관리자에만 전송됩니다. 기본 TTL(활성 시간)은 1입니다. TTL은 SMF(서비스 관리 기능) 등록 정보 `1dmd/hops`를 사용하여 구성할 수 있습니다.

각 Logical Domains Manager는 다음과 같은 작업을 담당합니다.

- 멀티캐스트 메시지 수신
- 도메인에 지정된 MAC 주소 추적
- 중복 확인
- 중복이 발생하지 않도록 응답

어떤 이유로 시스템에 있는 Logical Domains Manager가 종료된 경우, Logical Domains Manager가 작동 중지된 동안 중복된 MAC 주소가 발생할 수 있습니다.

자동 MAC 할당은 논리적 도메인 또는 네트워크 장치가 생성될 때 발생하며, 장치 또는 논리적 도메인이 제거될 때까지 유지됩니다.

주 - 논리적 도메인 또는 네트워크 장치가 생성될 때 중복된 MAC 주소를 감지하기 위한 검사가 수행되고 논리적 도메인이 시작됩니다.

Oracle Solaris 10을 실행하는 도메인에서 네트워크 어댑터 사용

Oracle Solaris 10 논리적 도메인 환경에서 서비스 도메인에서 실행되는 가상 스위치 서비스는 GLDv3 준수 네트워크 어댑터와 직접 상호 작용할 수 있습니다. GLDv3 비준수 네트워크 어댑터를 이러한 시스템에 사용할 수는 있지만, 가상 스위치가 이러한 시스템과 직접 연결될 수 없습니다. GLDv3 비준수 네트워크 어댑터 사용법에 대한 자세한 내용은 [“NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성” \[240\]](#)을 참조하십시오.

주 - Oracle Solaris 11 환경의 경우 GLDv3 준수 문제가 해당되지 않습니다.

링크 통합 사용에 대한 자세한 내용은 [“가상 스위치에서 링크 통합 사용” \[262\]](#)을 참조하십시오.

▼ GLDv3 준수 네트워크 어댑터인지 여부를 확인하는 방법

이 절차는 Oracle Solaris 10 도메인에만 적용됩니다.

- 네트워크 어댑터가 GLDv3 준수 네트워크 어댑터인지 확인합니다.
다음 예에서는 bge0이 네트워크 장치 이름으로 사용됩니다.

```
# dladm show-link bge0
```

```
bge0          type: non-vlan  mtu: 1500    device: bge0
```

type: 필드의 값은 다음 중 하나입니다.

- GLDV3 준수 드라이버의 유형은 non-vlan 또는 vlan입니다.
- GLDV3 비준수 드라이버의 유형은 legacy입니다.

NAT 및 경로 지정을 위해 가상 스위치 및 서비스 도메인 구성

Oracle Solaris 10 OS에서 가상 스위치(*vsw*)는 계층 2 스위치로서, 서비스 도메인에서 네트워크 장치로 사용할 수도 있습니다. 가상 스위치가 여러 논리적 도메인의 가상 네트워크 장치 간 스위치로만 사용되고 물리적 장치를 통해 외부 네트워크에 연결할 수 없도록 구성할 수 있습니다. 이 모드에서 *vsw*를 네트워크 장치로 만들고 서비스 도메인에서 IP 경로 지정을 사용으로 설정하면 가상 네트워크가 서비스 도메인을 라우터로 사용하여 외부와 통신할 수 있습니다. 이 작동 모드는 물리적 네트워크 어댑터가 GLDV3 비준수 어댑터일 경우 도메인에 대한 외부 연결을 제공하는 데 반드시 필요합니다.

이 구성의 이점은 다음과 같습니다.

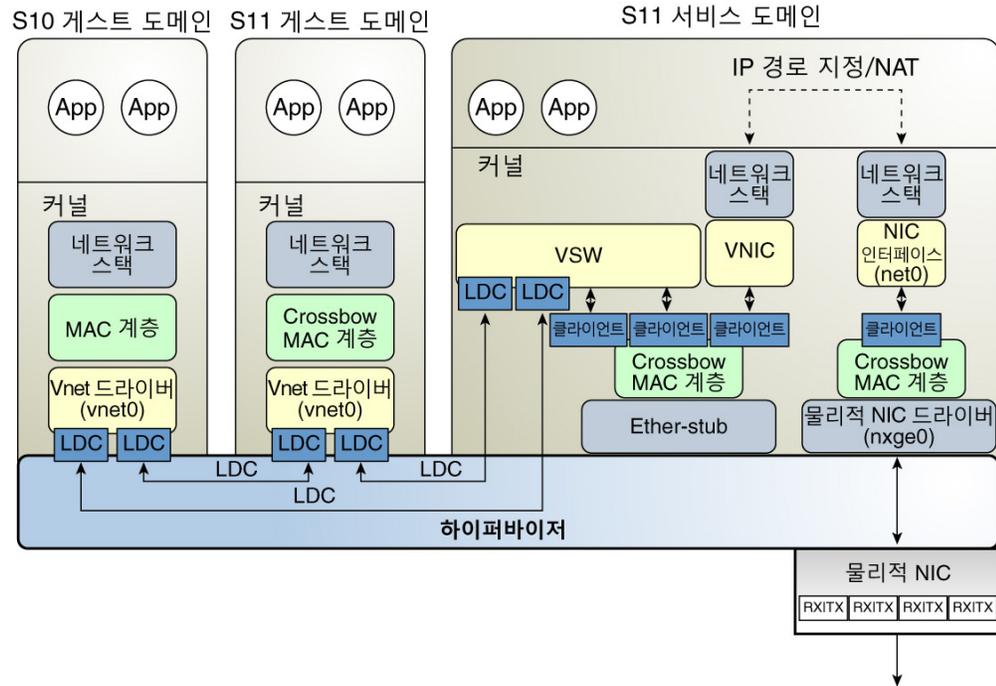
- 가상 스위치가 물리적 장치를 직접 사용할 필요가 없으며 기본 장치가 GLDV3 비준수 장치일 경우에도 외부 연결을 제공할 수 있습니다.
- Oracle Solaris OS의 IP 경로 지정 및 필터링 기능을 구성에 활용할 수 있습니다.

Oracle Solaris 11 시스템에서 NAT 구성

Oracle Solaris 11 네트워크 가상화 기능에는 의사 네트워크 장치인 *etherstub*가 포함되어 있습니다. 이 장치는 물리적 네트워크 장치와 비슷한 기능을 제공하지만, 클라이언트와의 개인 통신에 한합니다. 이 의사 장치는 가상 네트워크 간 개인 통신을 제공하는 가상 스위치에 대한 네트워크 백엔드 장치로 사용할 수 있습니다. *etherstub* 장치를 백엔드 장치로 사용하면 게스트 도메인이 같은 *etherstub* 장치에 있는 VNIC와도 통신할 수 있습니다. 이 방식으로 *etherstub*를 사용하면 게스트 도메인이 서비스 도메인의 영역과 통신할 수 있습니다. *etherstub* 장치를 만들려면 `dladm create-etherstub` 명령을 사용하십시오.

다음 다이어그램은 가상 스위치, *etherstub* 장치 및 VNIC를 사용하여 서비스 도메인에서 NAT (Network Address Translation)를 설정하는 방법을 보여줍니다.

그림 15 가상 네트워크 경로 지정



지속 경로 사용을 고려해 볼 수 있습니다. 자세한 내용은 [Troubleshooting Network Administration Issues in Oracle Solaris 11.3](#) 의 “Troubleshooting Issues When Adding a Persistent Route” 및 [Configuring and Managing Network Components in Oracle Solaris 11.3](#) 의 “Creating Persistent (Static) Routes”를 참조하십시오.

▼ 도메인에 대한 외부 연결을 제공하기 위해 가상 스위치를 설정하는 방법 (Oracle Solaris 11)

1. Oracle Solaris 11 etherstub 장치를 만듭니다.

```
primary# dladm create-etherstub stub0
```

2. stub0을 물리적 백엔드 장치로 사용하는 가상 스위치를 만듭니다.

```
primary# ldm add-vsw net-dev=stub0 primary-stub-vsw0 primary
```

3. stub0 장치에 VNIC를 만듭니다.

```
primary# dladm create-vnic -l stub0 vnic0
```

4. vnic0을 네트워크 인터페이스로 구성합니다.

```
primary# ipadm create-ip vnic0
primary# ipadm create-addr -T static -a 192.168.100.1/24 vnic0/v4static
```

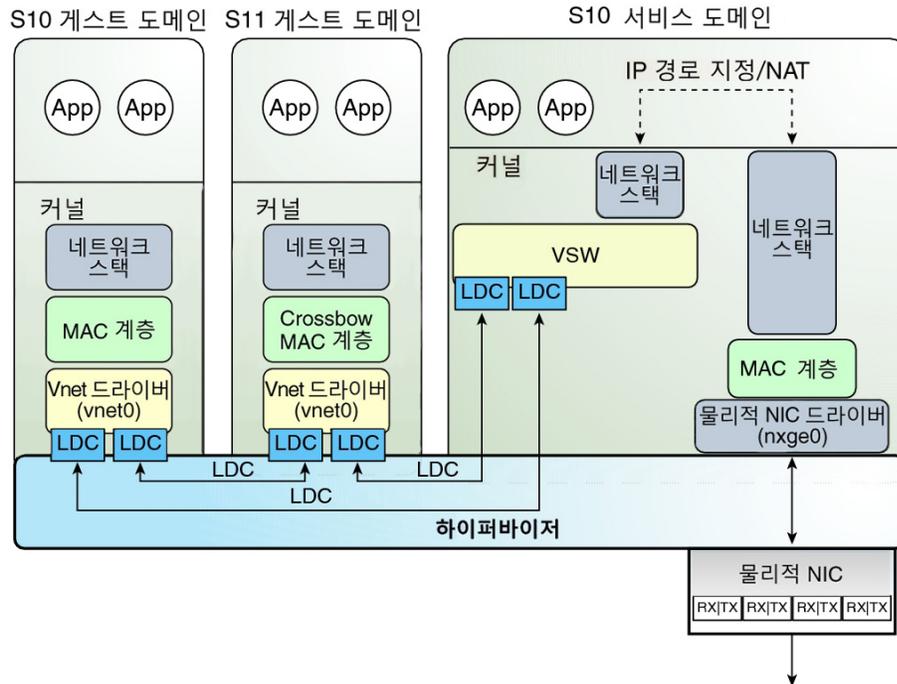
5. IPv4 전달을 사용으로 설정하고 NAT 규칙을 만듭니다.

[Configuring and Managing Network Components in Oracle Solaris 11.3](#) 의 “Customizing IP Interface Properties and Addresses” 및 [Oracle Solaris Administration: IP Services](#) 의 “Packet Forwarding and Routing on IPv4 Networks”을 참조하십시오.

Oracle Solaris 10 시스템에서 NAT 구성

다음 다이어그램은 게스트 도메인에 대한 외부 연결을 제공하기 위해 가상 스위치를 사용하여 서비스 도메인에서 NAT(Network Address Translation)를 구성하는 방법을 보여줍니다.

그림 16 가상 네트워크 경로 지정



▼ 도메인에 대한 외부 연결을 제공하기 위해 Oracle Solaris 10 서비스 도메인에 가상 스위치를 설정하는 방법

1. 연관된 물리적 장치가 없는 가상 스위치를 만듭니다.
주소를 지정한 경우 가상 스위치의 MAC 주소가 고유한지 확인합니다.

```
primary# ldm add-vsw [mac-addr=xx:xx:xx:xx:xx:xx] ldg1-vsw0 ldg1
```
2. 가상 스위치를 도메인에서 사용 중인 물리적 네트워크 장치 및 네트워크 장치로 만듭니다.
가상 스위치 만들기에 대한 자세한 내용은 [가상 스위치를 주 인터페이스로 구성하는 방법 \[38\]](#)을 참조하십시오.
3. 필요한 경우 DHCP용 가상 스위치 장치를 구성합니다.
DHCP용 가상 스위치 장치 만들기에 대한 자세한 내용은 [가상 스위치를 주 인터페이스로 구성하는 방법 \[38\]](#)을 참조하십시오.
4. 필요한 경우 `/etc/dhcp.vsw` 파일을 만듭니다.
5. 서비스 도메인에서 IP 경로 지정을 구성한 다음 모든 도메인에서 필요한 경로 지정 테이블을 설정합니다.
IP 경로 지정에 대한 자세한 내용은 [Oracle Solaris Administration: IP Services](#) 의 “[Packet Forwarding and Routing on IPv4 Networks](#)”을 참조하십시오.

Oracle VM Server for SPARC 환경에서 IPMP 구성

Oracle VM Server for SPARC 소프트웨어는 가상 네트워크 장치에 대해 링크 기반 IPMP(IP Network Multipathing)를 지원합니다. 가상 네트워크 장치에 대해 IPMP 그룹을 구성할 경우 링크 기반 감지를 사용할 그룹을 구성하십시오. 이전 버전의 Oracle VM Server for SPARC (Logical Domains) 소프트웨어를 사용 중인 경우 가상 네트워크 장치에 대해 프로브 기반 감지만 구성할 수 있습니다.

Oracle Solaris 11 도메인에서 가상 네트워크 장치를 IPMP 그룹으로 구성

[그림 17](#)는 가상 네트워크 두 개(`vnet0` 및 `vnet1`)가 서비스 도메인의 별도 가상 스위치 인스턴스(`vsw0` 및 `vsw1`)에 연결되어 있고, 서로 다른 물리적 인터페이스 두 개를 사용할을 보여 줍니다. 물리적 인터페이스는 Oracle Solaris 11 서비스 도메인의 `net0` 및 `net1`입니다.

서비스 도메인에서 물리적 링크 오류가 발생하면 해당 물리적 장치에 바인드된 가상 스위치 장치에서도 링크 오류가 발생합니다. 그러면 가상 스위치가 이 가상 스위치에 바인드된 해당 가상 네트워크 장치에 오류를 전파합니다. 가상 네트워크 장치는 이 링크 이벤트에 대한 알림을 LDom_A 게스트의 IP 계층에 전송합니다. 그러면 IPMP 그룹에 있는 다른 가상 네트워크 장치로 페일오버됩니다.

그림 17 개별 가상 스위치 인스턴스에 연결된 두 개의 가상 네트워크(Oracle Solaris 11)

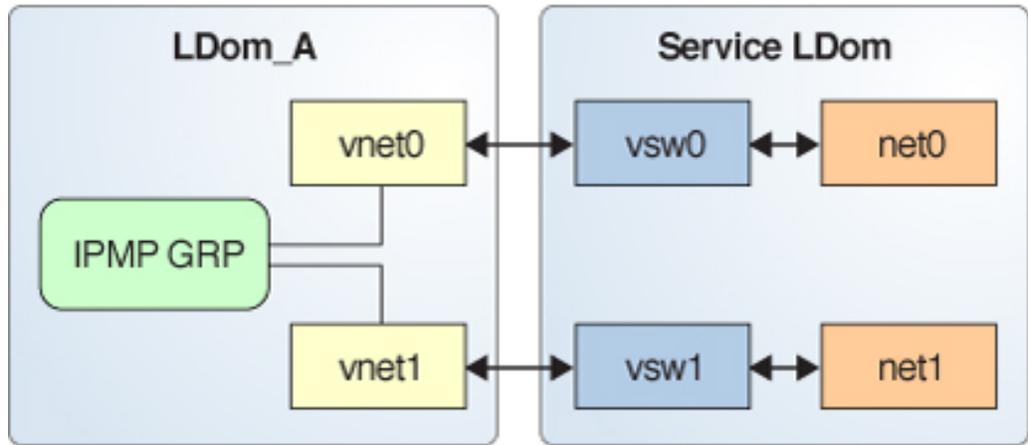
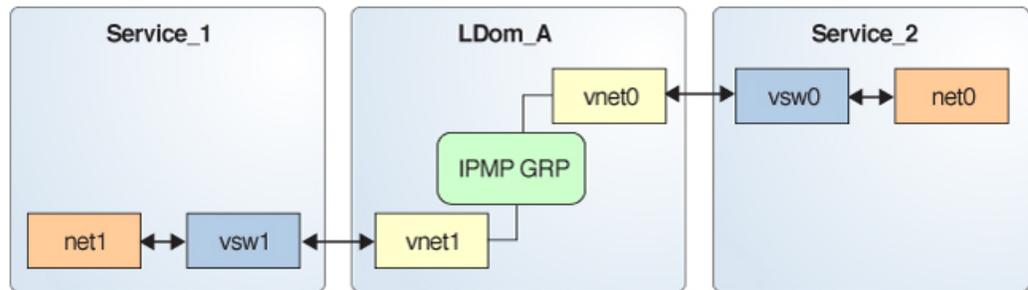


그림 18는 각각의 가상 네트워크 장치(vnet0 및 vnet1)를 다른 서비스 도메인의 가상 스위치 인스턴스에 연결하면 논리적 도메인의 신뢰성이 보다 향상될 수 있음을 보여 줍니다. 이 경우 LDom_A는 물리적 네트워크 오류 이외에 가상 네트워크 오류를 감지하고 서비스 도메인 충돌 및 종료 후 페일오버를 트리거합니다.

그림 18 다른 서비스 도메인에 연결된 개별 가상 네트워크 장치(Oracle Solaris 11)



자세한 내용은 [Oracle Solaris 11.3 Information Library](#)의 “Establishing an Oracle Solaris Network”를 참조하십시오.

Oracle Solaris 10 도메인에서 가상 네트워크 장치를 IPMP 그룹으로 구성

그림 19는 가상 네트워크 두 개(`vnet0` 및 `vnet1`)가 서비스 도메인의 별도 가상 스위치 인스턴스(`vsw0` 및 `vsw1`)에 연결되어 있고, 서로 다른 물리적 인터페이스 두 개를 사용함을 보여 줍니다. 물리적 인터페이스는 Oracle Solaris 10 서비스 도메인의 `nxge0` 및 `nxge1`입니다.

서비스 도메인에서 물리적 링크 오류가 발생하면 해당 물리적 장치에 바인드된 가상 스위치 장치에서도 링크 오류가 발생합니다. 그러면 가상 스위치가 이 가상 스위치에 바인드된 해당 가상 네트워크 장치에 오류를 전파합니다. 가상 네트워크 장치는 이 링크 이벤트에 대한 알림을 `LDom_A` 게스트의 IP 계층에 전송합니다. 그러면 IPMP 그룹에 있는 다른 가상 네트워크 장치로 페일오버됩니다.

그림 19 개별 가상 스위치 인스턴스에 연결된 두 개의 가상 네트워크(Oracle Solaris 10)

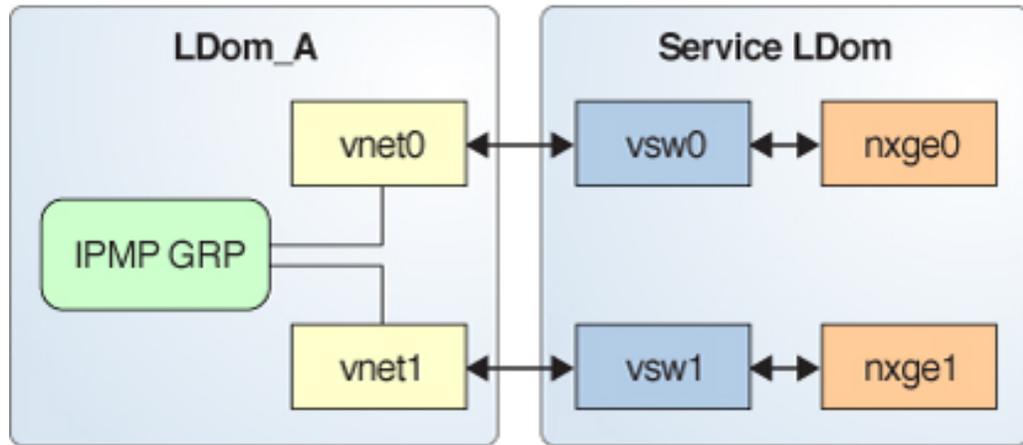
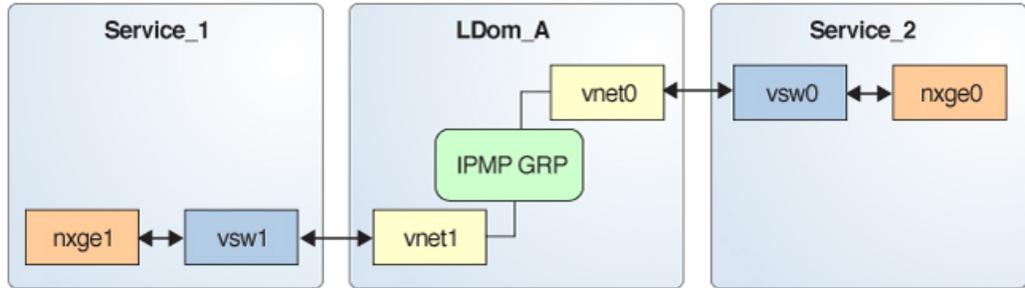


그림 20는 각각의 가상 네트워크 장치(`vnet0` 및 `vnet1`)를 다른 서비스 도메인의 가상 스위치 인스턴스에 연결하면 논리적 도메인의 신뢰성이 보다 향상될 수 있음을 보여 줍니다. 이 경우 `LDom_A`는 물리적 네트워크 오류 이외에 가상 네트워크 오류를 감지하고 서비스 도메인 충돌 및 종료 후 페일오버를 트리거합니다.

그림 20 다른 서비스 도메인에 연결된 개별 가상 네트워크 장치(Oracle Solaris 10)



자세한 내용은 [Oracle Solaris Administration: IP Services](#)를 참조하십시오.

서비스 도메인에서 IPMP 구성 및 사용

Oracle Solaris 11 시스템에서는 가상 네트워크나 도메인이 없는 시스템과 같은 방법으로, 물리적 인터페이스를 하나의 그룹으로 구성하여 서비스 도메인에서 IPMP를 구성할 수 있습니다. Oracle Solaris 10 시스템에서는 가상 스위치 인터페이스를 하나의 그룹으로 구성하여 서비스 도메인에서 IPMP를 구성할 수 있습니다. [그림 21](#) 및 [그림 22](#)는 다른 물리적 장치 두 개에 바인드된 가상 스위치 인스턴스(vsw0 및 vsw1) 두 개를 보여 줍니다. 그런 다음 두 개의 가상 스위치 인터페이스를 만들어 IPMP 그룹으로 구성할 수 있습니다. 물리적 링크 오류가 발생하면 해당 물리적 장치에 바인드된 가상 스위치 장치에서도 링크 오류가 발생합니다. 그런 다음 가상 스위치 장치가 이 링크 이벤트에 대한 알림을 서비스 도메인의 IP 계층에 전송합니다. 그러면 IPMP 그룹에 있는 다른 가상 스위치 장치로 페일오버됩니다. 두 개의 물리적 인터페이스는 Oracle Solaris 11의 경우 net0 및 net1이고, Oracle Solaris 10의 경우 nxge0 및 nxge1입니다.

그림 21 IPMP 그룹의 일부로 구성된 두 개의 물리적 NIC(Oracle Solaris 11)

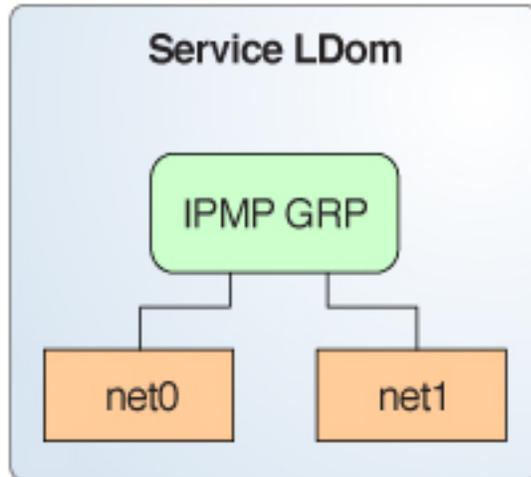
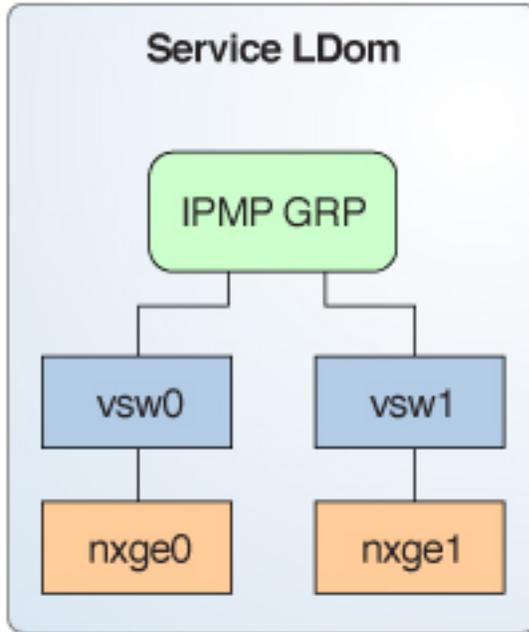


그림 22 IPMP 그룹의 일부로 구성된 두 개의 가상 스위치 인터페이스(Oracle Solaris 10)



Oracle VM Server for SPARC 가상 네트워킹에서 링크 기반 IPMP 사용

가상 네트워크 및 가상 스위치 장치는 네트워크 스택에 대한 링크 상태 업데이트를 지원합니다. 기본적으로 가상 네트워크 장치는 가상 링크(가상 스위치에 대한 LDC) 및 물리적 링크의 상태를 보고합니다. 이 구성은 기본적으로 사용으로 설정되므로 구성 단계를 추가로 수행할 필요가 없습니다.

주 - linkprop 등록 정보는 백업 장치 지원에 따라 기본적으로 `phys-state`로 설정됩니다. linkprop 등록 정보를 수동으로 사용 안함으로 설정하고 이 등록 정보를 `phys-state` 값으로 설정하려고 하지 않는 한 이 절에 나와 있는 작업을 수행할 필요가 없습니다.

표준 Oracle Solaris 네트워크 관리 명령(예: `dladm` and `ifconfig`)을 사용하여 링크 상태를 확인할 수 있습니다. 또한 링크 상태는 `/var/adm/messages` 파일에도 기록됩니다. Oracle Solaris

10의 경우, `dladm(1M)` 및 `ifconfig(1M)` 매뉴얼 페이지를 참조하십시오. Oracle Solaris 11의 경우, `dladm(1M)`, `ipadm(1M)` 및 `ipmpstat(1M)` 매뉴얼 페이지를 참조하십시오.

주 - Oracle VM Server for SPARC 시스템에서는 현재 링크 상태 미인식 및 링크 상태 인식 `vnet` 및 `vsw` 드라이버를 모두 실행할 수 있습니다. 그러나 링크 기반 IPMP를 구성하려는 경우에는 링크 상태 인식 드라이버를 설치해야 합니다. 물리적 링크 상태 업데이트를 사용으로 설정하려면 `vnet` 및 `vsw` 드라이버를 Oracle Solaris 10 1/13 OS로 업그레이드하고 Logical Domains Manager 버전 1.3 이상을 실행하십시오.

▼ 물리적 링크 상태 업데이트를 구성하는 방법

이 절차에서는 가상 네트워크 장치에 대한 물리적 링크 상태 업데이트를 사용으로 설정하는 방법을 보여줍니다.

`linkprop=phys-state` 옵션을 `ldm add-vsw` 및 `ldm set-vsw` 명령에 지정하면 가상 스위치 장치에 대해서도 물리적 링크 상태 업데이트를 사용으로 설정할 수 있습니다.

주 - `linkprop=phys-state` 옵션은 가상 스위치 장치 자체가 인터페이스로 생성된 경우에만 사용해야 합니다. `linkprop=phys-state`를 지정했는데 물리적 링크가 작동하지 않을 경우, 가상 스위치에 대한 연결이 작동하더라도 가상 네트워크 장치가 링크 상태를 `down`으로 보고합니다. 이러한 상황은 Oracle Solaris OS에서 현재 두 개의 고유 링크 상태(예: `virtual-link-state` 및 `physical-link-state`)를 보고할 수 있는 인터페이스를 제공하지 않기 때문에 발생합니다.

1. 관리자로 전환합니다.

Oracle Solaris 11.3의 경우, [Securing Users and Processes in Oracle Solaris 11.3](#)의 1장, "About Using Rights to Control Users and Processes"를 참조하십시오.

2. 가상 장치에 대해 물리적 링크 상태 업데이트를 사용으로 설정합니다.

다음과 같은 방법으로 가상 네트워크 장치에 대해 물리적 링크 상태 업데이트를 사용으로 설정할 수 있습니다.

- `ldm add-vnet` 명령을 실행할 때 `linkprop=phys-state`를 지정하여 가상 네트워크 장치를 만듭니다.

`linkprop=phys-state` 옵션을 지정하면 물리적 링크 상태 업데이트를 확인하고 이를 스택에 보고하도록 가상 네트워크 장치가 구성됩니다.

주 - `linkprop=phys-state`를 지정했는데 물리적 링크가 작동하지 않을 경우, 가상 스위치에 대한 연결이 작동하더라도 가상 네트워크 장치가 링크 상태를 `down`으로 보고합니다. 이러한 상황은 Oracle Solaris OS에서 현재 두 개의 고유 링크 상태(예: `virtual-link-state` 및 `physical-link-state`)를 보고할 수 있는 인터페이스를 제공하지 않기 때문에 발생합니다.

```
primary# ldm add-vnet linkprop=phys-state if-name vswitch-name domain-name
```

다음 예는 논리적 도메인 `ldom1`에서 `primary-vsw0`에 연결된 `ldom1_vnet0`에 대한 물리적 링크 상태 업데이트를 사용으로 설정합니다.

```
primary# ldm add-vnet linkprop=phys-state ldom1_vnet0 primary-vsw0 ldom1
```

- `ldm set-vnet` 명령을 실행할 때 `linkprop=phys-state`를 지정하여 기존 가상 네트워크 장치를 수정합니다.

```
primary# ldm set-vnet linkprop=phys-state if-name domain-name
```

다음 예는 논리적 도메인 `ldom1`에서 `vnet0`에 대한 물리적 링크 상태 업데이트를 사용으로 설정합니다.

```
primary# ldm set-vnet linkprop=phys-state ldom1_vnet0 ldom1
```

물리적 링크 상태 업데이트를 사용 안함으로 설정하려면 `ldm set-vnet` 명령을 실행할 때 `linkprop=`를 지정합니다.

다음 예는 논리적 도메인 `ldom1`에서 `ldom1_vnet0`에 대한 물리적 링크 상태 업데이트를 사용 안함으로 설정합니다.

```
primary# ldm set-vnet linkprop= ldom1_vnet0 ldom1
```

예 45 링크 기반 IPMP 구성

다음 예에서는 물리적 링크 상태 업데이트를 사용으로 설정하거나 설정하지 않고 링크 기반 IPMP를 구성하는 방법에 대해 설명합니다.

- 다음 예는 도메인에 가상 네트워크 장치를 구성합니다. 각 가상 네트워크 장치는 링크 기반 IPMP를 사용하기 위해 서비스 도메인의 개별 가상 스위치 장치에 연결되어 있습니다.

주 - 이러한 가상 네트워크 장치에는 테스트 주소가 구성되지 않습니다. 또한 `ldm add-vnet` 명령을 사용하여 이러한 가상 네트워크 장치를 만들 때 추가 구성을 수행할 필요가 없습니다.

다음 명령은 도메인에 가상 네트워크 장치를 추가합니다. `linkprop=phys-state`가 지정되어 있지 않으므로 가상 스위치에 대한 링크에 대해서만 상태 변경을 모니터링합니다.

```
primary# ldm add-vnet ldom1_vnet0 primary-vsw0 ldom1
```

```
primary# ldm add-vnet ldom1_vnet1 primary-vsw1 ldom1
```

다음 명령은 게스트 도메인에서 가상 네트워크 장치를 구성하고 이 장치를 IPMP 그룹에 지정합니다. 링크 기반 오류 감지를 사용 중이므로 이러한 가상 네트워크 장치에는 테스트 주소가 구성되지 않습니다.

- **Oracle Solaris 10 OS:** `ifconfig` 명령을 사용합니다.

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

두번째와 세번째 명령은 ipmp0 인터페이스를 IP 주소로 적절하게 구성합니다.

- **Oracle Solaris 11 OS:** ipadm 명령을 사용합니다.

net0 및 net1은 vnet0 및 vnet1 각각에 대한 Oracle Solaris 11 베니티 이름입니다.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 ipmp0
```

- 다음 예는 도메인에 가상 네트워크 장치를 구성합니다. 각 도메인은 링크 기반 IPMP를 사용하기 위해 서비스 도메인의 개별 가상 스위치 장치에 연결되어 있습니다. 가상 네트워크 장치도 물리적 링크 상태 업데이트를 확인하도록 구성되어 있습니다.

```
primary# ldm add-vnet linkprop=phys-state ldom1_vnet0 primary-vsw0 ldom1
primary# ldm add-vnet linkprop=phys-state ldom1_vnet1 primary-vsw1 ldom1
```

주 - 도메인을 성공적으로 바인드하려면 가상 스위치에 물리적 네트워크 장치가 지정되어 있어야 합니다. 도메인이 이미 바인드되었으며 가상 스위치에 물리적 네트워크 장치가 지정되지 않은 경우 ldm add-vnet 명령이 실패합니다.

다음 명령은 가상 네트워크 장치를 만들어 이 장치를 IPMP 그룹에 지정합니다.

- **Oracle Solaris 10 OS:** ifconfig 명령을 사용합니다.

```
# ifconfig vnet0 plumb
# ifconfig vnet1 plumb
# ifconfig vnet0 192.168.1.1/24 up
# ifconfig vnet1 192.168.1.2/24 up
# ifconfig vnet0 group ipmp0
# ifconfig vnet1 group ipmp0
```

- **Oracle Solaris 11 OS:** ipadm 명령을 사용합니다.

net0 및 net1은 vnet0 및 vnet1 각각에 대한 베니티 이름입니다.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i net0 -i net1 ipmp0
# ipadm create-addr -T static -a 192.168.1.1/24 ipmp0/v4addr1
# ipadm create-addr -T static -a 192.168.1.2/24 ipmp0/v4addr2
```

VLAN 태그 지정 사용

Oracle VM Server for SPARC 소프트웨어는 네트워크 기반구조에서 802.1Q VLAN 태깅을 지원합니다.

가상 스위치(*vsw*) 및 가상 네트워크(*vnet*) 장치는 VLAN(Virtual Local-Area Network) 식별자 (ID)를 기준으로 이더넷 패킷 전환을 지원하며, 이더넷 프레임의 필요한 태그 지정 또는 태그 미지정을 처리합니다.

게스트 도메인의 가상 네트워크 장치를 통해 여러 VLAN 인터페이스를 만들 수 있습니다. 가상 네트워크 장치를 통해 VLAN 인터페이스를 만들려면 Oracle Solaris 10 *ifconfig* 명령 또는 Oracle Solaris 11 *dladm* 및 *ipadm* 명령을 사용하십시오. 만드는 방법은 다른 물리적 네트워크 장치를 통해 VLAN 인터페이스를 구성할 때 사용한 방법과 같습니다. Oracle VM Server for SPARC 환경에서는 추가로 *ldm* 명령을 사용하여 *vsw* 또는 *vnet* 가상 네트워크 장치에 VLAN을 지정해야 합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

마찬가지로, Oracle Solaris 10 서비스 도메인에서 가상 스위치 장치를 통해 VLAN 인터페이스를 구성할 수 있습니다. VLAN ID 2 ~ 4094가 유효합니다. VLAN ID 1은 `default-vlan-id`로 예약되어 있습니다. Oracle Solaris 11 서비스 도메인에는 가상 스위치에 VLAN ID를 구성할 필요가 없습니다.

게스트 도메인에서 가상 네트워크 장치를 만드는 경우 *ldm add-vnet* 명령에 *pvid=* 및 *vid=* 인수를 사용하여 포트 VLAN ID 및 0개 이상의 VLAN ID를 지정하는 방식으로 필요한 VLAN에 해당 가상 네트워크 장치를 지정해야 합니다. 이렇게 하면 Oracle VM Server for SPARC 네트워크에서 여러 개의 VLAN을 지원하고 네트워크에서 MAC 주소와 VLAN ID를 모두 사용하여 패킷을 전환하는 가상 스위치가 구성됩니다.

Oracle Solaris 10 서비스 도메인에서 사용되는 모든 VLAN은 *vsw* 장치에 구성되어야 합니다. *ldm add-vsw* 또는 *ldm set-vsw* 명령을 사용하여 *pvid* 및 *vid* 등록 정보 값을 지정합니다.

ldm set-vnet 또는 *ldm set-vsw* 명령을 사용하여 장치가 속한 VLAN을 변경할 수 있습니다.

포트 VLAN ID

PVID(포트 VLAN ID)는 가상 네트워크 장치가 태그 미지정 모드에서 속해야 하는 VLAN을 지정합니다. 이 경우 *vsw* 장치는 PVID로 지정된 VLAN을 통해 *vnet* 장치에 프레임의 필요한 태그 지정 또는 태그 미지정을 제공합니다. 태그가 지정되지 않은 가상 네트워크의 아웃바운드 프레임은 가상 스위치에 의해 PVID로 태그 지정됩니다. 이 PVID로 태그 지정된 인바운드 프레임은 *vnet* 장치로 전송되기 전에 가상 스위치에 의해 태그 지정이 취소됩니다. 따라서 가상 네트워크에 PVID를 지정하면 암시적으로 가상 스위치의 해당 가상 네트워크 포트가 PVID로 지정된 VLAN에 대해 태그 미지정으로 표시됨을 의미합니다. 가상 네트워크 장치에 대해 PVID를 하나만 사용할 수 있습니다.

해당 가상 네트워크 인터페이스가 VLAN ID 없이 해당 장치 인스턴스만 사용하여 구성된 경우 암시적으로 가상 네트워크의 PVID로 지정된 VLAN에 인터페이스가 지정됩니다.

예를 들어, 다음 명령 중 하나를 사용하여 가상 네트워크 인스턴스 `0`을 만들고 `vnet`에 대한 `pvid=` 인수를 `10`으로 지정한 경우 `vnet0` 인터페이스는 암시적으로 VLAN 10에 속하는 것으로 지정됩니다. 다음 명령은 Oracle Solaris 10에 해당하는 `vnet0` 인터페이스 이름을 보여줍니다. Oracle Solaris 11의 경우 `net0`과 같은 일반 이름을 대신 사용하십시오.

- **Oracle Solaris 10 OS:** `ifconfig` 명령을 사용합니다.

```
# ifconfig vnet0 plumb
```

- **Oracle Solaris 11 OS:** `ipadm` 명령을 사용합니다.

```
# ipadm create-ip net0
```

VLAN ID

VID(VID ID)는 태그 지정 모드에서 가상 네트워크 장치 또는 가상 스위치가 속해 있어야 하는 VLAN을 지정합니다. 가상 네트워크 장치는 VID로 지정된 VLAN을 통해 태그 지정된 프레임을 보내고 받습니다. 가상 스위치는 지정된 VID로 태그 지정된 프레임을 가상 네트워크 장치와 외부 네트워크 간에 전달합니다.

VLAN 지정 및 사용

다음 작업에 사용되는 예제 장치는 도메인에서 인스턴스 번호 `0`을 사용합니다. 다음 서브넷에 VLAN이 매핑됩니다.

- VLAN 20 서브넷 192.168.1.0 (넷마스크: 255.255.255.0)
- VLAN 21 서브넷 192.168.2.0 (넷마스크: 255.255.255.0)
- VLAN 22 서브넷 192.168.3.0 (넷마스크: 255.255.255.0)

▼ Oracle Solaris 11 서비스 도메인에서 VLAN을 지정하고 사용하는 방법

1. 가상 스위치(`vsw`)를 지정합니다.

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

2. 서비스 도메인에 VLAN 인터페이스를 만듭니다.

Oracle Solaris 11.1 OS보다 오래된 Oracle Solaris 11 OS를 실행 중인 경우에만 `ipadm create-addr` 명령의 `-T static` 옵션이 필요합니다. Oracle Solaris 11 OS부터 `-T static`이 기본 동작입니다.

```
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.169.2.100/24 net0
# dladm create-vlan -l net0 -v 20 vlan20
# ipadm create-ip vlan20
# ipadm create-addr -T static -a 192.168.1.100/24 vlan20
```

Oracle Solaris 11 OS에서 VLAN 인터페이스를 구성하는 방법에 대한 자세한 내용은 [Managing Network Datalinks in Oracle Solaris 11.3](#)의 3 장, “Configuring Virtual Networks by Using Virtual Local Area Networks”을 참조하십시오.

▼ Oracle Solaris 10 서비스 도메인에서 VLAN을 지정하고 사용하는 방법

1. 두 개의 VLAN에 가상 스위치(vsw)를 지정합니다.

예를 들어 VLAN 21은 태그 미지정으로 구성하고 VLAN 20은 태그 지정으로 구성합니다. 서비스 도메인은 VLAN ID 22에 액세스하도록 구성되지 않았습니다.

```
primary# ldm add-vsw net-dev=nxge0 pvid=21 vid=20 primary-vsw0 primary
```

2. 서비스 도메인에 VLAN 인터페이스를 만듭니다.

```
# ifconfig vsw0 plumb
# ifconfig vsw0 192.168.2.100 netmask 0xfffff00 broadcast + up
# ifconfig vsw20000 plumb
# ifconfig vsw20000 192.168.1.100 netmask 0xfffff00 broadcast + up
```

▼ Oracle Solaris 11 게스트 도메인에서 VLAN을 지정하고 사용하는 방법

이 작업을 완료한 후 `ldom1` 게스트 도메인은 `primary` 서비스 도메인과 통신하고, 외부 태그가 지정된 VLAN ID 21과 IP 주소 192.168.2.0/24를 사용하는 원격/외부 시스템과 통신할 수 있습니다. 또한 `ldom1` 게스트 도메인은 서비스 도메인은 물론, 태그가 지정된 VLAN ID 20과 IP 주소 192.168.1.0/24를 사용하는 외부 시스템과 통신할 수 있습니다. `ldom1` 게스트 도메인은 외부 시스템과 통신할 수 있지만, VLAN 22와 IP 주소 192.168.3.0/24를 사용하는 서비스 도메인과는 통신할 수 없습니다.

1. 두 개의 VLAN에 가상 네트워크(vnet)를 지정합니다.

예를 들어 VLAN 21은 태그 미지정으로 구성하고 VLAN 20은 태그 지정으로 구성합니다.

```
primary# ldm add-vnet pvid=21 vid=20,22 vnet0 primary-vsw0 ldom1
ldom1# ipadm create-ip net0
ldom1# ipadm create-addr -t 192.168.2.101/24 net0
```

2. 게스트 도메인에 VLAN 인터페이스를 만듭니다.

```
ldom1# dladm create-vlan -l net0 -v 20 vlan20
ldom1# ipadm create-ip vlan20
```

```
ldom1# ipadm create-addr -t 192.168.1.101/24 v1an20

ldom1# dladm create-vlan -l net0 -v 22 v1an22
ldom1# ipadm create-ip v1an22
ldom1# ipadm create-addr -t 192.168.3.101/24 v1an22
```

▼ Oracle Solaris 10 게스트 도메인에서 VLAN을 지정하고 사용하는 방법

이 작업을 완료한 후 `ldom1` 게스트 도메인은 `primary` 서비스 도메인과 통신하고, 외부 태그가 지정된 VLAN ID 21과 IP 주소 192.168.2.0/24를 사용하는 원격/외부 시스템과 통신할 수 있습니다. 또한 `ldom1` 게스트 도메인은 서비스 도메인은 물론, 태그가 지정된 VLAN ID 20과 IP 주소 192.168.1.0/24를 사용하는 외부 시스템과 통신할 수 있습니다. `ldom1` 게스트 도메인은 외부 시스템과 통신할 수 있지만, VLAN 22와 IP 주소 192.168.3.0/24를 사용하는 서비스 도메인과는 통신할 수 없습니다.

1. 두 개의 VLAN에 가상 네트워크(vnet)를 지정합니다.

예를 들어 VLAN 21은 태그 미지정으로 구성하고 VLAN 20은 태그 지정으로 구성합니다.

```
primary# ldm add-vnet pvid=21 vid=20,22 vnet0 primary-vswo ldom1
ldom1# ifconfig vnet0 plumb
ldom1# ifconfig vnet0 192.168.2.101 netmask 0xffffffff00 broadcast + up
```

2. 게스트 도메인에 VLAN 인터페이스를 만듭니다.

```
ldom1# ifconfig vnet20000 plumb
ldom1# ifconfig vnet20000 192.168.1.102 netmask 0xffffffff00 broadcast + up
ldom1# ifconfig vnet22000 plumb
ldom1# ifconfig vnet22000 192.168.3.102 netmask 0xffffffff00 broadcast + up
```

▼ 설치 서버가 VLAN에 있을 때 게스트 도메인을 설치하는 방법

설치 서버가 VLAN에 있을 때 Oracle Solaris JumpStart 기능을 사용하여 네트워크를 통해 게스트 도메인을 설치하는 경우 주의하십시오. 이 기능은 Oracle Solaris 10 시스템에서만 지원됩니다.

Oracle Solaris JumpStart 기능을 사용하여 게스트 도메인을 설치하는 자세한 방법은 [Oracle Solaris 10 게스트 도메인에서 Oracle Solaris JumpStart 기능을 사용하는 방법 \[50\]](#)을 참조하십시오.

1. 태그 미지정 모드에서 네트워크 장치를 구성합니다.

예를 들어 설치 서버가 VLAN 21에 있는 경우, 가상 네트워크를 처음에 다음과 같이 구성합니다.

```
primary# ldm add-vnet pvid=21 vnet01 primary-vsw0 ldom1
```

해당 가상 네트워크 장치에 대해서는 태그 지정된 VLAN(vid)을 구성하지 마십시오. OBP (OpenBoot PROM)에서 VLAN을 인식하지 못해 VLAN 태그 지정 네트워크 패킷을 처리할 수 없기 때문입니다.

2. 설치가 완료되고 **Oracle Solaris OS**가 부트되면 태그 지정 모드에서 가상 네트워크를 구성합니다.

```
primary# ldm set-vnet pvid= vid=21, 22, 23 vnet01 primary-vsw0 ldom1
```

이제 태그 지정 모드에서 가상 네트워크 장치를 다른 VLAN에 추가할 수 있습니다.

전용 VLAN 사용

PVLAN(전용 VLAN) 방식을 통해 일반 VLAN을 하위 VLAN으로 분할하여 네트워크 트래픽을 격리할 수 있습니다. PVLAN 방식은 [RFC 5517](http://tools.ietf.org/html/rfc5517) (<http://tools.ietf.org/html/rfc5517>)에 정의되어 있습니다. 대개 일반 VLAN은 단일 브로드캐스트 도메인이지만, PVLAN 등록 정보로 구성된 경우 기존 계층 3 구성을 유지하면서 단일 브로드캐스트 도메인이 보다 작은 브로드캐스트 하위 도메인으로 분할됩니다. PVLAN을 구성하는 경우 일반 VLAN을 주 VLAN이라고 하며 하위 VLAN을 보조 VLAN이라고 합니다.

2개의 가상 네트워크가 물리적 링크에서 동일한 VLAN ID를 사용하는 경우 2개의 가상 네트워크 간에 모든 브로드캐스트 트래픽이 전달됩니다. 하지만 PVLAN 등록 정보를 사용하는 가상 네트워크를 만드는 경우 특정 상황에서는 패킷 전달 동작이 적용되지 않을 수도 있습니다.

다음 표에서는 격리된 PVLAN 및 커뮤니티 PVLAN에 대한 브로드캐스트 패킷 전달 규칙을 보여줍니다.

표 2 브로드캐스트 패킷 전달 규칙

PVLAN 유형	격리	커뮤니티 A	커뮤니티 B
격리	아니오	아니오	아니오
커뮤니티 A	아니오	예	아니오
커뮤니티 B	아니오	아니오	예

예를 들어, vnet0 및 vnet1 가상 네트워크가 모두 net0 네트워크에 격리된 경우 net0은 두 가상 네트워크 간에 브로드캐스트 트래픽을 전달하지 않습니다. 그러나 net0 네트워크가 격리된 VLAN에서 트래픽을 수신하면 VLAN과 관련된 격리된 포트로 트래픽이 전달되지 않습니다. 이 상황은 격리된 가상 네트워크가 주 VLAN의 트래픽만 수용하기 때문에 발생합니다.

inter-vnet-link 기능은 격리된 PVLAN 및 커뮤니티 PVLAN의 통신 제한을 지원합니다. Inter-vnet-link는 격리된 PVLAN에 대해서는 사용 안함으로 설정되고 커뮤니티 PVLAN의 동일한

커뮤니티에 있는 가상 네트워크에 대해서만 사용으로 설정됩니다. 커뮤니티 외부의 다른 가상 네트워크에서 들어오는 직접 트래픽은 허용되지 않습니다.

주 - 대상 서비스 도메인이 PVLAN 기능을 지원하지 않으면 PVLAN 용도로 구성된 게스트 도메인의 마이그레이션을 실패할 수 있습니다.

PVLAN 요구사항

`ldm add-vnet` 및 `ldm set-vnet` 명령을 사용하여 PVLAN을 구성할 수 있습니다. 이 명령을 사용하여 `pvlan` 등록 정보를 설정합니다. PVLAN을 성공적으로 구성하려면 `pvid` 등록 정보도 지정해야 합니다.

이 기능은 최소한 Oracle Solaris 11.2 SRU 4 OS 이상이 필요합니다.

PVLAN을 구성하려면 다음 정보를 지정해야 합니다.

- **주 VLAN ID.** 주 VLAN ID는 단일 가상 네트워크 장치에 대한 PVLAN을 구성하는 데 사용되는 PVID(포트 VLAN ID)입니다. 이 구성을 수행하면 게스트 도메인이 VLAN 패킷을 수신합니다. PVLAN으로 VID를 구성할 수는 없습니다. 이 값은 `pvid` 등록 정보로 표현됩니다.
- **보조 VLAN ID.** 특정 VLAN이 PVLAN 기능을 제공하는 데 보조 VLAN ID를 사용합니다. 이 정보는 `pvlan` 값의 `secondary-vid` 부분으로 지정합니다. `secondary-vid`는 1-4094 범위의 정수 값입니다. 주 VLAN에는 여러 보조 VLAN이 포함될 수 있습니다. 단, 다음 제한 사항이 적용됩니다.
 - 주 VLAN ID와 보조 VLAN ID가 기본 VLAN ID와 동일할 수 없습니다.
 - 격리된 PVLAN 유형 및 커뮤니티 PVLAN 유형에 대한 주 VLAN ID와 보조 VLAN ID의 값은 동일할 수 없습니다.
 - 각 주 VLAN에는 격리된 PVLAN을 하나만 구성할 수 있습니다. 따라서 동일한 주 VLAN ID를 사용하는 2개의 격리된 PVLAN을 만들 수 없습니다.
 - 주 VLAN에는 여러 커뮤니티 VLAN이 포함될 수 있습니다. 단, 다음 제한 사항이 적용됩니다.
 - 주 VLAN ID를 보조 VLAN ID로 사용하여 다른 커뮤니티 PVLAN을 만들 수 없습니다. 예를 들어, 주 VLAN ID가 3이며 보조 VLAN ID가 100인 커뮤니티 PVLAN이 있을 경우 보조 VLAN ID가 3인 다른 커뮤니티 PVLAN을 만들 수 없습니다.
 - 보조 VLAN ID를 주 VLAN ID로 사용하여 커뮤니티 PVLAN을 만들 수 없습니다. 예를 들어, 주 VLAN ID가 3이며 보조 VLAN ID가 100인 커뮤니티 PVLAN이 있을 경우 주 VLAN ID가 100인 다른 커뮤니티 PVLAN을 만들 수 없습니다.
 - 보조 VLAN ID를 일반 가상 네트워크 또는 VNIC에 대한 VLAN ID로 사용할 수 없습니다.



주의 - Logical Domains Manager는 특정 가상 스위치의 가상 네트워크 구성만 검증할 수 있습니다. 동일한 백엔드 장치에서 Oracle Solaris VNIC에 대해 PVLAN 구성을 설정하는 경우 모든 VNIC와 가상 네트워크에서 동일한 요구사항이 충족되어야 합니다.

- **PVLAN 유형.** 이 정보는 `pvlan` 값의 `pvlan-type` 부분으로 지정합니다. `pvlan-type`은 다음 값 중 하나입니다.
 - `isolated`. 격리된 PVLAN과 연관되어 있는 포트가 모든 피어 가상 네트워크 및 백엔드 네트워크 장치의 Oracle Solaris 가상 NIC에서 격리됩니다. 패킷은 PVLAN에 대해 지정된 값을 기반으로 외부 네트워크에만 연결됩니다.
 - `community`. 커뮤니티 PVLAN과 연관되어 있는 포트가 동일한 커뮤니티 PVLAN에 있지만 기타 모든 포트에서 격리된 다른 포트와 통신할 수 있습니다. 패킷은 PVLAN에 대해 지정된 값을 기반으로 외부 네트워크에 연결됩니다.

PVLAN 구성

이 절에는 PVLAN을 만들고 PVLAN에 대한 정보를 나열하는 방법을 설명하는 작업이 포함됩니다.

PVLAN 만들기

`ldm add-vnet` 또는 `ldm set-vnet` 명령을 통해 `pvlan` 등록 정보 값을 설정하여 PVLAN을 구성할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 명령을 사용하여 PVLAN을 만들거나 제거할 수 있습니다.

- `ldm add-vnet`를 사용하여 PVLAN을 만듭니다.

```
ldm add-vnet pvid=port-VLAN-ID pvlan=secondary-vid, pvlan-type \  
if-name vswitch-name domain-name
```

다음 명령은 주 `vlan-id`가 4이며 보조 `vlan-id`가 200이고 `pvlan-type`이 `isolated`인 PVLAN으로 가상 네트워크를 만드는 방법을 보여줍니다.

```
primary# ldm add-vnet pvid=4 pvlan=200,isolated vnet1 primary-vsw0 ldg1
```

- `ldm set-vnet`를 사용하여 PVLAN을 만듭니다.

```
ldm set-vnet pvid=port-VLAN-ID pvlan=secondary-vid, pvlan-type if-name domain-name
```

다음 명령은 주 `vlan-id`가 3이며 보조 `vlan-id`가 300이고 `pvlan-type`이 `community`인 PVLAN으로 가상 네트워크를 만드는 방법을 보여줍니다.

```
primary# ldm set-vnet pvid=3 pvlan=300,community vnet2 ldg1
```

- `ldm set-vnet`를 사용하여 PVLAN을 제거합니다.

```
ldm set-vnet pvlan= if-name domain-name
```

다음 명령은 `vnet0` 가상 네트워크에 대한 PVLAN 구성을 제거합니다. `vnet0` 가상 네트워크를 일반 VLAN ID로 되돌리려면 먼저 PVLAN ID를 제거해야 합니다.

```
primary# ldm set-vnet pvlan= vnet0 ldg1
```

PVLAN 정보 보기

다양한 Logical Domains Manager 나열 하위 명령을 사용하여 PVLAN에 대한 정보를 볼 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 명령을 사용하여 PVLAN 정보를 볼 수 있습니다.

- `ldm list-domain -o network`를 사용하여 PVLAN 정보를 나열합니다.

```
ldm list-domain [-e] [-l] -o network [-p] [domain-name...]
```

다음 예는 `ldm list-domain -o network` 명령을 사용하여 `ldg1` 도메인에서 PVLAN 구성에 대한 정보를 보여 줍니다.

- 다음 `ldm list-domain` 명령은 `ldg1` 도메인의 PVLAN 구성에 대한 정보를 보여줍니다.

```
primary# ldm list-domain -o network ldg1
NAME
ldg1

MAC
00:14:4f:fb:22:79

NETWORK
NAME          SERVICE          MACADDRESS PVID|PVLAN|VIDs
----          -
vnet0        primary-vsw0@primary 00:14:4f:f8:6e:d9 2|300,community|--
DEVICE              :network@0      ID      :0
LINKPROP            :--              MTU     :1500
MAXBW               :--              MODE    :--
CUSTOM              :disable
PRIORITY            :--              COS     :--
PROTECTION          :--
```

- 다음 `ldm list-domain` 명령은 `ldg1` 도메인에 대해 구문 분석이 가능한 형식으로 PVLAN 구성 정보를 보여줍니다.

```
primary# ldm list-domain -o network -p ldg1
VERSION 1.19
```

```
DOMAIN|name=ldg1|
MAC|mac-addr=00:14:4f:fb:22:79
VNET|name=vnet0|dev=network@0|service=primary-vsw0@primary|mac-addr=00:14:4f:f8:
6e:d9|mode=|pvid=2|vid=|mtu=1500|linkprop=|id=0|alt-mac-addr=|maxbw=|pvlan=300,
community|protection=|priority=|cos=|custom=disable|max-mac-addr=|max-vlans=
```

- `ldm list-bindings`를 사용하여 PVLAN 정보를 나열합니다.

```
ldm list-bindings [-e] [-p] [domain-name...]
```

다음 예에서는 `ldm list-bindings network` 명령을 사용하여 `ldg1` 도메인의 PVLAN 구성에 대한 정보를 보여줍니다.

- 다음 `ldm list-bindings` 명령은 `ldg1` 도메인의 PVLAN 구성에 대한 정보를 보여줍니다.

```
primary# ldm list-bindings -o network ldg1
NAME
ldg1

MAC
00:14:4f:fb:22:79

NETWORK
NAME          SERVICE          MACADDRESS PVID|PVLAN|VIDs
----          -
vnet0         primary-vsw0@primary 00:14:4f:f8:6e:d9 2|300,community|--

PEER          MACADDRESS          PVID|PVLAN|VIDs
----          -
primary-vsw0@primary 00:14:4f:f9:08:28 1|--|--
```

- 다음 `ldm list-bindings` 명령은 `ldg1` 도메인에 대해 구문 분석이 가능한 형식으로 PVLAN 구성 정보를 보여줍니다.

```
primary# ldm list-bindings -o network -p ldg1
VERSION 1.19
DOMAIN|name=ldg1|
MAC|mac-addr=00:14:4f:fb:22:79
VNET|name=vnet0|dev=network@0|service=primary-vsw0@primary|mac-addr=00:14:4f:f8:
6e:d9|mode=|pvid=2|vid=|mtu=1500|linkprop=|id=0|alt-mac-addr=|maxbw=|pvlan=300,
community|protection=|priority=|cos=|custom=disable|max-mac-addr=|max-vlans=
|peer=primary-vsw0@primary|mac-addr=00:14:4f:f9:08:28|mode=|pvid=1|vid=|mtu=1500|
maxbw=
```

- `ldm list-constraints`를 사용하여 PVLAN 정보를 나열합니다.

```
ldm list-constraints [-x] [domain-name...]
```

다음은 `ldm list-constraints` 명령을 실행하여 생성되는 출력을 보여줍니다.

```
primary# ldm list-constraints -x ldg1
```

```

...
<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>network</rasd:OtherResourceType>
    <rasd:Address>auto-allocated</rasd:Address>
    <gprop:GenericProperty key="vnet_name">vnet0</gprop:GenericProperty>
    <gprop:GenericProperty key="service_name">primary-vsw0</gprop:GenericProperty>
    <gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
    <gprop:GenericProperty key="vid">3</gprop:GenericProperty>
    <gprop:GenericProperty key="pvlan">200,community</gprop:GenericProperty>
    <gprop:GenericProperty key="maxbw">1700000000</gprop:GenericProperty>
    <gprop:GenericProperty key="device">network@0</gprop:GenericProperty>
    <gprop:GenericProperty key="id">0</gprop:GenericProperty>
  </Item>

```

패킷 처리량 성능 조정

ldm add-vnet 및 ldm set-vnet 명령을 통해 다음 데이터 링크 등록 정보 값을 설정하여 패킷 처리량 성능을 조정할 수 있습니다.

priority	CPU 패킷 처리 우선순위를 지정합니다.
cos	IEEE 802.1p 링크 서비스 등급을 링크에 지정합니다.
protection	패킷 트래픽 보안 유형을 지정합니다.

유효 및 기본 등록 정보 값에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 46 데이터 링크 패킷 등록 정보 설정 및 보기

다음 예제는 ldm set-vnet 명령을 사용하여 단일 명령에 priority, protection, cos 등록 정보 값을 설정하는 방법을 보여줍니다. ldm add-vnet 명령을 사용하여 지정된 데이터 링크 등록 정보 값을 사용하는 가상 네트워크를 새로 추가할 수도 있습니다.

```

primary# ldm set-vnet allowed-ips=192.168.100.1,192.168.100.2 \
allowed-dhcp-cids=oracle@system1.company.com, \
00:14:4f:fb:22:79,system2,00:14:4f:fb:22:56 cos=7 priority=high \
protection=restricted,mac-nospoof,ip-nospoof,dhcp-nospoof vnet3_ldg3 ldg3

```

ldm list -o network 명령은 ldg3 도메인에서 이전 ldm set-vnet 명령으로 설정된 데이터 링크 등록 정보 값을 보여줍니다. protection 값은 192.168.100.1,192.168.100.2 MAC 주소에 대해 mac-nospoof, restricted, ip-nospoof이고 system1@company.com,00:14:4f:fb:d3:88,system2,00:14:4f:fb:61:6e에 대해 dhcp-nospoof입니다. priority는 high로 설정되고 서비스 등급(cos)은 7로 설정됩니다.

```

primary# ldm list-domain -o network ldg3
NAME
ldg3

MAC
00:14:4f:f8:5b:12
NAME          SERVICE          MACADDRESS PVID|PVLAN|VIDs
-----
vnet3_ldg3   primary-vsw0@primary 00:14:4f:f8:dd:96 1|--|--
  DEVICE      :network@1        ID       :1
  LINKPROP    :phys-state      MTU      :1500
  MAXBW       :--                MODE     :--
  CUSTOM      :disable
  PRIORITY    :high          COS      :7
  PROTECTION  :mac-nospoof
                restricted
                ip-nospoof
                [192.168.100.1
                192.168.100.2]
                dhcp-nospoof
                [oracle@system1.company.com
                00:14:4f:fb:22:79
                system2
                00:14:4f:fb:22:56]

```

가상 스위치에서 링크 통합 사용

링크 통합을 사용하도록 가상 스위치를 구성할 수 있습니다. 링크 통합은 가상 스위치의 네트워크 장치로 사용되어 물리적 네트워크에 연결합니다. 이 구성을 사용하면 가상 스위치가 IEEE 802.3ad Link Aggregation Standard에서 제공하는 기능을 활용할 수 있습니다. 이러한 기능에는 대역폭 증가, 로드 균형 조정 및 페일오버가 포함됩니다. 링크 통합을 구성하는 방법에 대한 자세한 내용은 [Managing Network Datalinks in Oracle Solaris 11.3](#)의 “Creating a Link Aggregation”를 참조하십시오.

링크 통합을 만든 후에는 가상 스위치에 지정할 수 있습니다. 이와 같이 지정하는 것은 물리적 네트워크 장치를 가상 스위치에 지정하는 것과 비슷합니다. `net-dev` 등록 정보를 설정하려면 `ldm add-vswitch` 또는 `ldm set-vswitch` 명령을 사용하십시오.

링크 통합이 가상 스위치에 지정되면 물리적 네트워크로 들어오거나 물리적 네트워크에서 나가는 트래픽은 통합을 통해 이동합니다. 필요한 균형 조정 또는 페일오버는 기본 통합 프레임워크를 통해 투명하게 처리됩니다. 링크 통합은 게스트 도메인에 있으며 통합을 사용하는 가상 스위치에 바인드된 가상 네트워크(vnet) 장치에 전혀 영향을 주지 않습니다.

주 - 가상 네트워크 장치(vnet 및 vsw)는 링크 통합으로 그룹화할 수 없습니다.

서비스 도메인에서 링크 통합을 사용하도록 구성된 가상 스위치를 만들어 사용할 수 있습니다. [가상 스위치를 주 인터페이스로 구성하는 방법 \[38\]](#)을 참조하십시오.

[그림 23](#) 및 [그림 24](#)은 각각의 물리적 인터페이스 `net0` 및 `net1`과 `nxge0` 및 `nxge1`을 통한 통합 `aggr1`을 사용하여 구성된 가상 스위치를 보여 줍니다.

그림 23 링크 통합을 사용하도록 가상 스위치 구성(Oracle Solaris 11)

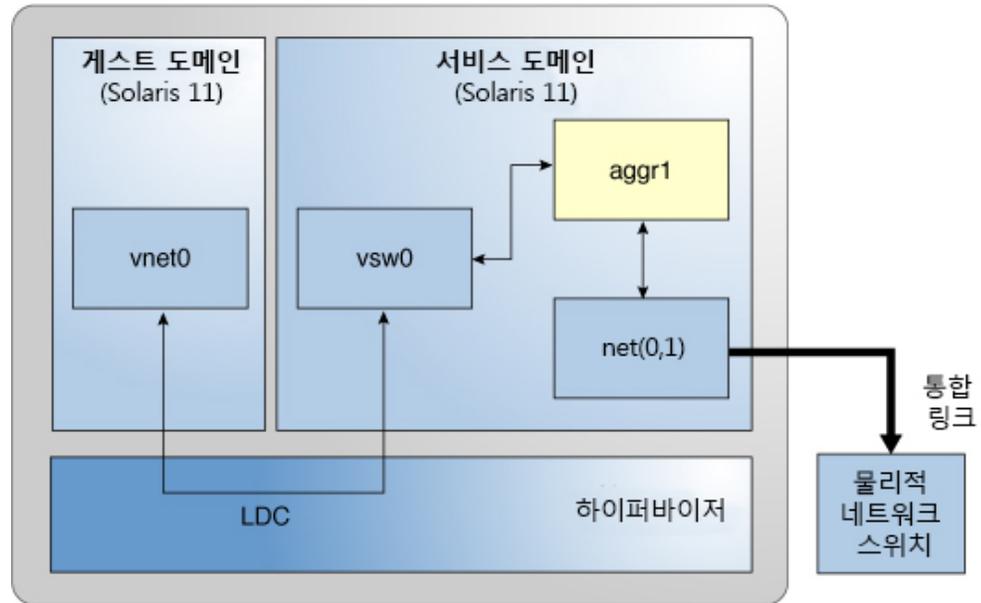
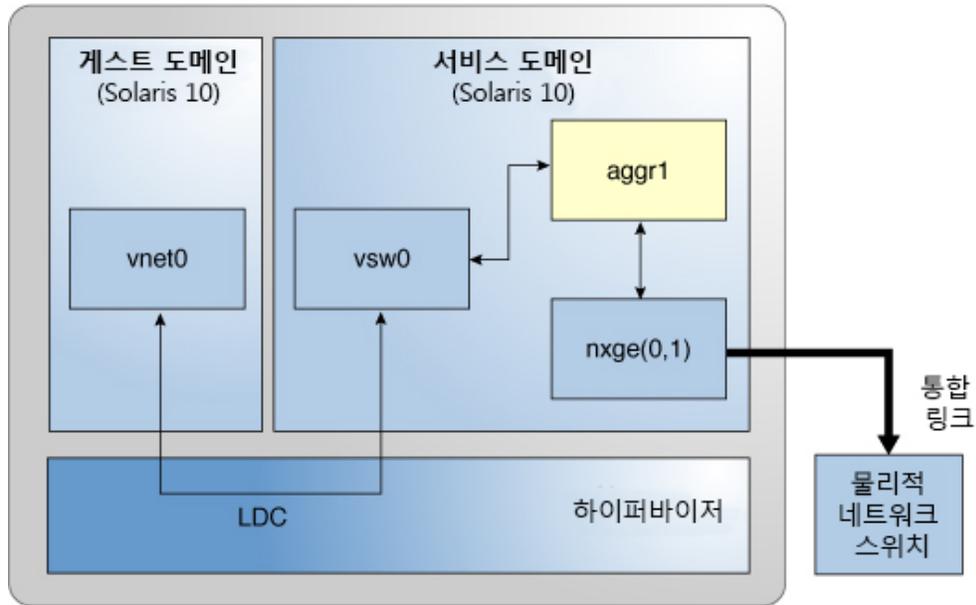


그림 24 링크 통합을 사용하도록 가상 스위치 구성(Oracle Solaris 10)



점보 프레임 구성

Oracle VM Server for SPARC 가상 스위치(vsw) 및 가상 네트워크(vnet) 장치는 이제 페이로드 크기가 1500바이트를 초과하는 이더넷 프레임을 지원할 수 있습니다. 따라서 해당 드라이버는 이제 네트워크 처리량을 늘릴 수 있습니다.

점보 프레임을 사용으로 설정하려면 가상 스위치 장치의 MTU(최대 전송 단위)를 지정하십시오. 이 경우 가상 스위치 장치 및 가상 스위치 장치에 바인드된 모든 가상 네트워크 장치가 지정된 MTU 값을 사용합니다.

가상 네트워크 장치에 대해 필요한 MTU 값이 가상 스위치에서 지원되는 것보다 작아야 할 경우 가상 네트워크 장치에서 직접 MTU 값을 지정할 수 있습니다.

주 - Oracle Solaris 10 5/09 OS에서만 물리적 장치의 MTU가 가상 스위치의 MTU와 일치하도록 구성해야 합니다. 특정 드라이버 구성에 대한 자세한 내용은 Oracle Solaris 참조 설명서의 7D절에서 해당 드라이버와 일치하는 매뉴얼 페이지를 참조하십시오. 예를 들어 Oracle Solaris 10 nxge 드라이버에 대한 정보는 [nxge\(7D\)](#) 매뉴얼 페이지를 참조하십시오.

드문 경우지만, `ldm add-vnet` 또는 `ldm set-vnet` 명령을 사용하여 가상 네트워크 장치의 MTU 값을 가상 스위치의 MTU 값과 다르게 지정해야 할 수 있습니다. 예를 들어 가상 네트워크 장치에 대해 VLAN을 구성하고 가장 큰 VLAN MTU가 가상 스위치의 MTU 값보다 작을 경우 가상 네트워크 장치의 MTU 값을 변경할 수 있습니다. 기본 MTU 값만 사용되는 도메인에는 점보 프레임을 지원하는 `vnet` 드라이버가 필요하지 않을 수 있습니다. 그러나 도메인에서 가상 네트워크 장치가 점보 프레임을 사용하는 가상 스위치에 바인드된 경우, `vnet` 드라이버가 점보 프레임을 지원하는지 확인하십시오.

`ldm set-vnet` 명령을 사용하여 가상 네트워크 장치에서 `mtu` 값을 지정하는 경우 가상 스위치 장치의 MTU 값을 이후에 업데이트하면 업데이트 사항이 해당 가상 네트워크 장치에 전파되지 않습니다. 가상 네트워크 장치를 다시 사용으로 설정하여 가상 스위치 장치에서 MTU 값을 가져오려면 다음 명령을 실행하십시오.

```
primary# ldm set-vnet mtu= vnet-name domain-name
```

컨트롤 도메인에서 Logical Domains Manager는 `ldm set-vsw` 및 `ldm set-vnet` 명령으로 시작된 MTU 값을 지연된 재구성 작업으로 업데이트합니다. 컨트롤 도메인이 아닌 다른 도메인에서 MTU를 업데이트하려면 `ldm set-vsw` 또는 `ldm set-vnet` 명령을 실행하여 MTU 값을 수정하기 전에 도메인을 중지해야 합니다.

▼ 점보 프레임을 사용하도록 가상 네트워크 및 가상 스위치 장치를 구성하는 방법

1. 컨트롤 도메인에 로그인합니다.

2. 관리자로 전환합니다.

Oracle Solaris 11.3의 경우, [Securing Users and Processes in Oracle Solaris 11.3](#)의 1장, "About Using Rights to Control Users and Processes"를 참조하십시오.

3. 가상 네트워크에 사용할 MTU 값을 확인합니다.

MTU 값은 1500 - 16000바이트 범위에서 지정할 수 있습니다. 지정된 MTU는 가상 스위치에 지정된 물리적 네트워크 장치의 MTU와 일치해야 합니다.

4. 가상 스위치 또는 가상 네트워크 장치의 MTU 값을 지정합니다.

다음 중 하나를 수행합니다.

- 해당 MTU를 `mtu` 등록 정보의 값으로 지정하여 서비스 도메인의 새 가상 스위치 장치에 점보 프레임을 사용으로 설정합니다.

```
primary# ldm add-vsw net-dev=device mtu=value vswitch-name ldom
```

이 명령은 가상 스위치를 구성하는 것 이외에도 이 가상 스위치에 바인드될 각 가상 네트워크 장치의 MTU 값을 업데이트합니다.

- 해당 MTU를 `mtu` 등록 정보의 값으로 지정하여 서비스 도메인의 기존 가상 스위치 장치에 점보 프레임을 사용으로 설정합니다.

```
primary# ldm set-vsw net-dev=device mtu=value vswitch-name
```

이 명령은 가상 스위치를 구성하는 것 이외에도 이 가상 스위치에 바인드될 각 가상 네트워크 장치의 MTU 값을 업데이트합니다.

예 47 가상 스위치 및 가상 네트워크 장치에서 점보 프레임 구성

- 다음 예는 MTU 값 9000을 사용하는 가상 스위치 장치를 새로 추가하는 방법을 보여줍니다. MTU 값은 가상 스위치 장치에서 모든 클라이언트 가상 네트워크 장치로 전파됩니다.

먼저, `ldm add-vsw` 명령이 MTU 값으로 9000을 사용하는 가상 스위치 장치 `ldg1-vsw0`을 만듭니다. 네트워크 장치 `net0`의 인스턴스 0이 `net-dev` 등록 정보의 값으로 지정됩니다.

```
primary# ldm add-vsw net-dev=net0 mtu=9000 ldg1-vsw0 ldg1
```

다음으로, `ldm add-vnet` 명령이 클라이언트 가상 네트워크 장치를 이 가상 스위치 `ldg1-vsw0`에 추가합니다. 가상 네트워크 장치의 MTU는 암시적으로 가상 네트워크 장치가 바인드된 가상 스위치에서 지정됩니다. 그 결과 `ldm add-vnet` 명령에 `mtu` 등록 정보의 값을 지정할 필요가 없습니다.

```
primary# ldm add-vnet vnet01 ldg1-vsw0 ldg1
```

실행 중인 Oracle Solaris OS의 버전에 따라 다음을 수행하십시오.

- **Oracle Solaris 11 OS:** `ipadm` 명령을 사용하여 기본 인터페이스의 `mtu` 등록 정보 값을 확인합니다.

```
# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0 mtu ipv4 rw 9000 -- 9000 68-9000
```

`ipadm` 명령이 게스트 도메인 `ldg1`에 가상 네트워크 인터페이스를 만듭니다. `ipadm show-ifprop` 명령 출력은 `mtu` 등록 정보의 값이 9000임을 보여줍니다.

```
ldg1# ipadm create-ip net0
ldg1# ipadm create-addr -T static -a 192.168.1.101/24 net0/ipv4
ldg1# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0 mtu ipv4 rw 9000 -- 9000 68-9000
```

- **Oracle Solaris 10 OS:** `ifconfig` 명령이 서비스 도메인 `ldg1`에 가상 스위치 인터페이스를 만듭니다. `ifconfig vsw0` 명령 출력은 `mtu` 등록 정보의 값이 9000임을 보여줍니다.

```
ldg1# ifconfig vsw0 plumb
ldg1# ifconfig vsw0 192.168.1.100/24 up
ldg1# ifconfig vsw0
vsw0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 5
inet 192.168.1.100 netmask ffffffff broadcast 192.168.1.255
```

```
ether 0:14:4f:fa:0:99
```

ifconfig 명령이 게스트 도메인 ldg1에 가상 네트워크 인터페이스를 만듭니다. ifconfig vnet0 명령 출력은 mtu 등록 정보의 값이 9000임을 보여줍니다.

```
ldg1# ifconfig vnet0 plumb
ldg1# ifconfig vnet0 192.168.1.101/24 up
ldg1# ifconfig vnet0
vnet0: flags=201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS> mtu 9000 index 4
      inet 192.168.1.101 netmask ffffffff0 broadcast 192.168.1.255
      ether 0:14:4f:f9:c4:13
```

- 다음 예는 인터페이스의 MTU를 4000으로 변경하는 방법을 보여줍니다.

인터페이스의 MTU는 Logical Domains Manager에서 지정한 장치의 MTU보다 작은 값으로만 변경할 수 있습니다. 이 방법은 VLAN이 구성되어 있고 각 VLAN 인터페이스에 다른 MTU가 필요한 경우 유용합니다.

- **Oracle Solaris 11 OS:** ipadm 명령을 사용합니다.

```
primary# ipadm set-ifprop -p mtu=4000 net0
primary# ipadm show-ifprop -p mtu net0
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net0    mtu      ipv4  rw   4000    --      9000    68-9000
```

- **Oracle Solaris 10 OS:** ifconfig 명령을 사용합니다.

```
primary# ifconfig vnet0 mtu 4000
primary# ifconfig vnet0
vnet0: flags=1201000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,CoS,FIXEDMTU>
mtu 4000 index 4
      inet 192.168.1.101 netmask ffffffff0 broadcast 192.168.1.255
      ether 0:14:4f:f9:c4:13
```

vnet 및 vsw 드라이버의 이전(점보 미인식) 버전과의 호환성 (Oracle Solaris 10)

주 - 이 절의 내용은 Oracle Solaris 10 OS에만 적용됩니다.

점보 프레임을 지원하는 드라이버는 같은 시스템의 점보 프레임을 지원하지 않는 드라이버와 상호 운용될 수 있습니다. 가상 스위치를 만들 때 점보 프레임 지원이 사용으로 설정되지 않는 한 이 상호 운용성이 지원됩니다.

주 - 가상 스위치와 연관된 게스트 또는 서비스 도메인에서 점보 프레임을 지원하는 Oracle VM Server for SPARC 드라이버를 사용하지 않는 경우 mtu 등록 정보를 설정하지 마십시오.

가상 스위치의 `mtu` 등록 정보를 기본값 1500에서 변경하여 점보 프레임의 사용으로 설정할 수 있습니다. 이 경우 이전 드라이버 버전은 `mtu` 설정을 무시하고 기본값을 계속 사용합니다. `ldm list` 출력에는 기본값이 아닌 사용자가 지정한 MTU 값이 표시됩니다. 기본 MTU보다 큰 프레임은 해당 장치로 전송되지 않고 새 드라이버에 의해 삭제됩니다. 이러한 상황으로 인해 이전 드라이버를 여전히 사용하는 게스트에서 일관되지 않는 네트워크 동작이 발생할 수 있습니다. 이 제한 사항은 클라이언트 게스트 도메인과 서비스 도메인에 모두 적용됩니다.

따라서 점보 프레임이 사용으로 설정된 경우 Oracle VM Server for SPARC 네트워크의 모든 가상 장치가 점보 프레임을 지원하는 새 드라이버를 사용하도록 업그레이드되었는지 확인하십시오. 점보 프레임을 구성하려면 Logical Domains 1.2 이상이 실행 중이어야 합니다.

가상 네트워크에서 가상 NIC 사용

Oracle Solaris 11 OS를 사용하면 가상 네트워크 인터페이스 카드(VNIC), 가상 스위치 및 `etherstub`로 구성되는 가상 네트워크를 정의할 수 있습니다. Oracle Solaris 영역은 운영체제 서비스를 가상화하고 논리적 도메인의 동일한 Oracle Solaris OS 인스턴스 내에서 실행 중인 응용 프로그램에 대해 격리되고 안전한 환경을 제공합니다.

전역 영역의 네트워크 등록 정보를 상속하지만 고유의 네트워크 주소나 다른 등록 정보를 설정할 수 없었던 Oracle Solaris 10 “공유 IP” 영역 모델이 Oracle Solaris 11에서 향상되었습니다. 이제, 가상 네트워크 장치와 함께 영역을 사용하여 여러 개의 격리된 가상 NIC를 구성하고 각 가상 네트워크와 영역을 연관시키며 격리, 연결성, QoS(서비스 품질)에 대한 규칙을 설정할 수 있습니다.

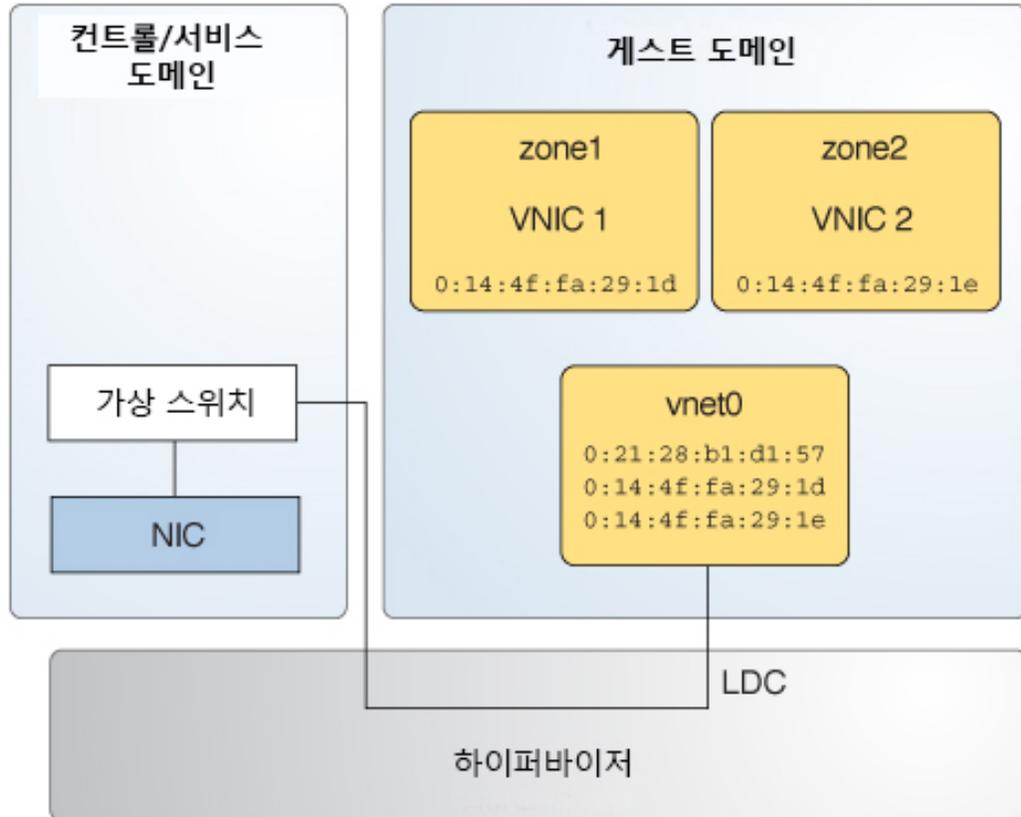
자세한 내용은 [Oracle Solaris 11.3 Information Library \(http://docs.oracle.com/cd/E53394_01/\)](http://docs.oracle.com/cd/E53394_01/)의 네트워킹 문서를 참조하십시오.

논리적 도메인의 가상 네트워크 장치는 여러 개의 Oracle Solaris 11 가상 NIC를 지원할 수 있습니다. 가상 네트워크 장치는 각 가상 NIC에 하나씩 여러 MAC 주소를 지원하도록 구성해야 합니다. 논리적 도메인의 Oracle Solaris 영역은 가상 NIC에 연결합니다.

그림 25는 Oracle Solaris OS에 대해 `vnet1`이라고 하는 단일 가상 네트워크 장치를 제공하는 논리적 도메인 `domain1`을 보여 줍니다. 이 가상 네트워크 장치는 여러 개의 Oracle Solaris 11 가상 네트워크 장치를 호스트할 수 있으며, 각각 고유의 MAC 주소를 가지고 영역에 개별적으로 지정할 수 있습니다.

`domain1` 도메인 내에는 Oracle Solaris 11 영역(`zone1` 및 `zone2`)이 있습니다. 각 영역은 `vnet1` 가상 네트워크 장치에 기반한 가상 NIC에 의해 네트워크에 연결됩니다.

그림 25 가상 네트워크 장치의 가상 NIC



다음 절에서는 가상 네트워크 장치에 가상 NIC를 구성하고 이 가상 NIC로 도메인에 영역을 만드는 방법을 설명합니다.

- “가상 네트워크 장치에 가상 NIC 구성” [270]
- “도메인에 Oracle Solaris 11 영역 만들기” [270]

이더넷 SR-IOV 가상 기능에서 가상 NIC를 사용하는 방법은 다음 절을 참조하십시오.

- “이더넷 가상 기능 만들기” [86]
- “이더넷 SR-IOV 가상 기능 수정” [93]
- “SR-IOV 가상 기능에 VNIC 만들기” [99]

가상 네트워크 장치에 가상 NIC 구성

가상 네트워크 장치에 가상 NIC를 구성하려면 컨트롤 도메인에서 최소한 Oracle Solaris 11.1 SRU 4 OS를 실행하고 게스트 도메인에서 최소한 Oracle Solaris 11.1 OS를 실행해야 합니다.

여러 개의 MAC 주소를 호스트하도록 가상 네트워크 장치를 구성하려면 `ldm add-vnet` 또는 `ldm set-vnet` 명령을 사용하여 `alt-mac-addr`s 등록 정보에 대해 하나 이상의 콤마로 구분된 값을 지정합니다. 유효한 값은 옥테트 MAC 주소와 `auto`입니다. `auto` 값은 시스템이 MAC 주소를 생성함을 나타냅니다.

예를 들어, 다음 방법 중 하나로 가상 네트워크 장치에 3개의 시스템 생성 대체 MAC 주소를 지정할 수 있습니다.

- `ldm add-vnet` 명령 사용. 다음 `ldm add-vnet` 명령은 `domain1` 도메인에 `vnet1` 가상 네트워크 장치를 만들고 이 장치에 3개의 시스템 생성 MAC 주소를 사용 가능하게 합니다.

```
primary# ldm add-vnet alt-mac-addr=auto,auto,auto vnet1 primary-vsw0 domain1
```

- `ldm add-vnet` 및 `ldm set-vnet` 명령의 조합 사용. 다음 `ldm add-vnet` 및 `ldm set-vnet` 명령은 가상 네트워크 장치를 만들고, 이어서 기존 가상 네트워크 장치에 추가 MAC 주소를 지정하는 방법을 보여줍니다.

첫번째 명령은 `ldm add-vnet` 명령을 사용하여 `domain1` 도메인에 `vnet1` 가상 네트워크 장치를 만듭니다. 두번째 명령은 `ldm set-vnet` 명령을 사용하여 `vnet1` 가상 네트워크 장치에 3개의 시스템 생성 MAC 주소를 사용 가능하게 합니다.

```
primary# ldm add-vnet vnet1 primary-vsw0 domain1
primary# ldm set-vnet alt-mac-addr=auto,auto,auto vnet1 domain1
```

도메인에 Oracle Solaris 11 영역 만들기

“가상 네트워크 장치에 가상 NIC 구성” [270]에서 가상 NIC를 만든 후에 사용 가능한 MAC 주소와 연관된 영역을 만듭니다. Oracle Solaris 영역에 대한 자세한 내용은 [Creating and Using Oracle Solaris Zones](#)을 참조하십시오.

`zonecfg` 명령을 사용하여 영역에 사용할 MAC 주소를 지정합니다.

```
zonecfg:zone-name> set mac-address=[MAC-address, auto]
```

`auto` 값을 지정하여 사용 가능한 MAC 주소 중 하나를 자동으로 선택하거나, `ldm set-vnet` 명령으로 만든 특정한 대체 MAC 주소를 제공할 수 있습니다.

신뢰할 수 있는 가상 네트워크 사용

신뢰할 수 있는 가상 네트워크 기능은 권한을 신뢰할 수 있는 게스트 도메인으로 확장하여 사용자정의 대체 MAC 주소 및 대체 VLAN ID를 vnet 장치로 동적으로 지정합니다. 이러한 MAC 주소 및 VLAN ID는 가상 장치를 구성하는 데 사용됩니다. 이 기능을 도입하기 전에는 Logical Domains Manager에서만 해당 지정을 수행할 수 있습니다. 또한, 대체 MAC 주소 지정을 수행하려면 가상 네트워크 장치를 호스트하는 도메인이 바인드된 상태여야 합니다. 이 기능을 사용하면 가상 네트워크 장치 위에 vNIC 및 VLAN과 같은 가상 장치를 동적으로 만들 수 있습니다.

vnet 장치에서 신뢰할 수 있는 가상 네트워크 기능을 사용하려면 Logical Domains Manager를 사용하여 신뢰할 수 있는 모드에서 장치를 만들거나 구성해야 합니다. 기본적으로 vnet 장치는 신뢰할 수 있는 모드가 사용 안함으로 설정된 상태로 생성됩니다.

신뢰할 수 있는 가상 네트워크 기능은 라이브 마이그레이션, 서비스 도메인 재부트, 여러 서비스 도메인 기능을 효과적으로 지원합니다.

신뢰할 수 있는 가상 네트워크 요구사항 및 제한 사항

ldm add-vnet 및 ldm set-vnet 명령을 사용하여 custom=enable 등록 정보를 설정하는 방식으로 신뢰할 수 있는 가상 네트워크를 구성할 수 있습니다. 사용자정의 MAC 주소 및 VLAN 수가 지정된 가상 네트워크 장치에 대해 제한되어 있음을 확인하려면 custom/max-mac-addr 및 custom/max-vlans 등록 정보에 대한 값을 제공해야 합니다. 두 가지 등록 정보 값은 기본적으로 4096으로 설정됩니다.

신뢰할 수 있는 가상 네트워크 기능을 사용하려면 Oracle Solaris 11.3 SRU 8 OS 이상이 필요합니다.

사용자정의 가상 네트워크 장치가 있는 게스트 도메인과 해당 가상 스위치 장치가 있는 서비스 도메인 모두 지원되는 시스템 펌웨어의 최신 레벨이 필요합니다.

신뢰할 수 있는 가상 네트워크를 구성하려면 다음 정보를 지정해야 합니다.

- custom – 신뢰할 수 있는 가상 네트워크 기능을 사용 또는 사용 안함으로 설정합니다. 이 기능을 사용하면 신뢰할 수 있는 엔티티가 사용자정의 대체 VLAN ID 및 사용자정의 대체 MAC 주소를 동적으로 추가할 수 있습니다.
- custom/max-mac-addr – 특정 신뢰할 수 있는 가상 네트워크 장치에 구성할 최대 사용자정의 대체 MAC 주소 수를 지정합니다.
- custom/max-vlans – 특정 신뢰할 수 있는 가상 네트워크 장치에 구성할 최대 사용자정의 대체 VLAN ID 수를 지정합니다.

다음 제한 사항은 신뢰할 수 있는 가상 네트워크 기능에 적용됩니다.

- Logical Domains Manager를 사용하여 지정된 신뢰할 수 있는 가상 네트워크에서 대체 MAC 주소 또는 VLAN ID를 구성할 수 없습니다.

- 사용자정의 또는 기존 대체 MAC 주소를 수정하려면 도메인이 바인드된 상태여야 합니다.
- `custom/max-mac-addr`s 및 `custom/max-vlans` 등록 정보 값을 동적으로 늘릴 수 있습니다. 하지만 이러한 등록 정보 값을 줄이려면 도메인이 바인드된 상태여야 합니다.

주 - 해당 등록 정보 값을 줄이면 원치 않는 부작용이 발생할 수 있습니다. 따라서 OS가 보존할 MAC 주소 또는 VLAN ID에 대한 제어 권한이 없으므로 호스트에서 만들어진 불필요한 VNIC 또는 VLAN을 삭제해야 합니다.

또한 `ldm set-vnet` 명령을 통해 사용자정의 가상 네트워크 장치에 대한 최대 VLAN ID 및 MAC 주소 수를 줄이기 전에 가상 네트워크 장치에서 `custom=disable`를 설정하십시오.



주의 - 이 등록 정보를 제한하고 제어하면 이 기능을 효과적으로 사용할 수 있습니다.

- 사용자정의 VLAN ID 또는 사용자정의 대체 MAC 주소 수를 줄이기 전에 만든 VNIC 및 VLAN 장치가 제거되었는지 확인합니다. 제거되지 않은 경우 게스트 도메인에는 구성할 수 없는 VNIC가 존재하므로 수동으로 제거해야 합니다.
- `dladm show-vnic -m` 명령은 지정된 가상 네트워크에 구성된 MAC 주소 및 VLAN ID를 표시합니다. `dladm show-vnic -m` 명령은 게스트 도메인에서 사용 중인 대체 MAC 주소 및 VLAN ID를 표시합니다. 이렇게 하면 모든 대체 MAC 주소 및 VLAN ID가 가상 스위치에 미리 구성된 기존 릴리스에서 벗어나게 됩니다.
- 신뢰할 수 있는 가상 네트워크 기능은 PVLAN 기능과 상호 배타적입니다.
- Logical Domains Manager는 사용자정의 기능을 사용으로 설정하기 전에 이 기능에 대한 게스트 도메인 및 서비스 도메인 지원 검증을 시도합니다. 게스트 도메인이 실행 중인 상태가 아니면 서비스 도메인이 이 기능을 지원하지 않으면 사용자정의가 아닌 대체 MAC 주소 및 VLAN ID를 다시 사용으로 설정하기 전에 `custom=disabled`를 설정해야 합니다.
- 대상 서비스 도메인이 신뢰할 수 있는 가상 네트워크 기능을 지원하는 경우에만 신뢰할 수 있는 가상 네트워크로 도메인의 라이브 마이그레이션을 수행할 수 있습니다.

신뢰할 수 있는 가상 네트워크 구성

이 절에는 신뢰할 수 있는 가상 네트워크를 만드는 방법 및 신뢰할 수 있는 가상 네트워크에 대한 정보를 가져오는 방법에 대한 작업이 포함되어 있습니다.

`ldm add-vnet` 또는 `ldm set-vnet` 명령을 사용하여 `custom` 등록 정보 값을 설정하는 방식으로 신뢰할 수 있는 가상 네트워크를 구성할 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 48 신뢰할 수 있는 가상 네트워크 만들기

다음 명령을 사용하여 ldg1 도메인의 primary-vsw0 가상 스위치에 신뢰할 수 있는 가상 네트워크 ldg1_vnet0을 만들 수 있습니다. custom/max-mac-addr 및 custom/max-vlans 등록 정보 값은 기본값 4096을 사용합니다.

```
primary# ldm add-vnet custom=enable ldg1_vnet0 primary-vsw0 ldg1
primary# ldm list -o network ldg1
...
NETWORK
  NAME          SERVICE          MACADDRESS          PVID|PVLAN|VIDs
  ----          -
  ldg1-vnet0    primary-vsw0@primary 00:14:4f:fa:d7:5e  1|--|--
                DEVICE          :network@1          ID          :1
                LINKPROP       :phys-state         MTU         :1500
                MAXBW           :--                 MODE        :--
                CUSTOM        :enable
                MAX-CUSTOM-MACS:4096          MAX-CUSTOM-VLANS:4096
                PRIORITY      :--                 COS         :--
                PROTECTION   :--
```

예 49 기존 가상 네트워크에서 신뢰할 수 있는 가상 네트워크 기능 사용으로 설정

다음 예는 ldg1 도메인의 ldg1_vnet0 가상 네트워크 장치에 대해 custom=enable을 설정하여 신뢰할 수 있는 가상 네트워크 기능을 사용으로 설정하는 방법을 보여 줍니다. custom/max-mac-addr 및 custom/max-vlans 등록 정보 값은 기본값 4096을 사용합니다.

```
primary# ldm set-vnet custom=enabled ldg1_vnet0 ldg1
primary# ldm list -o network ldg1
...
NETWORK
  NAME          SERVICE          MACADDRESS          PVID|PVLAN|VIDs
  ----          -
  ldg1-vnet0    primary-vsw0@primary 00:14:4f:fa:d7:5e  1|--|--
                DEVICE          :network@1          ID          :1
                LINKPROP       :phys-state         MTU         :1500
                MAXBW           :--                 MODE        :--
                CUSTOM        :enable
                MAX-CUSTOM-MACS:4096          MAX-CUSTOM-VLANS:4096
                PRIORITY      :--                 COS         :--
                PROTECTION   :--
```

예 50 custom/max-mac-addr 및 custom/max-vlans 등록 정보 설정

다음 예는 custom/max-vlans 등록 정보 값을 12로, custom/max-mac-addr 등록 정보 값을 13으로 설정합니다.

이러한 새 등록 정보 값이 이전 값보다 작기 때문에 이 설정을 동적으로 변경할 수 없습니다. 바인드된 도메인 또는 비활성 도메인에만 이런 변경을 수행할 수 있습니다.

```
primary# ldm stop ldg1
primary# ldm set-vnet custom/max-vlans=12 custom/max-mac-addr=13 ldg1_vnet0 ldg1
primary# ldm list -o network ldg1
...
NETWORK
  NAME          SERVICE          MACADDRESS          PVID|PVLAN|VIDs
```

```

-----
ldg1-vnet0  primary-vsw0@primary  00:14:4f:fa:d7:5e  1|--|--
      DEVICE      :network@1      ID      :1
      LINKPROP    :phys-state     MTU     :1500
      MAXBW      :--              MODE    :--
      CUSTOM      :enable
      MAX-CUSTOM-MACS:13          MAX-CUSTOM-VLANS:12
      PRIORITY   :--              COS     :--
      PROTECTION  :--
    
```

예 51 custom/max-mac-addr 및 custom/max-vlans 등록 정보 재설정

다음 예는 널 값을 지정하여 custom/max-mac-addr 등록 정보 값을 기본값 4096으로 재설정하는 방법을 보여 줍니다.

custom=enabled인 경우 custom/max-vlans 등록 정보 값, custom/max-mac-addr 등록 정보 값 또는 둘 다를 재설정할 수 있습니다.

```

primary# ldm set-vnet custom/max-mac-addr= ldg1_vnet0 ldg1
primary# ldm list -o network ldg1
...
NETWORK
  NAME      SERVICE      MACADDRESS      PVID|PVLAN|VIDs
  ----      -
  ldg1-vnet0 primary-vsw0@primary  00:14:4f:fa:d7:5e  1|--|--
      DEVICE      :network@1      ID      :1
      LINKPROP    :phys-state     MTU     :1500
      MAXBW      :--              MODE    :--
      CUSTOM      :enable
      MAX-CUSTOM-MACS:4096          MAX-CUSTOM-VLANS:12
      PRIORITY   :--              COS     :--
      PROTECTION  :--
    
```

예 52 custom/max-mac-addr 및 custom/max-vlans 등록 정보 값 변경

다음 예는 custom/max-vlans 등록 정보 값을 늘리고 custom/max-mac-addr 등록 정보 값을 줄이는 방법을 보여 줍니다. 24가 이전 값인 12보다 크기 때문에 custom/max-vlans 등록 정보 값을 24로 늘릴 수 있습니다. 하지만 custom/max-mac-addr에 대한 최대값을 4096에서 11로 줄이는 것이기 때문에 먼저 도메인을 중지해야 합니다.

```

primary# ldm set-vnet custom/max-vlans=24 ldg1_vnet0 ldg1
primary# ldm stop ldg1
primary# ldm set-vnet custom/max-mac-addr=11 ldg1_vnet0 ldg1
primary# ldm list -o network ldg1
...
NETWORK
  NAME      SERVICE      MACADDRESS      PVID|PVLAN|VIDs
  ----      -
  ldg1-vnet0 primary-vsw0@primary  00:14:4f:fa:d7:5e  1|--|--
      DEVICE      :network@1      ID      :1
      LINKPROP    :phys-state     MTU     :1500
      MAXBW      :--              MODE    :--
      CUSTOM      :enable
      MAX-CUSTOM-MACS:11          MAX-CUSTOM-VLANS:24
      PRIORITY   :--              COS     :--
      PROTECTION  :--
    
```

예 53 신뢰할 수 있는 가상 네트워크 기능 사용 안함으로 설정

다음 예는 `ldg1` 도메인에서 `ldg1_vnet0` 가상 네트워크 장치에 대한 `custom` 등록 정보를 사용 안함으로 설정하는 방법을 보여 줍니다.

```
primary# ldm set-vnet custom=disabled ldg1_vnet0 ldg1
...
NETWORK
NAME          SERVICE          MACADDRESS          PVID|PVLAN|VIDs
-----
ldg1-vnet0    primary-vsw0@primary 00:14:4f:fa:d7:5e  1|--|--
  DEVICE      :network@1      ID      :1
  LINKPROP    :phys-state      MTU     :1500
  MAXBW       :--             MODE    :--
  CUSTOM      :disable
  PRIORITY    :--             COS     :--
  PROTECTION  :--
```

신뢰할 수 있는 가상 네트워크 정보 보기

Logical Domains Manager `list` 하위 명령 몇 가지를 사용하여 신뢰할 수 있는 가상 네트워크 설정에 대한 정보를 가져올 수 있습니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 예는 `ldm list-domain -o network`, `ldm list-bindings` 및 `ldm list-constraints` 명령을 사용하여 신뢰할 수 있는 가상 네트워크 구성에 대한 정보를 표시합니다.

- 다음 예는 `ldm list-domain` 명령을 사용하여 `ldg1` 도메인에 대한 신뢰할 수 있는 가상 네트워크 구성 정보를 보는 방법을 보여 줍니다.

```
primary# ldm list-domain -o network ldg1
...
NETWORK
NAME          SERVICE          MACADDRESS          PVID|PVLAN|VIDs
-----
ldg1-vnet0    primary-vsw0@primary 00:14:4f:fa:d7:5e  1|--|--
  DEVICE      :network@1      ID      :1
  LINKPROP    :phys-state      MTU     :1500
  MAXBW       :--             MODE    :--
  CUSTOM      :enable
  MAX-CUSTOM-MACS:11      MAX-CUSTOM-VLANS:24
  PRIORITY    :--             COS     :--
  PROTECTION  :--
```

- 다음 예는 `ldm list-domain` 명령을 사용하여 `ldg1` 도메인에 대한 구문 분석이 가능한 형식의 신뢰할 수 있는 가상 네트워크 구성 정보를 확인하는 방법을 보여 줍니다.

```
primary# ldm list-domain -o network -p ldg1
VERSION 1.19
DOMAIN|name=ldg1|
MAC|mac-addr=00:14:4f:f9:4b:d0
```

```
VNET|name=ldg1-vnet0|dev=network@1|service=primary-vsw0@primary|mac-addr=00:14:4f:fa:d7:5e|mode=|pvid=1|vid=|mtu=1500|linkprop=phys-state|id=1|alt-mac-addr=|maxbw=|pvlan=|protection=|priority=|cos=|custom=enable|max-mac-addr=11|max-vlans=24
```

- 다음 예는 `ldm list-bindings` 명령을 사용하여 `ldg1` 도메인에 대한 신뢰할 수 있는 가상 네트워크 구성 정보를 확인하는 방법을 보여 줍니다.

```
primary# ldm list-bindings -e -o network ldg1
...
NETWORK
NAME          SERVICE          MACADDRESS          PVID|PVLAN|VIDs
----          -
ldg1-vnet0    primary-vsw0@primary 00:14:4f:fa:d7:5e  1|--|--
      DEVICE      :network@1      ID   :1
      LINKPROP    :phys-state     MTU  :1500
      MAXBW       :--             MODE :--
      CUSTOM      :enable
      MAX-CUSTOM-MACS:11      MAX-CUSTOM-VLANS:24
      PRIORITY    :--             COS   :--
      PROTECTION  :--

PEER          MACADDRESS          PVID|PVLAN|VIDs
----          -
primary-vsw0@primary 00:14:4f:f9:08:28  1|--|--
      LINKPROP    :--             MTU  :1500
      MAXBW       :--             LDC  :0x5
      MODE        :--
```

- 다음 예는 `ldm list-bindings` 명령을 사용하여 `ldg1` 도메인에 대해 구문 분석 가능한 형식으로 신뢰할 수 있는 가상 네트워크 구성 정보를 보는 방법을 보여 줍니다.

```
primary# ldm list-bindings -p ldg1
...
VNET|name=ldg1-vnet0|dev=network@1|service=primary-vsw0@primary|mac-addr=00:14:4f:fa:d7:5e|mode=|pvid=1|vid=|mtu=1500|linkprop=phys-state|id=1|alt-mac-addr=|maxbw=|pvlan=|protection=|priority=|cos=|custom=enable|max-mac-addr=11|max-vlans=24|peer=primary-vsw0@primary|mac-addr=00:14:4f:f9:08:28|mode=|pvid=1|vid=|mtu=1500|maxbw=
```

- 다음 예는 `ldm list-constraints -x` 명령을 실행하여 XML을 생성하는 방법을 보여 줍니다.

```
primary# ldm list-constraints -x ldg1
...
<Section xsi:type="ovf:VirtualHardwareSection_Type">
  <Item>
    <rasd:OtherResourceType>network</rasd:OtherResourceType>
    <rasd:Address>auto-allocated</rasd:Address>
    <gprop:GenericProperty key="vnet_name">ldg1-vnet0</gprop:GenericProperty>
    <gprop:GenericProperty key="service_name">primary-vsw0</gprop:GenericProperty>
    <gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
```

```

<gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
<gprop:GenericProperty key="custom">enable</gprop:GenericProperty>
<gprop:GenericProperty key="max-mac-addr">11</gprop:GenericProperty>
<gprop:GenericProperty key="max-vlans">24</gprop:GenericProperty>
<gprop:GenericProperty key="device">network@1</gprop:GenericProperty>
<gprop:GenericProperty key="id">1</gprop:GenericProperty>
</Item>
</Section>

```

Oracle Solaris 11 네트워킹 관련 기능 차이점

도메인에서 Oracle Solaris 10 OS가 실행되는 경우 일부 Oracle VM Server for SPARC 네트워킹 기능이 Oracle Solaris 11 OS에서와 다르게 작동합니다. 도메인에서 Oracle Solaris 11 OS가 실행되는 경우 Oracle VM Server for SPARC 가상 네트워크 장치와 가상 스위치에 대한 기능 차이점은 다음과 같습니다.

- **`vsw` 장치를 기본 네트워크 인터페이스로 구성하면 서비스 도메인이 게스트 도메인과 통신할 수 있음**

이 구성은 Oracle Solaris 10 OS를 실행하는 도메인에만 필요합니다. Oracle Solaris 11의 경우 가상 스위치가 Oracle Solaris 11 네트워크 스택을 사용하므로, 가상 네트워크 장치가 자동으로 백엔드 장치(예: `net0`)에 해당하는 네트워크 인터페이스와 통신할 수 있습니다. Oracle Solaris 11에서 `vsw` 장치를 네트워크 인터페이스로 구성하는 기능은 지원되지 않습니다.
- **Oracle Solaris 11 `etherstub` 장치를 백엔드 장치로 사용하면 개인 가상 스위치가 생성된 백엔드 장치에 연결되지 않은 경우 가상 스위치는 게스트 도메인 간의 통신만 제공하고, 게스트 도메인과 서비스 도메인 간의 통신을 제공하지 않습니다. `etherstub`를 백엔드 장치로 사용하면 게스트 도메인이 Oracle Solaris 11 서비스 도메인에 구성된 영역(전역 영역 포함)과 통신할 수 있습니다. 이 구성은 해당 `etherstub`에 연결된 VNIC를 사용하여 수행됩니다.**
- **가상 스위치 및 가상 네트워크 장치에 일반 이름 사용**

Oracle Solaris 11 OS에서는 `vsw` 및 `vnet` 장치에 일반 이름이 지정됩니다. 다른 `vsw` 또는 `vnet` 장치인 백엔드 장치로 가상 스위치를 만들지 마십시오. 일반 네트워크 장치 이름과 연관된 실제 물리적 장치를 확인하려면 `dladm show-phys` 명령을 사용하십시오.
- **Oracle Solaris 11 VNIC를 사용하여 이더넷 스텝에 VLAN 만들기**

Oracle Solaris 11 서비스 도메인에서 가상 스위치 인터페이스에 VLAN을 구성하지 마십시오. 이 구성은 지원되지 않습니다. 대신, 가상 스위치의 `net-dev` 등록 정보 값에 해당하는 인터페이스에 VLAN을 만듭니다.

Oracle Solaris 10의 경우 `net-dev` 등록 정보를 널 값으로 설정하여 경로 지정된 가상 스위치를 만들 수 있습니다. 그러나 이 방법은 Oracle Solaris 11에 지원되지 않습니다. 대신, 이더넷 스텝 장치를 통해 VNIC가 VLAN의 일부가 되도록 구성합니다.

다음 예는 이더넷 스텝에 VNIC를 만드는 방법을 보여줍니다. `dladm create-etherstub` 명령이 이더넷 스텝 `estub100`을 만듭니다. 이것은 `ldm add-vsw` 명령에서 가상 스위치를 만들 때 사용되는 지원 장치입니다. `ldm add-vsw` 명령이 가상 스위치를 만듭니다. `dladm create-vnic` 명령이 해당 가상 스위치의 VLAN을 만들기 위해 `etherstub` 위에 VNIC를 만듭니다.

```
primary# dladm create-etherstub estub100
primary# ldm add-vsw net-dev=estub100 vid=100 inter-vnet-link=off \
primary-vsw100 primary
primary# dladm create-vnic -l estub100 -m auto -v 100 vnic100
```

다음 `ldm add-vnet` 명령은 VLAN 100을 통해 `ldg1` 및 `ldg2` 도메인 간에 통신할 수 있는 두 개의 VNIC를 만듭니다.

```
primary# ldm add-vnet vid=100 ldg1-vnet100 primary-vsw100 ldg1
primary# ldm add-vnet vid=100 ldg2-vnet100 primary-vsw100 ldg2
```

다음 예에서 `dladm` 명령은 `ldg1` 및 `ldg2` 게스트 도메인에 VLAN을 만듭니다. `ipadm` 명령이 `ldg1` 및 `ldg2` 도메인에 만든 VNIC의 IP 주소를 만듭니다.

```
ldg1# dladm create-vlan -l net1 -v 100 vlan100
ldg1# ipadm create-ip vlan100
ldg1# ipadm create-ipaddr -T static -a 192.168.100.10/24 vlan100/v4
ldg2# dladm create-vlan -l net1 -v 100 vlan100
ldg2# ipadm create-ip vlan100
ldg2# ipadm create-ipaddr -T static -a 192.168.100.20/24 vlan100/v4
```

- 가상 스위치 및 가상 네트워크 장치에 일반 이름 사용

Oracle Solaris 11 OS에서는 `vsw n` 및 `vnet n` 장치에 일반 이름이 지정됩니다. 다른 `vsw` 또는 `vnet` 장치인 백엔드 장치로 가상 스위치를 만들지 마십시오. 일반 네트워크 장치 이름과 연관된 실제 물리적 장치를 확인하려면 `dladm show-phys` 명령을 사용하십시오.

- 가상 스위치 및 가상 네트워크 장치에 VNIC 사용

`vsw n` 장치에서는 VNIC를 사용할 수 없습니다. `vsw n` 에서 VNIC를 만들려고 시도하면 실패합니다.

- Oracle Solaris 11 게스트 도메인에서 네트워크 관찰성 명령 사용

`ldm list-netdev` 및 `ldm list-netstat` 명령을 사용하여 Oracle Solaris 11 게스트 도메인에 대한 정보를 얻을 수 있습니다.

도메인 마이그레이션

이 장에서는 호스트 시스템 간에 도메인을 마이그레이션하는 방법에 대해 설명합니다.
이 장에서는 다음 주제를 다룹니다.

- “도메인 마이그레이션 소개” [279]
- “마이그레이션 작업 개요” [280]
- “소프트웨어 호환성” [281]
- “마이그레이션 작업 보안” [281]
- “도메인 마이그레이션용 FIPS 140-2 모드” [283]
- “도메인 마이그레이션 제한 사항” [285]
- “도메인 마이그레이션” [287]
- “활성 도메인 마이그레이션” [288]
- “바인드된 도메인 또는 비활성 도메인 마이그레이션” [294]
- “Dry Run 수행” [287]
- “진행 중인 마이그레이션 모니터” [295]
- “진행 중인 마이그레이션 취소” [296]
- “실패한 마이그레이션 복구” [297]
- “비대화식 마이그레이션 수행” [288]
- “마이그레이션 예” [297]

주 - 이 장에서 설명되는 마이그레이션 기능을 사용하려면 최신 버전의 Logical Domains Manager, 시스템 펌웨어 및 Oracle Solaris OS를 실행 중이어야 합니다. Oracle VM Server for SPARC 이전 버전을 사용하여 마이그레이션하는 방법은 [Oracle VM Server for SPARC 3.4 릴리스 노트](#) 및 관리 설명서의 관련 버전을 참조하십시오.

도메인 마이그레이션 소개

도메인 마이그레이션을 통해 호스트 시스템 간에 게스트 도메인을 마이그레이션할 수 있습니다. 마이그레이션이 시작되는 시스템을 소스 시스템이라고 합니다. 도메인이 마이그레이션되는 시스템을 대상 시스템이라고 합니다.

마이그레이션 작업이 진행되는 동안 소스 시스템의 마이그레이션할 도메인이 대상 시스템의 마이그레이션 대상 도메인으로 전송됩니다.

라이브 마이그레이션 기능은 활성 도메인을 계속 실행하면서 마이그레이션할 수 있도록 하여 성능을 향상시킵니다. 라이브 마이그레이션 외에 바인드된 도메인 또는 비활성 도메인을 마이그레이션할 수 있는데, 이를 콜드 마이그레이션이라고 합니다.

도메인 마이그레이션을 사용하여 다음과 같은 작업을 수행할 수 있습니다.

- 시스템 간의 로드 균형 조정
- 게스트 도메인을 계속 실행하면서 하드웨어 유지 관리 수행

최상의 마이그레이션 성능을 달성하려면 소스 시스템과 대상 시스템이 모두 최신 버전의 Logical Domains Manager를 실행 중인지 확인합니다.

마이그레이션 작업 개요

소스 시스템의 Logical Domains Manager는 도메인 마이그레이션 요청을 수락하고 대상 시스템에서 실행되는 Logical Domains Manager에 대한 보안 네트워크 연결을 설정합니다. 마이그레이션은 이 연결이 설정된 후 발생합니다. 마이그레이션 작업은 다음 단계에 따라 수행됩니다.

1단계: 소스 시스템이 대상 시스템에서 실행되는 Logical Domains Manager에 대한 연결을 설정하면 소스 시스템 및 마이그레이션할 도메인에 대한 정보가 대상 시스템으로 전송됩니다. 이 정보는 마이그레이션 가능 여부를 확인하기 위해 일련의 검사를 수행하는 데 사용됩니다. 수행할 검사는 마이그레이션할 도메인의 상태를 기반으로 합니다. 예를 들어, 마이그레이션할 도메인이 활성 상태이면 해당 도메인이 바인드 또는 비활성 상태인 경우와는 다른 일련의 검사가 수행됩니다.

2단계: 1단계의 모든 검사가 성공하면 소스 및 대상 시스템이 마이그레이션을 준비합니다. 대상 시스템에서는 마이그레이션할 도메인을 수신할 도메인이 만들어집니다. 마이그레이션할 도메인이 비활성 또는 바인드 상태이면 마이그레이션 작업이 5단계를 진행합니다.

3단계: 마이그레이션할 도메인이 활성 상태이면 도메인의 런타임 상태 정보가 대상 시스템으로 전송됩니다. 마이그레이션할 도메인이 계속 실행되고 동시에 Logical Domains Manager는 OS가 이 도메인에 대해 적용 중인 수정 사항을 추적합니다. 이 정보는 소스 시스템의 하이퍼바이저에서 검색되어 대상 시스템의 하이퍼바이저에 설치됩니다.

4단계: 마이그레이션할 도메인이 일시 중지됩니다. 이 단계에서 수정된 나머지 상태 정보가 모두 대상 시스템으로 다시 복사됩니다. 따라서 도메인 중단이 거의 발생하지 않거나 아주 잠깐 동안 발생합니다. 중단 시간은 작업 부하에 따라 다릅니다.

5단계: 소스 시스템의 Logical Domains Manager에서 대상 시스템의 Logical Domains Manager로 전달이 발생합니다. 마이그레이션 대상 도메인의 실행이 재개될 때 전달이 발생 (마이그레이션할 도메인이 활성 상태인 경우)하며 소스 시스템의 도메인은 삭제됩니다. 이때부터 마이그레이션 대상 도메인은 실행 중인 도메인의 유일한 버전입니다.

소프트웨어 호환성

마이그레이션을 수행하려면 소스 시스템과 대상 시스템이 다음과 같이 호환되는 소프트웨어를 실행 중이어야 합니다.

- 두 시스템에서 실행 중인 Logical Domains Manager 버전은 현재 버전 또는 가장 최근 릴리스된 이전 버전이어야 합니다.
- 라이브 마이그레이션이 지원되도록 소스 시스템과 대상 시스템에 호환되는 펌웨어 버전이 설치되어 있어야 합니다. 두 시스템은 이 릴리스의 Oracle VM Server for SPARC 소프트웨어에서 지원되는 펌웨어의 최소 버전 이상을 실행 중이어야 합니다.

자세한 내용은 [“마이그레이션에 대한 버전 제한 사항” \[285\]](#)을 참조하십시오.

마이그레이션 작업 보안

Oracle VM Server for SPARC에서는 마이그레이션 작업을 위해 다음과 같은 보안 기능을 제공합니다.

- **인증.** 마이그레이션 작업은 두 시스템에서 실행되므로 어떤 경우 사용자는 소스 시스템과 대상 시스템에서 모두 인증되어야 합니다. 특히 슈퍼유저 이외의 다른 사용자는 `LDom`s Management 권한 프로파일을 사용해야 합니다. 그러나 SSL 인증서로 마이그레이션을 수행하는 경우 사용자를 소스 시스템과 대상 시스템에서 모두 인증할 필요는 없으며 다른 사용자를 지정할 수 없습니다.

`ldm migrate-domain` 명령은 대상 시스템에서의 인증을 위해 선택적으로 대체 사용자 이름을 지정할 수 있도록 허용합니다. 이 대체 사용자 이름을 지정하지 않을 경우 마이그레이션 명령을 실행 중인 사용자의 사용자 이름이 사용됩니다. [예 56. “게스트 도메인 마이그레이션 및 이름 바꾸기”](#)를 참조하십시오. 어떤 경우든 `-p` 옵션을 사용하여 비대화식 마이그레이션을 시작하지 않는 한 대상 시스템에 대한 암호를 제공하라는 메시지가 표시됩니다. [“비대화식 마이그레이션 수행” \[288\]](#)을 참조하십시오.

- **암호화.** Oracle VM Server for SPARC에서는 SSL을 통해 마이그레이션 트래픽을 암호화하여 중요한 데이터가 악용되지 않도록 보호하고 추가 하드웨어 및 전용 네트워크에 대한 요구사항이 발생하지 않도록 합니다.

암호화 장치가 있는 플랫폼에서 소스 및 대상 시스템의 `primary` 도메인에 암호화 장치가 지정된 경우 마이그레이션 작업 속도가 빨라집니다. SSL 작업이 암호화 장치로 오프로드될 수 있기 때문입니다.

CPU에 암호화 명령이 있는 플랫폼에서는 자동으로 마이그레이션 작업 속도가 향상됩니다. 소프트웨어가 아닌 암호화 명령으로 SSL 작업을 수행할 수 있기 때문입니다.

- **FIPS 140-2.** Oracle Solaris FIPS 140-2 공인 OpenSSL 라이브러리를 사용하여 도메인 마이그레이션을 수행하도록 시스템을 구성할 수 있습니다. [“도메인 마이그레이션용 FIPS 140-2 모드” \[283\]](#)를 참조하십시오.

마이그레이션용 SSL 인증서 구성

인증서 기반의 인증을 수행하려면 `ldm migrate-domain` 명령에 `-c` 옵션을 사용합니다. 이 옵션은 암호 파일 및 대체 사용자 옵션과 상호 배타적입니다. `-c` 옵션을 지정하지 않으면 마이그레이션 작업에서 암호 인증을 수행합니다.

▼ 마이그레이션용 SSL 인증서를 구성하는 방법

SSL 인증서를 구성하려면 소스 시스템의 컨트롤 도메인에서 이 작업 단계를 수행해야 합니다.

1. 아직 존재하지 않으면 `/var/share/ldomsmanager/trust` 디렉토리를 만듭니다.

```
src-primary# mkdir /var/share/ldomsmanager/trust
```

2. `ldmd` 인증서를 대상 서버에서 로컬의 신뢰할 수 있는 인증서 디렉토리로 복사합니다.

원격 `ldmd` 인증서는 원격 호스트에 있는 `/var/share/ldomsmanager/server.crt`입니다. 로컬 `ldmd` 신뢰할 수 있는 인증서 디렉토리는 `/var/share/ldomsmanager/trust`입니다. 원격 인증서 파일 `target-hostname.pem`의 이름을 바꿉니다(예: `tgt-primary.pem`).

3. 신뢰할 수 있는 인증서 디렉토리의 인증서에서 `/etc/certs/CA`로 연결되는 심볼릭 링크를 만듭니다.

대상 서버 인증서 `tgt-primary.pem`을 가리키는 대상 서버의 호스트 이름으로 `REMOTE` 변수를 설정합니다.

```
src-primary# ln -s /var/share/ldomsmanager/trust/tgt-primary.pem /etc/certs/CA/
```

4. `svc:/system/ca-certificates` 서비스를 다시 시작합니다.

```
src-primary# svcadm restart svc:/system/ca-certificates
```

5. 구성이 올바른지 확인합니다.

```
src-primary# openssl verify /var/share/ldomsmanager/trust/tgt-primary.pem /var/share/ldomsmanager/trust/tgt-primary.pem: ok
```

6. `ca-certificates` 서비스가 온라인 상태인지 확인합니다.

필요한 경우 서비스를 다시 시작하거나 사용으로 설정합니다.

```
src-primary# svcs ca-certificates
/var/share/ldomsmanager/trust/tgt-primary.pem: ok
STATE          STIME      FMRI
online         0:22:38   svc:/system/ca-certificates:default
```

7. `ldmd` 데몬을 다시 시작합니다.

```
src-primary# svcadm restart ldmd
```

8. 대상 서버에서 이러한 단계를 반복합니다.

SSL 인증서 제거

`/var/share/ldomsmanager/trust` 및 `/etc/certs/CA` 디렉토리에서 `.pem` 파일을 제거하면 `svc:/system/ca-certificates` 서비스와 `ldmd` 서비스를 다시 시작해야 합니다. 서비스를 다시 시작할 때까지 `.pem` 파일을 사용한 마이그레이션은 계속 허용됩니다.

```
localhost# svcadm restart svc:/system/ca-certificates
localhost# svcadm restart ldmd
```

도메인 마이그레이션용 FIPS 140-2 모드

Oracle Solaris FIPS 140-2 공인 OpenSSL 라이브러리를 사용하여 도메인 마이그레이션을 수행하도록 Logical Domains Manager를 구성할 수 있습니다. Logical Domains Manager가 FIPS 140-2 모드인 경우, Logical Domains Manager가 FIPS 140-2 모드로 실행 중인 다른 시스템으로만 도메인을 마이그레이션할 수 있습니다. 비FIPS 시스템으로 마이그레이션하려고 시도하면 거부됩니다. Logical Domains Manager가 FIPS 140-2 모드가 아닌 경우 FIPS 140-2 모드에 있는 Logical Domains Manager로 마이그레이션할 수 없습니다.

Logical Domains Manager를 FIPS 140-2 모드로 성공적으로 시작하려면 FIPS 중개자를 사용으로 설정해야 합니다. 단계별 지침은 [Logical Domains Manager를 FIPS 140-2 모드로 실행하는 방법 \[283\]](#)을 참조하십시오.

FIPS 140 가능 OpenSSL 구현을 사용하는 방법에 대한 정보와 예제는 [Managing Encryption and Certificates in Oracle Solaris 11.3](#)의 “How to Switch to the FIPS 140-Capable OpenSSL Implementation” 및 [Oracle SuperCluster M7-8 Owner's Guide: Overview](#)의 “Example of Enabling Two Applications in FIPS 140 Mode on an Oracle Solaris System”를 참조하십시오.

▼ Logical Domains Manager를 FIPS 140-2 모드로 실행하는 방법

시작하기 전에 Logical Domains Manager를 FIPS 140-2 모드로 실행하기 전에 최소 3.2 버전의 Logical Domains Manager를 실행 중이고 `primary` 도메인에 최소 Oracle Solaris 11.2 OS가 실행되어야 합니다.

1. **FIPS 140-2 OpenSSL 중개자를 설치하고 사용으로 설정합니다.**

a. 필요한 경우 **FIPS 140-2 OpenSSL** 중개자를 설치합니다.

이 패키지는 Oracle Solaris 11.2 OS를 설치할 때 기본적으로 설치되어야 합니다.

```
# pkg install openssl-fips-140
```

b. 현재 **OpenSSL** 중개자를 나열합니다.

```
# pkg mediator openssl
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl vendor local default
```

c. 사용 가능한 **OpenSSL** 중개자를 나열합니다.

```
# pkg mediator -a openssl
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl vendor vendor default
openssl system system fips-140
```



주의 - 전환하려는 OpenSSL 구현이 시스템에 있어야 합니다. 시스템에 없는 구현으로 전환할 경우 시스템을 사용할 수 없게 될 수 있습니다.

d. **FIPS 140-2** 중개자를 사용으로 설정합니다.

```
# pkg set-mediator -I fips-140 openssl
```

e. 재부트합니다.

```
# reboot
```

f. **FIPS 140-2** 중개자가 설정되었는지 확인합니다.

```
# pkg mediator openssl
MEDIATOR VER. SRC. VERSION IMPL. SRC. IMPLEMENTATION
openssl system local fips-140
```

2. **ldmd** 데몬이 **FIPS 140-2** 모드를 사용하도록 구성합니다.

a. **ldmd** 데몬을 **FIPS 140-2** 모드에 넣습니다.

```
# svccfg -s ldms/ldmd setprop ldmd/fips1402_enabled = true
```

b. **ldmd** 데몬을 다시 시작합니다.

```
# svcadm refresh ldmd
# svcadm restart ldmd
```

▼ Logical Domains Manager를 FIPS 140-2 모드에서 기본 모드로 되돌리는 방법

1. 기본 OpenSSL 중개자로 되돌려서 FIPS 140-2 OpenSSL 중개자 사용을 중지합니다. 다른 응용 프로그램에 FIPS 140-2 중개자가 필요하지 않은 경우에만 이 단계를 수행합니다.

```
# pkg set-mediator -I default openssl
```

2. 재부트합니다.

```
# reboot
```

3. ldmd 데몬이 기본 모드를 사용하도록 구성합니다.

```
# svccfg -s ldoms/ldmd setprop ldmd/fips1402_enabled = false
```

4. ldmd 데몬을 다시 시작합니다.

```
# svcadm refresh ldmd
# svcadm restart ldmd
```

도메인 마이그레이션 제한 사항

다음 절에서는 도메인 마이그레이션의 제한 사항에 대해 설명합니다. 마이그레이션을 허용하려면 Logical Domains Manager 소프트웨어 및 시스템 펌웨어 버전이 호환되어야 합니다. 또한 성공적인 도메인 마이그레이션을 보장하기 위해서는 특정 CPU 요구사항을 충족해야 합니다.

모든 소스/대상 플랫폼과 시스템 펌웨어 버전의 조합에서 라이브 마이그레이션이 요건을 갖추고 지원되는 것은 아닙니다. 라이브 마이그레이션을 수행할 수 없는 조합의 경우 대신 콜드 마이그레이션을 수행할 수 있습니다.

마이그레이션에 대한 버전 제한 사항

이 절에서는 라이브 마이그레이션을 수행하는 데 적용되는 버전 제한 사항에 대해 설명합니다.

- **Logical Domains Manager 버전.** 한 시스템에서 최신 버전의 Logical Domains Manager를 실행하고 다른 시스템에서 바로 이전 버전의 Logical Domains Manager를 실행하는 경우에는 어떤 방향으로든지 라이브 마이그레이션을 수행할 수 있습니다.
- **Oracle Solaris OS 버전.** 최소 Oracle Solaris 10 9/10 OS를 실행하는 게스트 도메인에 라이브 마이그레이션을 수행할 수 있습니다. Oracle Solaris 10 10/09 OS 또는 이전

Oracle Solaris OS 버전을 실행하는 게스트 도메인에는 라이브 마이그레이션을 수행할 수 없습니다. 여전히 이전 Oracle Solaris OS 버전을 부트하여 해당 도메인에 콜드 마이그레이션을 수행할 수 있습니다.

- **시스템 펌웨어 버전.** 일반적으로 소스 시스템과 대상 시스템 모두에서 적합한 최소 시스템 펌웨어 버전을 지원하는 경우 두 시스템 간에 라이브 마이그레이션을 수행할 수 있습니다. [Oracle VM Server for SPARC 3.4 설치 설명서](#)의 “최소 시스템 펌웨어 버전”을 참조하십시오.

CPU 간 마이그레이션 제한 사항

UltraSPARC T2, UltraSPARC T2 Plus 또는 SPARC T3 서버와 SPARC T5 서버로 시작하는 서버 간에 라이브 마이그레이션 작업을 수행할 수 없습니다.

perf-counters 설정을 위한 마이그레이션 제한 사항

perf-counters 등록 정보 값이 설정된 도메인의 마이그레이션을 수행할 때는 주의해야 합니다.

perf-counters 등록 정보 값이 global로 설정된 도메인의 마이그레이션을 수행하기 전에 대상 시스템에 perf-counters 등록 정보가 global로 설정된 다른 도메인이 없는지 확인하십시오.

마이그레이션 작업 동안 perf-counters 등록 정보는 소스 시스템, 대상 시스템 또는 양쪽에서 성능 액세스 기능을 사용할 수 있는지 여부에 따라 다르게 처리됩니다.

perf-counters 등록 정보 값은 다음과 같이 처리됩니다.

- **소스 시스템만.** perf-counters 등록 정보 값이 대상 시스템으로 전파되지 않습니다.
- **대상 시스템만.** 마이그레이션할 시스템의 perf-counters 등록 정보 값이 perf-counters=에 해당하는 값으로 업데이트됩니다.
- **소스와 대상 시스템.** perf-counters 등록 정보 값이 마이그레이션할 도메인에서 대상 시스템의 마이그레이션된 도메인으로 전파됩니다.

perf-counters 등록 정보에 대한 자세한 내용은 “[Perf-Counter 등록 정보 사용](#)” [332] 및 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

linkprop=phys-state 설정을 위한 마이그레이션 제한 사항

물리적 NIC 지원 장치가 있고 지원 장치(net-dev=)인 물리적 NIC가 없는 대상 도메인에 대해 linkprop=phys-state가 있는 가상 네트워크 장치를 마이그레이션할 수 있습니다. linkprop=phys-state 제약 조건이 엄격한 요구사항이므로 이러한 도메인이 사용 가능한 net-dev 값이 없는 시스템에 마이그레이션되는 경우 제약 조건이 보존되지만 충족되지는 않습니다. linkprop 등록 정보는 계속 phys-state로 유지되고 네트워크 장치 링크 상태는 링크 작동 중으로 표시됩니다.

많은 수의 가상 장치가 있는 도메인에 대한 마이그레이션 제한 사항

때때로 많은 수의 가상 장치가 있는 도메인을 마이그레이션하면 대상 시스템의 컨트롤 도메인의 응답성이 평소보다 저하됩니다. 이 시간 동안 `ldm` 명령이 중단될 것으로 보이며 표준 Oracle Solaris OS 명령은 평소보다 완료 시간이 좀 더 걸립니다.

이 중단은 마이그레이션된 도메인과 연관된 많은 수의 수신 가상 장치를 처리하는 가상 서버에서 발생합니다. 이 처리가 완료되면 컨트롤 도메인은 정상으로 돌아오고 지연된 `ldm` 명령이 완료됩니다.

도메인에서 사용하는 가상 장치 수를 1000 미만으로 제한하여 이러한 중단을 최소화할 수 있습니다.

도메인 마이그레이션

`ldm migrate-domain` 명령을 사용하여 호스트 시스템 간에 도메인 마이그레이션을 시작할 수 있습니다.

주 - 도메인을 마이그레이션할 때 `cid` 및 `mblock` 등록 정보를 사용하여 지정된 명명된 리소스는 삭제됩니다. 대신, 도메인은 대상 시스템에서 익명 리소스를 사용합니다.

활성 도메인을 계속 실행하면서 마이그레이션하는 방법은 [“활성 도메인 마이그레이션” \[288\]](#)을 참조하십시오. 바인드된 도메인 또는 비활성 도메인을 마이그레이션하는 방법은 [“바인드된 도메인 또는 비활성 도메인 마이그레이션” \[294\]](#)을 참조하십시오.

마이그레이션 옵션 및 피연산자에 대한 자세한 내용은 `ldm(1M)` 매뉴얼 페이지를 참조하십시오.

주 - 도메인 마이그레이션이 완료된 후 새 구성을 소스 및 대상 시스템의 SP에 저장합니다. 결과적으로 마이그레이션된 도메인의 상태는 소스 또는 대상 시스템이 전원 켜다 켜기를 수행할 경우 올바릅니다.

Dry Run 수행

`ldm migrate-domain` 명령에 `-n` 옵션을 제공하면 마이그레이션 검사가 수행되지만 도메인이 마이그레이션되지는 않습니다. 요구사항이 충족되지 않으면 오류로 보고됩니다. Dry Run 결과를 통해 실제 마이그레이션을 시도하기 전에 구성 오류를 수정할 수 있습니다.

주 - 논리적 도메인의 동적 특성으로 인해 Dry Run이 성공하면 실제 마이그레이션이 실패할 수 있으며, Dry Run이 실패하면 실제 마이그레이션이 성공할 수 있습니다.

비대화식 마이그레이션 수행

SSL 인증서 방법을 사용하여 비대화식 마이그레이션 작업을 수행합니다. 레거시 `ldm migrate-domain -p filename` 명령을 사용하여 비대화식 마이그레이션 작업을 시작하는 기능은 지원 중단될 예정이지만 여전히 사용할 수 있습니다.

-p 옵션에 대해 인수로 지정하는 파일 이름은 다음 특성을 가져야 합니다.

- 파일의 첫번째 행에 암호가 포함되어야 합니다.
- 암호는 일반 텍스트여야 합니다.
- 암호는 256자를 초과하지 않아야 합니다.

암호 끝의 줄바꿈 문자와 첫번째 행 뒤에 오는 모든 행은 무시됩니다.

대상 시스템의 암호를 저장한 파일은 적절히 보안되어야 합니다. 이 방식으로 암호를 저장하려면 루트 소유자 또는 권한이 있는 사용자만 파일을 읽거나 쓸 수 있도록 파일 권한이 설정되었는지 확인하십시오(400 또는 600).

활성 도메인 마이그레이션

활성 도메인을 마이그레이션하려는 경우 마이그레이션할 도메인, 소스 시스템 및 대상 시스템에는 특정 요구사항 및 제한 사항이 적용됩니다. 자세한 내용은 [“도메인 마이그레이션 제한 사항” \[285\]](#)을 참조하십시오.

팁 - 소스 시스템과 대상 시스템의 `primary` 도메인에 가상 CPU를 더 추가하여 전체 마이그레이션 시간을 단축시킬 수 있습니다. 각 `primary` 도메인에 2개 이상의 전체 코어가 있는 것이 좋지만 반드시 필요한 것은 아닙니다.

마이그레이션 프로세스를 수행하는 동안 도메인 "시간 손실"이 발생합니다. 이러한 시간 손실 문제를 줄이려면 마이그레이션할 도메인을 NTP(Network Time Protocol) 서버와 같은 외부 시간 소스와 동기화하십시오. 도메인을 NTP 클라이언트로 구성하면 마이그레이션이 완료된 후 바로 도메인의 날짜 및 시간이 수정됩니다.

도메인을 Oracle Solaris 10 NTP 클라이언트로 구성하려면 [System Administration Guide: Network Services](#)의 [“Managing Network Time Protocol \(Tasks\)”](#)를 참조하십시오. 도메

인을 Oracle Solaris 11 NTP 클라이언트로 구성하려면 [Introduction to Oracle Solaris 11 Network Services](#) 의 “Managing Network Time Protocol (Tasks)”를 참조하십시오.

주 - 마이그레이션을 종료할 때 일시 중지 단계 중 게스트 도메인에 짧은 지연이 발생할 수 있습니다. 특히 프로토콜에 재시도 방식(예: TCP)이 포함되거나 재시도 방식이 응용 프로그램 레벨(예: UDP를 통한 NFS)에서 존재하는 경우 이 지연으로 인해 네트워크 통신이 현저히 중단되지 않아야 합니다. 하지만 게스트 도메인이 RIP(Routing Information Protocol)와 같이 네트워크에 민감한 응용 프로그램을 실행하는 경우 작업을 시도할 때 도메인에 짧은 지연 또는 중단이 발생할 수 있습니다. 이 지연은 게스트 네트워크 인터페이스가 제거되어 일시 중지 단계 중 다시 만들어지고 있을 경우 잠깐 동안 발생합니다.

CPU에 대한 도메인 마이그레이션 요구사항

마이그레이션을 수행할 때 CPU에 적용되는 요구사항 및 제한 사항은 다음과 같습니다.

- 대상 시스템에는 마이그레이션할 도메인에서 사용 중인 가상 CPU 수가 수용되도록 사용 가능한 가상 CPU가 충분해야 합니다.
- 게스트 도메인에 `cpu-arch` 등록 정보를 설정하면 프로세서 유형이 다른 시스템 간에 도메인을 마이그레이션할 수 있습니다. `cpu-arch` 값을 변경하려면 게스트 도메인이 바운드 또는 비활성 상태여야 합니다.

지원되는 `cpu-arch` 등록 정보 값은 다음과 같습니다.

- `native`는 CPU 유형이 동일한 플랫폼 간에만 게스트 도메인을 마이그레이션할 수 있도록 하는 CPU 특정 하드웨어 기능을 사용합니다. `native`가 기본값입니다.
- `migration-class1`은 SPARC T4 이후의 SPARC 플랫폼에 대한 CPU 간 마이그레이션 모음입니다. 해당 플랫폼은 지원되는 CPU에 대한 하한이 설정되도록 마이그레이션 도중, 그리고 마이그레이션 후 하드웨어 암호화를 지원합니다.

이 값은 UltraSPARC T2, UltraSPARC T2 Plus, SPARC T3 플랫폼이나 Fujitsu M10 플랫폼과 호환되지 않습니다.

- `sparc64-class1`은 SPARC64 플랫폼에 대한 CPU 간 마이그레이션 모음입니다. `sparc64-class1` 값은 SPARC64 명령을 기반으로 하므로 명령 수가 `generic` 값보다 큼니다. 따라서 `generic` 값에 비해 성능 영향이 없습니다.

이 값은 Fujitsu M10 서버와만 호환됩니다.

- `generic`은 모든 플랫폼의 게스트 도메인이 CPU 유형에 관계없이 마이그레이션을 수행할 수 있도록 하는 데 사용되는 최소의 공통 CPU 하드웨어 기능을 사용합니다.

다음 `isainfo -v` 명령은 `cpu-arch=generic` 및 `cpu-arch=migration-class1`의 경우 시스템에서 사용 가능한 명령을 보여줍니다.

```
■ cpu-arch=generic

# isainfo -v
64-bit sparcv9 applications
```

```

asi_blk_init vis2 vis popc
32-bit sparc applications
asi_blk_init vis2 vis popc v8plus div32 mul32
■ cpu-arch=migration-class1

# isainfo -v
64-bit sparcv9 applications
crc32c cbcond pause mont mpmul sha512 sha256 sha1 md5
camellia des aes ima hpc vis3 fmaf asi_blk_init vis2
vis popc
32-bit sparc applications
crc32c cbcond pause mont mpmul sha512 sha256 sha1 md5
camellia des aes ima hpc vis3 fmaf asi_blk_init vis2
vis popc v8plus div32 mul32

```

generic 값을 사용하면 native 값을 사용할 때 비해 게스트 도메인 성능이 저하될 수 있습니다. 성능 저하가 발생하는 것은 게스트 도메인이 특정 CPU의 고유 하드웨어 기능을 사용하지 않고 지원되는 모든 CPU 유형에서 사용 가능한 일반 CPU 기능만 사용하기 때문입니다. 이러한 기능을 사용하지 않음으로써 generic 값은 다른 기능을 지원하는 CPU를 사용하는 시스템 간에 도메인을 마이그레이션할 수 있는 유연성을 가능하게 합니다.

최소 SPARC T4 시스템 간에 도메인을 마이그레이션할 때 `cpu-arch=migration-class1`을 설정하여 게스트 도메인 성능을 향상시킬 수 있습니다. generic 값 사용 시 성능이 향상되지만, 여전히 native 값이 게스트 도메인에 최고 성능을 제공합니다.

`cpu-arch` 등록 정보가 native로 설정된 경우 다음과 같이 `psrinfo -pv` 명령을 사용하여 프로세서 유형을 확인할 수 있습니다.

```

# psrinfo -pv
The physical processor has 2 virtual processors (0 1)
SPARC-T5 (chipid 0, clock 3600 MHz)

```

`cpu-arch` 등록 정보가 native 이외의 다른 값으로 설정된 경우 `psrinfo -pv` 출력이 플랫폼 유형을 표시하지 않습니다. 대신 이 명령은 `sun4v-cpu` CPU 모듈이 로드되었음을 보여줍니다.

```

# psrinfo -pv
The physical processor has 2 cores and 13 virtual processors (0-12)
The core has 8 virtual processors (0-7)
The core has 5 virtual processors (8-12)
sun4v-cpu (chipid 0, clock 3600 MHz)

```

메모리에 대한 마이그레이션 요구사항

대상 시스템 메모리 요구사항은 다음과 같습니다.

- 도메인 마이그레이션이 수용되도록 사용 가능한 메모리가 충분해야 합니다.

- 사용 가능한 메모리가 호환되는 레이아웃으로 제공되어야 합니다.

호환성 요구사항은 각 SPARC 플랫폼마다 다릅니다. 그러나 최소한, 마이그레이션된 도메인의 각 메모리 블록에 대해 지원되는 최대 페이지 크기에 맞게 실제 주소와 물리적 주소 정렬이 보존되어야 합니다.

`pagesize` 명령을 사용하여 대상 시스템에서 지원되는 최대 페이지 크기를 확인합니다.

최소 Oracle Solaris 11.3 OS를 실행하는 게스트 도메인의 경우, 마이그레이션된 도메인의 메모리 블록이 마이그레이션 중에 자동으로 분할될 수 있으므로 마이그레이션된 도메인을 사용 가능한 메모리 블록보다 작게 맞출 수 있습니다. 최대 페이지 크기에 맞는 경계 안에서만 메모리 블록을 분할할 수 있습니다.

운영체제, 펌웨어, 플랫폼에 대한 다른 메모리 레이아웃 요구사항에 따라 주어진 마이그레이션 동안 메모리 블록이 분할되지 못할 수 있습니다. 이 상황이 발생하면 도메인에 사용 가능한 메모리 총량이 충분하더라도 마이그레이션을 실패할 수 있습니다.

물리적 I/O 장치에 대한 마이그레이션 요구사항

물리적 장치에 대한 직접 액세스 권한을 가진 도메인은 마이그레이션할 수 없습니다. 예를 들어, I/O 도메인은 마이그레이션할 수 없습니다. 하지만 물리적 장치와 연관된 가상 장치는 마이그레이션할 수 있습니다.

자세한 내용은 [“PCIe 끝점 장치에 대한 마이그레이션 요구사항” \[292\]](#) 및 [“PCIe SR-IOV 가상 기능에 대한 마이그레이션 요구사항” \[293\]](#)을 참조하십시오.

가상 I/O 장치에 대한 마이그레이션 요구사항

마이그레이션할 도메인에서 사용되는 모든 가상 I/O 서비스를 대상 시스템에서 사용할 수 있어야 합니다. 즉, 다음 조건이 충족되어야 합니다.

- 마이그레이션할 도메인에서 사용되는 각 가상 디스크 백엔드가 대상 시스템에 정의되어 있어야 합니다. 이 공유 저장소는 SAN 디스크나 NFS 또는 iSCSI 프로토콜을 통해 제공되는 저장소일 수 있습니다. 정의한 가상 디스크 백엔드의 볼륨 및 서비스 이름이 소스 시스템의 볼륨 및 서비스 이름과 동일해야 합니다. 백엔드 경로는 소스 시스템과 대상 시스템에서 다를 수 있지만 반드시 동일한 백엔드를 가리켜야 합니다.



주의 - 소스 시스템과 대상 시스템의 가상 디스크 백엔드 경로가 동일한 저장소를 가리키지 않는 경우에도 마이그레이션이 성공합니다. 하지만 대상 시스템의 도메인 동작을 예측할 수 없으며 도메인을 사용하지 못할 수도 있습니다. 이 문제를 해결하려면 도메인을 중지하고 구성 문제를 수정한 다음 도메인을 다시 시작하십시오. 이러한 단계를 수행하지 않을 경우 도메인이 불일치 상태로 유지될 수 있습니다.

- 마이그레이션할 도메인의 각 가상 네트워크 장치와 대상 시스템의 가상 네트워크 스위치가 상응해야 합니다. 각 가상 네트워크 스위치의 이름은 소스 시스템에서 장치가 연결된 가상 네트워크 스위치의 이름과 동일해야 합니다.

예를 들어, 마이그레이션할 도메인의 `vnet0`이 `switch-y`라는 가상 스위치 서비스에 연결된 경우 대상 시스템의 도메인이 `switch-y`라는 가상 스위치 서비스를 제공해야 합니다.

주 - 마이그레이션 대상 도메인이 필요한 네트워크 리소스에 액세스할 수 있도록 대상 시스템의 물리적 네트워크가 올바르게 구성되어 있어야 합니다. 그렇지 않으면 마이그레이션이 완료된 후 도메인에서 일부 네트워크 서비스를 사용하지 못할 수 있습니다.

예를 들어, 도메인이 올바른 네트워크 서브넷에 액세스할 수 있는지 확인해야 할 수 있습니다. 또한 도메인이 대상 시스템에서 필요한 원격 시스템에 연결할 수 있도록 게이트웨이, 라우터 또는 방화벽이 제대로 구성되어 있는지 확인해야 할 수도 있습니다.

마이그레이션할 도메인에서 사용되며 자동으로 할당된 범위에 속한 MAC 주소를 대상 시스템에서 사용할 수 있어야 합니다.

- 가상 콘솔 집중기(vcc) 서비스가 대상 시스템에 존재해야 하며 하나 이상의 사용 가능한 포트를 가져야 합니다. 명시적 콘솔 제약 조건은 마이그레이션 중 무시됩니다. 마이그레이션 대상 도메인 이름을 콘솔 그룹으로 사용하고 컨트롤 도메인에서 제공되는 vcc 장치에서 사용 가능한 포트를 사용하여 마이그레이션 대상 도메인의 콘솔이 만들어집니다. 컨트롤 도메인에서 제공되는 사용 가능한 포트가 없을 경우 서비스 도메인에서 제공되는 vcc 장치에서 사용 가능한 포트를 사용하여 콘솔이 만들어집니다. 기본 그룹 이름이 충돌할 경우 마이그레이션이 실패합니다.
- 마이그레이션할 도메인에서 사용되는 각 가상 SAN이 대상 시스템에 정의되어 있어야 합니다.

PCIe 끝점 장치에 대한 마이그레이션 요구사항

PCIe 끝점 장치로 구성된 I/O 도메인에서는 도메인 마이그레이션을 수행할 수 없습니다.

직접 I/O 기능에 대한 자세한 내용은 ["PCIe 끝점 장치를 지정하여 I/O 도메인 만들기" \[135\]](#)를 참조하십시오.

PCIe SR-IOV 가상 기능에 대한 마이그레이션 요구사항

PCIe SR-IOV 가상 기능으로 구성된 I/O 도메인에서는 도메인 마이그레이션을 수행할 수 없습니다.

SR-IOV 기능에 대한 자세한 내용은 8장 [PCIe SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기](#)를 참조하십시오.

암호화 장치에 대한 마이그레이션 요구사항

암호화 장치가 있는 플랫폼에서 바인드된 암호화 장치가 있는 게스트 도메인을 마이그레이션할 수 있습니다. 단, 게스트 도메인이 암호화 장치 DR(동적 재구성)을 지원하는 운영체제에서 실행되어야 합니다.

마이그레이션 시작 시 Logical Domains Manager는 마이그레이션할 도메인이 암호화 장치 DR을 지원하는지 여부를 확인합니다. 지원되는 경우 Logical Domains Manager는 도메인에서 암호화 장치를 제거하려고 합니다. 마이그레이션이 완료되면 마이그레이션 대상 도메인에 암호화 장치가 다시 추가됩니다.

주 - 대상 시스템에서 암호화 장치에 대한 제약 조건이 충족되지 않는 경우에도 마이그레이션 작업이 차단되지 않습니다. 이 경우 마이그레이션 대상 도메인의 암호화 장치 수가 마이그레이션 작업 전의 수보다 줄어든 수 있습니다.

활성 도메인의 지연된 재구성

소스 또는 대상 시스템에서의 활성 지연된 재구성 작업은 마이그레이션이 시작되지 않도록 합니다. 마이그레이션이 진행되는 동안에는 지연된 재구성 작업을 시작할 수 없습니다.

활성 도메인에 전원 관리 탄력적 정책이 적용되는 동안 마이그레이션

소스 시스템 또는 대상 시스템에 PM(전원 관리) 탄력적 정책이 적용 중인 경우 라이브 마이그레이션을 수행할 수 있습니다.

다른 도메인에 대한 작업

시스템에서 마이그레이션이 진행되는 동안 마이그레이션 중인 도메인의 상태 또는 구성을 수정할 수 있는 작업은 차단됩니다. 도메인 자체에 대한 모든 작업과 시스템의 다른 도메인에 대한 작업(예: 바인드 및 중지)도 차단됩니다.

OpenBoot PROM의 도메인 또는 커널 디버거에서 실행 중인 도메인 마이그레이션

도메인을 마이그레이션하려면 Logical Domains Manager와 마이그레이션할 도메인에서 실행 중인 Oracle Solaris OS 간에 조정을 수행해야 합니다. 마이그레이션할 도메인이 OpenBoot 또는 커널 디버거(kmdb)에서 실행 중인 경우 이 조정이 불가능합니다. 그 결과 마이그레이션 시도를 실패합니다.

마이그레이션할 도메인이 OpenBoot에서 실행 중인 경우 다음 메시지가 나타납니다.

```
primary# ldm migrate ldg1 system2
Migration is not supported while the domain ldg1 is in the 'OpenBoot Running' state
Domain Migration of LDom ldg1 failed
```

마이그레이션할 도메인이 커널 디버거(kmdb)에서 실행 중인 경우 다음 메시지가 나타납니다.

```
primary# ldm migrate ldg1 system2
Migration is not supported while the domain ldg1 is in the 'Solaris debugging' state
Domain Migration of LDom ldg1 failed
```

바인드된 도메인 또는 비활성 도메인 마이그레이션

바인드된 도메인 또는 비활성 도메인은 마이그레이션 시 실행 중이 아닌 상태이므로 해당 도메인에는 몇 가지 도메인 마이그레이션 제한 사항만 적용됩니다. 따라서 런타임 상태가 복사되지 않으므로 플랫폼 유형 간의 마이그레이션(예: SPARC T3에서 SPARC T5 플랫폼 또는 Fujitsu M10 플랫폼으로의 마이그레이션)을 수행할 수 있습니다.

바인드된 도메인을 마이그레이션하려면 대상 시스템이 마이그레이션할 도메인의 CPU, 메모리 및 I/O 제약 조건을 충족할 수 있어야 합니다. 이러한 제약 조건을 충족할 수 없을 경우 마이그레이션이 실패합니다.



주의 - 바인드된 도메인을 마이그레이션할 때 대상 시스템과 런타임 상태 정보가 교환되지 않으므로 가상 디스크 백엔드 options 및 mpgroup 값이 확인되지 않습니다. 활성 도메인을 마이그레이션할 때는 이러한 값이 확인됩니다.

비활성 도메인 마이그레이션에는 해당 요구사항이 적용되지 않습니다. 하지만 나중에 바인딩이 시도되거나 도메인 바인딩이 실패할 경우 대상 시스템이 마이그레이션 대상 도메인의 제약 조건을 충족해야 합니다.

주 - 도메인 마이그레이션이 완료된 후 새 구성을 소스 및 대상 시스템의 SP에 저장합니다. 결과적으로 마이그레이션된 도메인의 상태는 소스 또는 대상 시스템이 전원 켜다 켜기를 수행할 경우 올바릅니다.

가상 I/O 장치에 대한 마이그레이션 요구사항

비활성 도메인의 경우 VIO(가상 I/O) 제약 조건 검사가 수행되지 않습니다. 따라서 VIO 서버가 존재하지 않아도 마이그레이션이 성공합니다. 도메인이 바인드될 때는 비활성 도메인과 마찬가지로 VIO 서버가 존재해야 하며 사용 가능한 상태여야 합니다.

PCIe 끝점 장치에 대한 마이그레이션 요구사항

PCIe 끝점 장치로 구성된 I/O 도메인에서는 도메인 마이그레이션을 수행할 수 없습니다. 이 요구사항은 바인드된 도메인에만 적용되며 비활성 도메인에는 적용되지 않습니다.

DIO(직접 I/O) 기능에 대한 자세한 내용은 [“PCIe 끝점 장치를 지정하여 I/O 도메인 만들기” \[135\]](#)를 참조하십시오.

PCIe SR-IOV 가상 기능에 대한 마이그레이션 요구사항

PCIe SR-IOV 가상 기능으로 구성된 I/O 도메인에서는 도메인 마이그레이션을 수행할 수 없습니다. 이 요구사항은 바인드된 도메인에만 적용되며 비활성 도메인에는 적용되지 않습니다.

SR-IOV 기능에 대한 자세한 내용은 [8장. PCIe SR-IOV 가상 기능을 사용하여 I/O 도메인 만들기](#)를 참조하십시오.

진행 중인 마이그레이션 모니터

마이그레이션이 진행 중인 경우 상태 출력에 마이그레이션 중인 도메인과 마이그레이션 대상 도메인이 다르게 표시됩니다. `ldm list` 명령의 출력은 도메인 마이그레이션 상태를 나타냅니다.

FLAGS 필드의 여섯번째 열에는 다음 값 중 하나가 표시됩니다.

- s – 마이그레이션 소스인 도메인입니다.
- t – 마이그레이션 대상인 마이그레이션된 도메인입니다.
- e – 오류가 발생하여 사용자 개입이 필요합니다.

다음 명령은 `ldg-src` 도메인이 마이그레이션 소스임을 보여줍니다.

```
# ldm list ldg-src
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg-src   suspended -n---s    1    1G    0.0%   2h 7m
```

다음 명령은 `ldg-tgt` 도메인이 마이그레이션 대상임을 보여줍니다.

```
# ldm list ldg-tgt
NAME      STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg-tgt   bound      -----t 5000    1    1G
```

긴 형식의 상태 출력은 마이그레이션에 대한 추가 정보를 보여줍니다. 소스 시스템에서 상태 출력은 작업 완료율과 대상 시스템 및 마이그레이션 대상 도메인의 이름을 보여줍니다. 마찬가지로 대상 시스템에서 상태 출력은 작업 완료율과 소스 시스템 및 마이그레이션 중인 도메인의 이름을 보여줍니다.

다음 명령은 `ldg-src` 도메인에 대한 마이그레이션 작업 진행률을 보여줍니다.

```
# ldm list -o status ldg-src
NAME
ldg-src

STATUS
OPERATION  PROGRESS  TARGET
migration  17%       system2
```

진행 중인 마이그레이션 취소

마이그레이션이 시작된 후 KILL 신호로 `ldm` 명령이 중단되면 마이그레이션 작업이 종료됩니다. 마이그레이션 작업이 종료되면 마이그레이션 대상 도메인이 삭제되고 마이그레이션할 도메인(활성 상태인 경우)이 재개됩니다. `ldm` 명령의 제어 셸이 손실되면 마이그레이션이 계속 백그라운드로 실행됩니다.

라이브 마이그레이션을 취소할 경우 대상 시스템에 생성된 도메인 인스턴스의 메모리 내용을 하이퍼바이저에서 "지워야" 합니다. 이러한 스크러빙 프로세스는 보안 목적을 위해 수행되며, 메모리를 여유 메모리 풀로 반환하기 전에 완료해야 합니다. 이러한 스크러빙이 진행 중일 때는 `ldm` 명령이 응답하지 않습니다. 따라서 Logical Domains Manager가 중단된 것처럼 보입니다. 따라서 다른 `ldm` 명령을 실행하려고 시도하기 전에 스크러빙 요청이 완료될 때까지 기다립니다. 이 프로세스는 오래 걸릴 수 있습니다. 예를 들어, 500GB 메모리가 포함된 게스트 도메인에서 이 프로세스를 완료하는 데에는 SPARC T4 서버의 경우 최대 7분까지 소요될 수 있습니다.

실패한 마이그레이션 복구

마이그레이션 중인 도메인이 모든 런타임 상태 정보를 마이그레이션 대상 도메인으로 보낸 후 마이그레이션 대상 도메인이 도메인 재개를 확인하기 전에 네트워크 연결이 끊어질 경우 마이그레이션 작업이 종료됩니다.

다음 단계를 수행하여 마이그레이션이 성공적으로 완료되었는지 여부를 확인해야 합니다.

1. 마이그레이션 대상 도메인이 성공적으로 작업을 재개했는지 여부를 확인합니다. 마이그레이션 대상 도메인은 다음과 같은 두 가지 상태 중 하나입니다.
 - 마이그레이션이 성공적으로 완료된 경우 마이그레이션 대상 도메인은 정상 상태입니다.
 - 마이그레이션이 실패한 경우 대상 시스템이 마이그레이션 대상 도메인을 정리하고 삭제합니다.
2. 마이그레이션 대상 도메인이 성공적으로 작업을 재개한 경우 오류 상태의 소스 시스템에서 도메인을 삭제해도 됩니다. 하지만 마이그레이션 대상 도메인이 없을 경우 소스 시스템의 도메인이 여전히 도메인의 마스터 버전이므로 해당 도메인을 복구해야 합니다. 이 도메인을 복구하려면 소스 시스템에서 `ldm cancel-operation` 명령을 실행하십시오. 이 명령은 오류 상태를 처리하고 도메인을 원래 상태로 복원합니다.

마이그레이션 예

예 54 SSL 인증서를 사용하여 게스트 도메인 마이그레이션 수행

이 예에서는 `ldg1` 도메인을 `system2`라는 시스템으로 마이그레이션하는 방법을 보여줍니다. 마이그레이션 작업을 시작하기 전에 소스 시스템과 대상 시스템에서 모두 SSL 인증서를 구성해야 합니다. [마이그레이션용 SSL 인증서를 구성하는 방법 \[282\]](#)을 참조하십시오.

```
# ldm migrate-domain -c ldg1 system2
```

예 55 게스트 도메인 마이그레이션

이 예에서는 `ldg1` 도메인을 `system2`라는 시스템으로 마이그레이션하는 방법을 보여줍니다.

```
# ldm migrate-domain ldg1 system2
Target Password:
```

대상 시스템 암호를 묻지 않고 이 마이그레이션을 수행하려면 다음 명령을 사용하십시오.

```
# ldm migrate-domain -p pfile ldg1 system2
```

`-p` 옵션은 파일 이름을 인수로 사용합니다. 지정된 파일에는 대상 시스템에 대한 수퍼유저 암호가 들어 있습니다. 이 예에서는 `pfile`에 대상 시스템 `system2`에 대한 암호가 들어 있습니다.

예 56 게스트 도메인 마이그레이션 및 이름 바꾸기

이 예에서는 마이그레이션 작업의 일부로 도메인의 이름을 바꾸는 방법을 보여줍니다. 마이그레이션의 일부로 소스 시스템의 `ldg-src` 도메인 이름이 대상 시스템(`system2`)의 `ldg-tgt`로 바뀝니다. 또한 `ldm-admin` 사용자가 대상 시스템에서의 인증에 사용됩니다.

```
# ldm migrate ldg-src ldm-admin@system2:ldg-tgt
Target Password:
```

예 57 마이그레이션 실패 메시지

이 예에서는 대상 시스템이 최신 마이그레이션 기능을 지원하지 않을 경우 표시되는 오류 메시지를 보여줍니다.

```
# ldm migrate ldg1 dt212-346
Target Password:
The target machine is running an older version of the domain
manager that does not support the latest migration functionality.

Upgrading to the latest software will remove restrictions on
a migrated domain that are in effect until it is rebooted.
Consult the product documentation for a full description of
these restrictions.

The target machine is running an older version of the domain manager
that is not compatible with the version running on the source machine.

Domain Migration of LDom ldg1 failed
```

예 58 대상 시스템의 도메인에 대한 마이그레이션 상태 확인

이 예에서는 마이그레이션이 진행되는 동안 마이그레이션 대상 도메인의 상태를 확인하는 방법을 보여줍니다. 이 예에서 소스 시스템은 `t5-sys-1`입니다.

```
# ldm list -o status ldg-tgt
NAME
ldg-tgt

STATUS
  OPERATION   PROGRESS   SOURCE
  migration   55%       t5-sys-1
```

예 59 소스 시스템의 도메인에 대해 구문 분석이 가능한 마이그레이션 상태 확인

이 예에서는 마이그레이션이 진행되는 동안 마이그레이션 중인 도메인에 대해 구문 분석이 가능한 상태를 확인하는 방법을 보여줍니다. 이 예에서 대상 시스템은 `system2`입니다.

```
# ldm list -o status -p ldg-src
VERSION 1.6
DOMAIN|name=ldg-src|
STATUS
|op=migration|progress=42|error=no|target=system2
```

◆◆◆ 15 장

리소스 관리

이 장에는 Oracle VM Server for SPARC 시스템에서 리소스 관리 수행에 대한 정보가 포함되어 있습니다.

이 장에서는 다음 주제를 다룹니다.

- “리소스 재구성” [299]
- “리소스 할당” [301]
- “CPU 할당” [302]
- “하드 분할을 사용하여 시스템 구성” [305]
- “도메인에 물리적 리소스 지정” [311]
- “메모리 동적 재구성 사용” [315]
- “리소스 그룹 사용” [322]
- “전원 관리 사용” [323]
- “동적 리소스 관리 사용” [324]
- “도메인 리소스 나열” [326]
- “Perf-Counter 등록 정보 사용” [332]
- “리소스 관리 문제” [334]

리소스 재구성

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템은 가상 CPU, 가상 I/O 장치, 암호화 장치, 메모리 등의 리소스를 구성할 수 있습니다. 일부 리소스는 실행 중인 도메인에 동적으로 구성할 수 있는 반면, 나머지는 중지된 도메인에 구성해야 합니다. 컨트롤 도메인에서 리소스를 동적으로 구성할 수 없는 경우 먼저 지연된 재구성을 시작해야 합니다. 컨트롤 도메인이 재부트될 때까지 지연된 재구성이 구성 작업을 연기합니다.

동적 재구성

동적 재구성(DR)은 운영체제(OS) 실행 중에 리소스를 추가하거나 제거할 수 있도록 합니다. 특정 리소스 유형의 DR 수행 기능은 논리적 도메인에서 실행 중인 OS의 지원 여부에 따라 결정됩니다.

동적 재구성은 다음 리소스에 대해 지원됩니다.

- 가상 CPU – Oracle Solaris 10 OS 및 Oracle Solaris 11 OS의 모든 버전에서 지원됨
- CPU 전체 코어 – [Oracle VM Server for SPARC 3.4 설치 설명서](#)의 “정규화된 Oracle Solaris OS 버전” 참조
- 가상 I/O 장치 – Oracle Solaris 10 10/08 OS 및 Oracle Solaris 11 OS 이상에서 지원됨
- 암호화 장치 – Oracle Solaris 10 1/13 OS 및 Oracle Solaris 11 OS 이상에서 지원됨
- 메모리 – “메모리 동적 재구성 사용” [315] 참조
- 물리적 I/O devices – 지원되지 않음

DR 기능을 사용하려면 Logical Domains DR 도메인 `drd`가 변경할 도메인에서 실행되고 있어야 합니다. `drd(1M)` 매뉴얼 페이지를 참조하십시오.

지연된 재구성

즉시 발생하는 DR 작업과 달리 지연된 재구성은 다음 상황에서 적용됩니다.

- OS의 다음 재부트 이후
- OS가 실행되지 않는 경우 논리적 도메인 중지 및 시작 이후

일반적으로 지연된 재구성 작업은 컨트롤 도메인으로 제한됩니다. 기타 모든 도메인의 경우 리소스를 동적으로 재구성할 수 없다면 도메인을 중지하여 구성을 수정해야 합니다.

지연된 재구성 작업은 컨트롤 도메인으로 제한됩니다. 루트 도메인에서 지연된 재구성이 진행 중인 동안 제한된 수의 명령을 실행하여 동적으로 완료할 수 없는 작업을 지원할 수 있습니다. 해당 하위 명령은 `add-io`, `set-io`, `remove-io`, `create-vf`, `destroy-vf`입니다. 또한 루트 도메인에서 `ldm start-reconf` 명령을 실행할 수 있습니다. 기타 모든 도메인의 경우 리소스를 동적으로 재구성할 수 없다면 도메인을 중지하여 구성을 수정해야 합니다.

지연된 재구성이 진행 중인 동안 해당 도메인에 대한 다른 재구성 요청은 재부트, 중지 또는 시작될 때까지 연기됩니다.

`ldm cancel-reconf` 명령은 도메인에서 지연된 재구성 작업을 취소합니다. 지연된 재구성 기능 사용 방법에 대한 자세한 내용은 `ldm(1M)` 매뉴얼 페이지를 참조하십시오.

주 - 다른 `ldm remove-*` 명령이 이미 가상 I/O 장치에 대해 지연된 재구성 작업을 실행한 경우 `ldm cancel-reconf` 명령을 사용할 수 없습니다. `ldm cancel-reconf` 명령은 이러한 상황에서 실패합니다.

지연된 재구성을 사용하여 컨트롤 도메인의 리소스를 줄일 수 있습니다. 컨트롤 도메인에서 많은 수의 CPU를 제거하려면 “도메인에서 많은 수의 CPU를 제거할 경우 실패할 수 있음” [334]을 참조하십시오. 컨트롤 도메인에서 많은 양의 메모리를 제거하려면 “컨트롤 도메인의 메모리 줄이기” [317]를 참조하십시오.

주 - primary 도메인이 지연된 재구성 상태인 경우 Oracle VM Server for SPARC가 관리하는 리소스는 primary 도메인이 재부트된 후에만 전원 관리됩니다. Solaris Power Aware Dispatcher에서 관리하는 CPU와 같이 OS가 직접 관리하는 리소스는 이 상태의 영향을 받지 않습니다.

지연된 재구성 중 하나의 CPU 구성 작업만 수행할 수 있음

지연된 재구성 상태일 때는 primary 도메인에서 CPU 구성 작업을 두 개 이상 수행하려고 시도하지 마십시오. 더 많은 CPU 구성 요청을 시도하면 요청이 거부됩니다.

해결 방법: 다음 작업 중 하나를 수행합니다.

- 지연된 재구성을 취소하고, 다른 재구성을 시작한 후, 이전 지연된 재구성에서 손실된 구성 변경사항을 요청합니다.
- 잘못된 CPU 수가 포함된 컨트롤 도메인을 재부트한 후 도메인이 재부트된 다음에 할당을 수정합니다.

리소스 할당

리소스 할당 방식에서는 리소스 할당 제약 조건을 사용하여 바인드 시 도메인에 리소스를 지정합니다.

리소스 할당 제약 조건은 도메인에 리소스를 지정할 때 시스템이 충족해야 하는 엄격한 요구사항입니다. 제약 조건이 충족되지 않을 경우 리소스 할당과 도메인 바인드가 모두 실패합니다.



주의 - 각 도메인이 서로 서비스를 제공하는 두 도메인 사이에 순환 종속성을 만들지 마십시오. 이러한 구성은 한 도메인의 장애로 인해 다른 도메인을 사용할 수 없게 되는 단일 실패 조건 지점을 만듭니다. 또한 순환 종속성 구성의 경우 처음에 바인드된 후 도메인 바인드를 해제하지 못하게 됩니다.

Logical Domains Manager에서는 순환 도메인 종속성을 만들지 못하도록 하지 않습니다.

순환 종속성으로 인해 도메인을 바인드 해제할 수 없는 경우 종속성을 유발하는 장치를 제거한 다음 도메인 바인드 해제를 시도하십시오.

CPU 할당

별도의 도메인에서 동일한 코어의 스레드를 실행하는 경우 성능이 저하되거나 예측 불가능할 수 있습니다. Oracle VM Server for SPARC 소프트웨어는 CPU 유사성 기능을 사용하여 도메인을 시작하기 전에 발생하는 논리적 도메인 바인드 프로세스 중의 CPU 할당을 최적화합니다. 이 기능에서는 동일한 코어의 스레드를 동일한 논리적 도메인에 할당하려고 시도합니다. 이러한 유형의 할당을 사용하면 동일한 코어 내의 스레드 간 캐시 공유가 개선되기 때문입니다.

CPU 유사성 기능에서는 다른 리소스가 있는 한 도메인 사이의 코어 공유가 발생하지 않도록 합니다. 도메인에 부분 코어가 할당되고 추가 스트랜드가 요청되는 경우 해당 부분 코어의 스트랜드가 먼저 바인드되고, 그런 다음 필요에 따라 요청을 충족하기 위해 사용 가능한 다른 코어가 검색됩니다.

CPU 할당 방식에서는 CPU 리소스에 대해 다음 제약 조건을 사용합니다.

- **전체 코어 제약 조건.** 이 제약 조건은 CPU 코어가 가상 CPU가 아닌 도메인에 할당되도록 지정합니다. 도메인에 사용으로 설정된 최대 코어 제약 조건이 없는 한 전체 코어 제약 조건은 각각 `ldm set-core` 또는 `ldm set-vcpu` 명령을 사용하여 추가하거나 제거할 수 있습니다. 도메인은 비활성, 바인드된 또는 활성일 수 있습니다. 하지만 제약 조건 적용 요청을 충족하려면 사용 가능한 충분한 코어가 있어야 합니다. 가장 나쁜 사례로는 다른 도메인과 코어를 공유하는 도메인에서 전체 코어 제약 조건을 요청할 경우 요청을 충족하려면 해제 목록의 코어를 사용할 수 있어야 하는 경우입니다. 가장 좋은 사례로는 코어의 모든 가상 CPU가 이미 코어 경계에 있어서 CPU 리소스 변경 없이 제약 조건이 적용되는 경우입니다.
- **최대 코어 수(max-cores) 제약 조건.** 이 제약 조건은 바인드된 도메인 또는 활성 도메인에 지정할 수 있는 최대 코어 수를 지정합니다.

▼ 전체 코어 제약 조건을 적용하는 방법

최대 코어 제약 조건을 설정하기 전에 도메인에서 전체 코어 제약 조건이 사용으로 설정되었는지 확인하십시오.

1. 도메인에 전체 코어 제약 조건을 적용합니다.

```
primary# ldm set-core 1 domain-name
```

2. 도메인에서 전체 코어 제약 조건이 사용으로 설정되었는지 확인합니다.

```
primary# ldm ls -o resmgmt domain-name
```

`max-cores`는 `unlimited`로 설정되어 있습니다. 도메인은 최대 코어 제약 조건이 사용으로 설정될 때까지 하드 분할과 함께 사용할 수 없습니다.

예 60 전체 코어 제약 조건 적용

이 예에서는 `ldg1` 도메인에서 전체 코어 제약 조건을 적용하는 방법을 보여줍니다. 첫번째 명령은 제약 조건을 적용하고, 두번째 명령은 제약 조건이 사용으로 설정되었는지 여부를 확인합니다.

```
primary# ldm set-core 1 ldg1
primary# ldm ls -o resgmt ldg1
NAME
ldg1

CONSTRAINT
  cpu=whole-core
  max-cores=unlimited
```

▼ 최대 코어 제약 조건을 적용하는 방법

최대 코어 제약 조건을 설정하기 전에 도메인에서 전체 코어 제약 조건이 사용으로 설정되었는지 확인하십시오.

최대 코어 제약 조건은 바인드되거나 활성화인 도메인이 아닌 비활성 도메인에서만 수정하거나 사용/사용 안함으로 설정할 수 있습니다. 컨트롤 도메인에서 최대 코어 제약 조건을 업데이트 하면 `ldm set-domain` 명령이 자동으로 지연된 재구성을 시작합니다.

1. 도메인에서 최대 코어 제약 조건을 사용으로 설정합니다.

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores domain-name
```

주 - 이러한 코어와 연결된 암호화 장치는 코어 추가의 영향을 받지 않습니다. 따라서 시스템에서는 연결된 암호화 장치를 도메인에 자동으로 추가하지 않습니다. 하지만 코어의 마지막 가상 CPU가 제거될 경우에만 암호화 장치가 자동으로 제거됩니다. 이 작업은 암호화 장치가 “고립화”되지 않도록 합니다.

2. 전체 코어 제약 조건이 사용으로 설정되었는지 확인합니다.

```
primary# ldm ls -o resgmt domain-name
```

3. 도메인을 바인드하고 다시 시작합니다.

```
primary# ldm bind domain-name
primary# ldm start domain-name
```

이제 하드 분할과 함께 도메인을 사용할 수 있습니다.

예 61 최대 코어 제약 조건 적용

다음 예에서는 `max-cores` 등록 정보를 설정하여 최대 코어를 3코어로 제한하는 방법을 보여주고 해당 제약 조건이 사용으로 설정되었는지 확인합니다.

```
primary# ldm set-domain max-cores=3 ldg1
```

```
primary# ldm ls -o resgmt ldg1
NAME
ldg1

CONSTRAINT
  cpu=whole-core
  max-cores=3
```

이제 하드 분할과 함께 도메인을 사용할 수 있습니다.

다음 예에서는 바인드 해제 및 비활성 상태인 ldg1 도메인에서 최대 코어 제약 조건을 제거하지만 전체 코어 제약 조건을 그대로 남겨둡니다.

```
primary# ldm stop ldg1
primary# ldm unbind ldg1
primary# ldm set-domain max-cores=unlimited ldg1
```

또 다른 경우로, ldg1 도메인에서 최대 코어 제약 조건과 전체 코어 제약 조건을 모두 제거하려면 다음과 같이 코어 대신 가상 CPU를 지정합니다.

```
primary# ldm set-vcpu 8 ldg1
```

두 경우 모두 도메인을 바인드하고 다시 시작하십시오.

```
primary# ldm bind ldg1
primary# ldm start ldg1
```

전체 코어 제약 조건과 다른 도메인 기능 사이의 상호 작용

이 절에서는 전체 코어 제약 조건과 다음 기능 사이의 상호 작용을 설명합니다.

- [“CPU 동적 재구성” \[304\]](#)
- [“동적 리소스 관리” \[304\]](#)

CPU 동적 재구성

전체 코어 제약 조건은 CPU 동적 재구성(DR)과 완벽하게 호환됩니다. 도메인이 전체 코어 제약 조건으로 정의된 경우 ldm add-core, ldm set-core 또는 ldm remove-core 명령을 사용하여 활성 도메인에서 코어 수를 변경할 수 있습니다.

하지만 바인드된 도메인이나 활성 도메인이 지원된 재구성 모드가 아닌 경우 코어 수는 최대 코어 수를 초과할 수 없습니다. 이 최대값은 최대 코어 제약 조건으로 설정되며, 전체 코어 제약 조건이 사용으로 설정되면 자동으로 사용으로 설정됩니다. 최대 코어 제약 조건을 충족하지 못하는 모든 CPU DR 작업은 실패합니다.

동적 리소스 관리

전체 코어 제약 조건은 동적 리소스 관리(DRM)와 완벽하게 호환됩니다.

전체 코어 제약 조건과 DRM 간에 예상되는 상호 작용은 다음과 같습니다.

- DRM 정책이 도메인에 존재하는 동안, 전체 코어가 제약된 도메인에서 전체 코어 제약이 없는 도메인으로, 또는 전체 코어 제약이 없는 도메인에서 전체 코어가 제약된 도메인으로 전환할 수 없습니다. 예를 들어, 다음과 같습니다.
 - 전체 코어가 제약된 도메인의 경우 `ldm set-vcpu` 명령을 사용하여 가상 CPU 수를 지정하고 전체 코어 제약 조건을 제거할 수 없습니다.
 - 전체 코어 제약이 없는 도메인의 경우 `ldm set-core` 명령을 사용하여 전체 코어 수를 지정하고 전체 코어 제약 조건을 추가할 수 없습니다.
- 전체 코어가 제약된 도메인에서 `attack`, `decay`, `vcpu-min`, `vcpu-max` 값을 지정할 경우 전체 코어의 배수여야 합니다.

하드 분할을 사용하여 시스템 구성

이 절에서는 Oracle VM Server for SPARC 소프트웨어에서 하드 분할을 설명하고 Oracle CPU 라이선스 요구사항을 준수하도록 하드 분할을 사용하는 방법에 대해 설명합니다.

소프트웨어 라이선스의 Oracle의 하드 분할 요구사항에 대한 자세한 내용은 [Partitioning: Server/Hardware Partitioning \(http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf\)](http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf)을 참조하십시오.

- **CPU 코어 및 CPU 스레드.** 이 시스템에서 사용되는 프로세서에는 다중 CPU 코어가 있으며, 각 코어에는 다중 CPU 스레드가 포함되어 있습니다.
- **하드 분할 및 CPU 전체 코어.** CPU 전체 코어 구성을 사용하여 하드 분할이 강제 적용됩니다. CPU 전체 코어 구성에는 개별 CPU 스레드 대신 CPU 전체 코어가 할당된 도메인이 있습니다. 기본적으로 도메인은 CPU 스레드를 사용하도록 구성됩니다.

전체 코어 구성으로 도메인을 바인드할 경우 시스템은 지정된 수의 CPU 코어 및 모든 해당 CPU 스레드를 도메인에 만들고 구성합니다. CPU 전체 코어 구성을 사용하면 바인드된 도메인이나 활성 도메인에 동적으로 지정할 수 있는 CPU 코어 수가 제한됩니다.

- **Oracle 하드 분할 라이선스.** Oracle 하드 분할 라이선스 요구사항을 준수하려면 [Hard Partitioning With Oracle VM Server for SPARC \(http://www.oracle.com/technetwork/server-storage/vm/ovm-sparc-hard-partitioning-1403135.pdf\)](http://www.oracle.com/technetwork/server-storage/vm/ovm-sparc-hard-partitioning-1403135.pdf)을 참조하십시오.

또한 CPU 전체 코어를 다음과 같이 사용해야 합니다.

- Oracle 하드 분할 라이선스를 사용하는 응용 프로그램을 실행하는 도메인은 CPU 전체 코어 및 최대 코어로 구성되어야 합니다.
- Oracle 하드 분할 라이선스를 사용하는 응용 프로그램을 실행하지 않는 도메인은 CPU 전체 코어로 구성되지 않아도 됩니다. 예를 들어, 컨트롤 도메인에서 Oracle 응용 프로그램을 실행하지 않을 경우 해당 도메인은 CPU 전체 코어로 구성하지 않아도 됩니다.

도메인의 구성 확인

`ldm list-o` 명령을 사용하여 도메인이 CPU 전체 코어로 구성되었는지 여부를 확인하고 도메인에 지정된 CPU 코어를 나열하는 방법을 확인할 수 있습니다.

- 도메인이 CPU 전체 코어로 구성되었는지 여부를 확인하려면 다음 명령을 사용합니다.

```
primary# ldm list -o resmgmt domain-name
```

전체 코어 제약 조건이 출력에 나타나고 `max-cores` 등록 정보가 도메인에 대해 구성된 최대 CPU 코어 수를 지정하는지 확인합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 명령은 `ldg1` 도메인이 CPU 전체 코어 및 최대 5코어로 구성되었음을 보여줍니다.

```
primary# ldm list -o resmgmt ldg1
```

```
NAME
```

```
ldg1
```

```
CONSTRAINT
```

```
whole-core
```

```
max-cores=5
```

- 도메인이 바인드될 경우 CPU 코어가 도메인에 지정됩니다. 도메인에 지정된 CPU 코어를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list -o core domain-name
```

다음 명령은 `ldg1` 도메인에 지정된 코어를 보여줍니다.

```
primary# ldm list -o core ldg1
```

```
NAME
```

```
ldg1
```

```
CORE
```

```
CID PCPUSET
```

```
1 (8, 9, 10, 11, 12, 13, 14, 15)
```

```
2 (16, 17, 18, 19, 20, 21, 22, 23)
```

CPU 전체 코어로 도메인 구성

이 절의 작업에서는 CPU 전체 코어로 새 도메인을 만드는 방법, CPU 전체 코어로 기존 도메인을 구성하는 방법 및 CPU 전체 코어로 `primary` 도메인을 구성하는 방법에 대해 설명합니다.

도메인에서 CPU 전체 코어를 사용하도록 구성하려면 다음 명령을 사용합니다.

```
ldm set-core number-of-CPU-cores domain
```

이 명령은 도메인에 대한 최대 CPU 코어 수(최대 코어)도 지정합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

최대 코어 및 CPU 코어 할당은 별도의 명령으로 처리됩니다. 이러한 명령을 사용하면 CPU 코어 할당, 최대값 설정 또는 두 작업 모두 독립적으로 가능합니다. 설정된 최대 코어가 없더라도 할당 단위는 코어로 설정할 수 있습니다. 하지만 Oracle VM Server for SPARC 시스템에서 하드 분할을 구성하는 경우 이 모드에서 시스템 실행은 허용되지 않습니다.

- `add-core`, `set-core` 또는 `rm-core` 하위 명령을 사용하여 도메인에 지정된 수의 CPU 코어를 할당합니다.
- `create-domain` 또는 `set-domain` 하위 명령을 사용하여 `max-cores` 등록 정보 값을 지정함으로써 최대 코어를 설정합니다.

Oracle VM Server for SPARC 시스템에서 하드 분할을 구성하려는 경우 상한값을 설정해야 합니다.

▼ CPU 전체 코어로 새 도메인을 만드는 방법

주 - 선택적으로 최대 코어 제약 조건을 설정할 경우 도메인만 중지하고 바인드 해제하면 됩니다.

1. 도메인을 만듭니다.

```
primary# ldm create domain-name
```

2. 도메인에 대한 CPU 전체 코어 수를 설정합니다.

```
primary# ldm set-core number-of-CPU-cores domain
```

3. (선택사항) 도메인에 대한 max-cores 등록 정보를 설정합니다.

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores domain
```

4. 도메인을 구성합니다.

이 구성 중에 `ldm add-core`, `ldm set-core` 또는 `ldm rm-core` 명령을 사용합니다.

5. 바인드 후 도메인을 시작합니다.

```
primary# ldm bind domain-name
primary# ldm start domain-name
```

예 62 2개의 CPU 전체 코어로 새 도메인 만들기

이 예에서는 2개의 CPU 전체 코어로 `ldg1` 도메인을 만듭니다. 첫번째 명령은 `ldg1` 도메인을 만듭니다. 두번째 명령은 2개의 CPU 전체 코어로 `ldg1` 도메인을 구성합니다.

이 시점에 [CPU 전체 코어로 새 도메인을 만드는 방법 \[307\]](#)의 3단계에 설명된 제한 사항에 따라 도메인에서 추가 구성을 수행할 수 있습니다.

세번째 및 네번째 명령은 ldg1 도메인을 사용할 수 있도록 ldg1 도메인을 바인드하고 시작하는 방법을 보여줍니다.

```
primary# ldm create ldg1
primary# ldm set-core 2 ldg1
...
primary# ldm bind ldg1
primary# ldm start ldg1
```

▼ CPU 전체 코어로 기존 도메인을 구성하는 방법

도메인이 존재하고 CPU 스레드를 사용하도록 구성된 경우 CPU 전체 코어를 사용하도록 구성을 변경할 수 있습니다.

1. (선택사항) 도메인을 중지하고 바인드를 해제합니다.

이 단계는 최대 코어 제약 조건도 설정하는 경우에만 필요합니다.

```
primary# ldm stop domain-name
primary# ldm unbind domain-name
```

2. 도메인에 대한 CPU 전체 코어 수를 설정합니다.

```
primary# ldm set-core number-of-CPU-cores domain
```

3. (선택사항) 도메인에 대한 max-cores 등록 정보를 설정합니다.

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores domain
```

4. (선택사항) 도메인을 다시 바인드하고 다시 시작합니다.

이 단계는 최대 코어 제약 조건도 설정하는 경우에만 필요합니다.

```
primary# ldm bind domain-name
primary# ldm start domain-name
```

예 63 4개의 CPU 전체 코어로 기존 도메인 구성

이 예에서는 4개의 CPU 전체 코어로 기존 도메인 ldg1의 구성을 업데이트합니다.

```
primary# ldm set-core 4 ldg1
```

▼ CPU 전체 코어로 기본 도메인을 구성하는 방법

primary 도메인이 CPU 스레드를 사용하도록 구성된 경우 CPU 전체 코어를 사용하도록 구성을 변경할 수 있습니다.

1. (선택사항) primary 도메인을 지연된 재구성 모드로 전환합니다.

max-cores 등록 정보를 수정하려는 경우에만 지연된 재구성을 시작해야 합니다.

```
primary# ldm start-reconf primary
```

2. **primary** 도메인에 대한 CPU 전체 코어 수를 설정합니다.

```
primary# ldm set-core number-of-CPU-cores primary
```

3. (선택사항) **primary** 도메인에 대한 **max-cores** 등록 정보를 설정합니다.

```
primary# ldm set-domain max-cores=max-number-of-CPU-cores primary
```

4. (선택사항) **primary** 도메인을 재부트합니다.

시스템 구성에 따라 적절한 절차를 사용하여 **primary** 도메인을 재부트합니다. [“PCIe 끝점이 구성된 상태로 루트 도메인 재부트” \[140\]](#)를 참조하십시오.

max-cores 등록 정보를 수정하려는 경우에만 도메인을 재부트해야 합니다.

예 64 2개의 CPU 전체 코어로 컨트롤 도메인 구성

이 예에서는 **primary** 도메인에서 CPU 전체 코어를 구성합니다. 첫번째 명령은 **primary** 도메인에서 지연된 재구성 모드를 시작합니다. 두번째 명령은 2개의 CPU 전체 코어로 **primary** 도메인을 구성합니다. 세번째 명령은 **max-cores** 등록 정보를 2로 설정하고 네번째 명령은 **primary** 도메인을 재부트합니다.

```
primary# ldm start-reconf primary
primary# ldm set-core 2 primary
primary# ldm set-domain max-cores=2 primary
primary# shutdown -i 5
```

선택사항인 1단계와 4단계는 **max-cores** 등록 정보를 수정하는 경우에만 필요합니다.

하드 분할된 시스템과 다른 Oracle VM Server for SPARC 기능과의 상호 작용

이 절에서는 하드 분할된 시스템이 다른 Oracle VM Server for SPARC 기능과 상호 작용하는 방법에 대해 설명합니다.

CPU 동적 재구성

CPU 전체 코어로 구성된 도메인에서 CPU 동적 재구성을 사용할 수 있습니다. 단, 개별 CPU 슬레드가 아닌 전체 CPU 코어만 추가 또는 제거할 수 있습니다. 시스템의 하드 분할 상태는 CPU 동적 재구성 기능으로 유지 관리됩니다. 또한 CPU 코어가 도메인에 동적으로 추가될 경우 최대값이 강제 적용됩니다. 따라서 최대 CPU 수를 초과하는 경우 CPU DR 명령을 실패합니다.

주 - 도메인이 중지되고 바인드가 해제되지 않은 경우 `max-cores` 등록 정보를 변경할 수 없습니다. 따라서 전체 코어 제약 조건이 설정되었을 때 지정된 값에서 최대 코어 수를 높이려면 먼저 도메인을 중지 및 바인드 해제해야 합니다.

다음 명령을 사용하여 동적으로 CPU 전체 코어를 바인드된 도메인 또는 활성 도메인에 추가하거나 바인드된 도메인 또는 활성 도메인에서 제거하고, 동적으로 바인드된 도메인 또는 활성 도메인에 대한 CPU 전체 코어 수를 설정할 수 있습니다.

```
ldm add-core number-of-CPU-cores domain
```

```
ldm rm-core number-of-CPU-cores domain
```

```
ldm set-core number-of-CPU-cores domain
```

주 - 도메인이 활성 상태가 아닌 경우 이러한 명령은 도메인에 대한 최대 CPU 코어 수도 조정합니다. 도메인이 바인드되었거나 활성 상태인 경우 이러한 명령은 도메인에 대한 최대 CPU 코어 수에 영향을 주지 않습니다.

예 65 2개의 CPU 전체 코어를 도메인에 동적으로 추가

이 예에서는 동적으로 `ldg1` 도메인에 2개의 CPU 전체 코어를 추가하는 방법을 보여줍니다. `ldg1` 도메인은 CPU 전체 코어로 구성된 활성 도메인입니다. 첫번째 명령은 `ldg1` 도메인이 활성 상태임을 보여줍니다. 두번째 명령은 `ldg1` 도메인이 CPU 전체 코어 및 최대 4개의 CPU 코어로 구성되었음을 보여줍니다. 세번째 및 네번째 명령은 2개의 CPU 전체 코어 추가 전후에 도메인에 지정된 CPU 코어를 보여줍니다. 네번째 명령은 2개의 CPU 전체 코어를 `ldg1` 도메인에 동적으로 추가합니다.

```
primary# ldm list ldg1
NAME STATE FLAGS CONS VCPU MEMORY UTIL UPTIME
ldg1 active -n---- 5000 16 2G 0.4% 5d 17h 49m
primary# ldm list -o resmgt ldg1
NAME
ldg1

CONSTRAINT
  whole-core
  max-cores=4
primary# ldm list -o core ldg1
NAME
ldg1

CORE
CID PCPUSET
1 (8, 9, 10, 11, 12, 13, 14, 15)
2 (16, 17, 18, 19, 20, 21, 22, 23)
primary# ldm add-core 2 ldg1
primary# ldm list -o core ldg1
NAME
ldg1

CORE
CID PCPUSET
1 (8, 9, 10, 11, 12, 13, 14, 15)
2 (16, 17, 18, 19, 20, 21, 22, 23)
```

3 (24, 25, 26, 27, 28, 29, 30, 31)
 4 (32, 33, 34, 35, 36, 37, 38, 39)

CPU 동적 리소스 관리

동적 리소스 관리(DRM)를 사용하여 일부 도메인에서 CPU 리소스를 자동으로 관리할 수 있습니다.

전원 관리

각 하드 분할된 도메인에 대해 별도의 전원 관리(PM) 정책을 설정할 수 있습니다.

도메인 재부트 또는 재바인드

CPU 전체 코어로 구성된 도메인은 도메인이 다시 시작되거나 전체 시스템이 다시 시작될 경우에도 CPU 전체 코어로 계속 구성됩니다. 도메인은 바인드가 유지되는 기간 내내 동일한 물리적 CPU 코어를 사용합니다. 예를 들어, 도메인이 재부트될 경우 재부트 이전 및 이후 모두 동일한 물리적 CPU 코어를 사용합니다. 또는 도메인이 바인드된 동안 전체 시스템의 전원이 꺼질 경우 시스템의 전원이 켜지면 해당 도메인은 동일한 물리적 CPU 코어로 구성됩니다. 도메인을 바인드 해제한 다음 재바인드하거나 전체 시스템이 새로운 구성으로 다시 시작될 경우에는 도메인이 다른 물리적 CPU 코어를 사용할 수 있습니다.

도메인에 물리적 리소스 지정

Logical Domains Manager는 도메인에 지정될 물리적 리소스를 자동으로 선택합니다. 또한 Oracle VM Server for SPARC 3.4 소프트웨어를 사용하면 전문적인 관리자가 도메인에 지정하거나 도메인에서 제거할 물리적 리소스를 명시적으로 선택할 수 있습니다.

명시적으로 지정하는 리소스를 명명된 리소스라고 합니다. 자동으로 지정되는 리소스를 익명 리소스라고 합니다.



주의 - 전문적인 관리자가 아니면 명명된 리소스를 지정하지 마십시오.

컨트롤 도메인 및 게스트 도메인에 물리적 리소스를 명시적으로 지정할 수 있습니다. 컨트롤 도메인은 활성으로 유지되므로 물리적 리소스 지정 전에 컨트롤 도메인은 선택적으로 지연된 재구성 상태일 수 있습니다. 또는 물리적 지정 시 지연된 재구성이 자동으로 트리거될 수 있습니다.

니다. “컨트롤 도메인에서 물리적 리소스 관리” [314]를 참조하십시오. 물리적 리소스 제한 사항에 대한 자세한 내용은 “도메인에서 물리적 리소스 관리 제한 사항” [315]을 참조하십시오.

컨트롤 도메인 및 게스트 도메인에 다음 물리적 리소스를 명시적으로 지정할 수 있습니다.

- **물리적 CPU.** `cid` 등록 정보를 설정하여 물리적 코어 ID를 도메인에 지정합니다.

`cid` 등록 정보는 구성할 시스템 토폴로지에 대해 잘 알고 있는 관리자만 사용해야 합니다. 이 고급 구성 기능은 특정 할당 규칙을 적용하며 시스템의 전체 성능에 영향을 줄 수 있습니다.

다음 중 하나의 명령을 실행하여 이 등록 정보를 설정할 수 있습니다.

```
ldm add-core cid=core-ID[,core-ID[,...]] domain-name
```

```
ldm set-core cid=core-ID[,core-ID[,...]] domain-name
```

```
ldm rm-core [-f] cid=core-ID[,core-ID[,...]] domain-name
```

코어 ID를 `cid` 등록 정보의 값으로 지정할 경우 `core-ID`가 명시적으로 도메인에 지정되거나 도메인에서 제거됩니다.

주 - `ldm add-core` 명령을 사용하여 익명 코어 리소스를 이미 사용 중인 도메인에 명명된 코어 리소스를 추가할 수 없습니다.

- **물리적 메모리.** `mblock` 등록 정보를 설정하여 연속된 물리적 메모리 영역 세트를 도메인에 지정합니다. 각 물리적 메모리 영역은 물리적 메모리 시작 주소 및 크기로 지정됩니다.

`mblock` 등록 정보는 구성할 시스템 토폴로지에 대해 잘 알고 있는 관리자만 사용해야 합니다. 이 고급 구성 기능은 특정 할당 규칙을 적용하며 시스템의 전체 성능에 영향을 줄 수 있습니다.

다음 중 하나의 명령을 실행하여 이 등록 정보를 설정할 수 있습니다.

```
ldm add-mem mblock=PA-start:size[,PA-start:size[,...]] domain-name
```

```
ldm set-mem mblock=PA-start:size[,PA-start:size[,...]] domain-name
```

```
ldm rm-mem mblock=PA-start:size[,PA-start:size[,...]] domain-name
```

메모리 블록을 도메인에 지정하거나 도메인에서 제거하려면 `mblock` 등록 정보를 설정합니다. 유효한 값에는 콜론(:)으로 구분된 물리적 메모리 시작 주소(`PA-start`) 및 메모리 블록 크기(`size`)가 포함됩니다.

주 - `mblock` 또는 `cid` 등록 정보를 설정할 때 실행 중인 도메인 간에는 DR(동적 재구성)을 사용하여 메모리 또는 코어 리소스를 이동할 수 없습니다. 도메인 간에 리소스를 이동하려면 도메인이 바인드된 상태이거나 비활성 상태인지 확인합니다. 컨트롤 도메인에서 물리적 리소스 관리에 대한 자세한 내용은 “컨트롤 도메인에서 물리적 리소스 관리” [314]를 참조하십시오.

주 - 도메인을 마이그레이션할 때 `cid` 및 `mblock` 등록 정보를 사용하여 지정된 명명된 리소스는 삭제됩니다. 대신, 도메인은 대상 시스템에서 익명 리소스를 사용합니다.

`ldm list-constraints` 명령을 사용하여 도메인에 대한 리소스 제약 조건을 볼 수 있습니다. `physical-bindings` 제약 조건은 도메인에 물리적으로 지정된 리소스 유형을 지정합니다. 도메인이 만들어질 때 `physical-bindings` 제약 조건은 물리적 리소스가 해당 도메인에 지정될 때까지 설정되지 않습니다.

다음과 같은 경우 `physical-bindings` 제약 조건은 특정 값으로 설정됩니다.

- `memory - mblock` 등록 정보가 지정된 경우
- `core - cid` 등록 정보가 지정된 경우
- `core, memory - cid` 등록 정보와 `mblock` 등록 정보가 지정된 경우

▼ physical-bindings 제약 조건을 제거하는 방법

게스트 도메인의 `physical-bindings` 제약 조건을 제거하려면 먼저 물리적으로 바인드된 리소스를 모두 제거해야 합니다.

1. 도메인의 바인드를 해제합니다.

```
primary# ldm unbind domain-name
```

2. 명명된 리소스를 제거합니다.

- 명명된 코어를 제거하려면 다음 명령을 사용합니다.

```
primary# ldm set-core cid= domain-name
```

- 명명된 메모리를 제거하려면 다음 명령을 사용합니다.

```
primary# ldm set-mem mblock= domain-name
```

3. CPU 또는 메모리 리소스를 추가합니다.

- CPU 리소스를 추가하려면 다음 명령을 사용합니다.

```
primary# ldm add-vcpu number domain-name
```

- 메모리 리소스를 추가하려면 다음 명령을 사용합니다.

```
primary# ldm add-mem size[unit] domain-name
```

4. 도메인을 다시 바인드합니다.

```
primary# ldm bind domain-name
```

▼ 물리적으로 바인드되지 않은 모든 리소스를 제거하는 방법

`physical-bindings` 제약 조건이 없는 게스트 도메인을 제한하려면 먼저 물리적으로 바인드되지 않은 리소스를 모두 제거해야 합니다.

1. 도메인의 바인드를 해제합니다.

```
primary# ldm unbind domain-name
```

2. 리소스 수를 0으로 설정합니다.

- CPU 리소스를 설정하려면 다음 명령을 사용합니다.

```
primary# ldm set-core 0 domain-name
```

- 메모리 리소스를 설정하려면 다음 명령을 사용합니다.

```
primary# ldm set-mem 0 domain-name
```

3. 물리적으로 바인드된 CPU 또는 메모리 리소스를 추가합니다.

- CPU 리소스를 추가하려면 다음 명령을 사용합니다.

```
primary# ldm add-core cid=core-ID domain-name
```

- 메모리 리소스를 추가하려면 다음 명령을 사용합니다.

```
primary# ldm add-mem mblock=PA-start:size domain-name
```

4. 도메인을 다시 바인드합니다.

```
primary# ldm bind domain-name
```

컨트롤 도메인에서 물리적 리소스 관리

컨트롤 도메인에서 `physical-bindings` 제약 조건을 제거하거나 제약 조건을 적용하려면 이전 절의 해당 단계를 따르십시오. 하지만 도메인의 바인드를 해제하는 대신 컨트롤 도메인을 지연된 재구성 모드로 전환합니다.

익명 리소스와 물리적으로 바인드된 명명된 리소스 사이의 제약 조건 변경은 지연된 재구성을 자동으로 트리거합니다. `ldm start-reconf primary` 명령을 사용하여 명시적으로 지연된 재구성 모드로 전환할 수도 있습니다.

다른 모든 지연된 재구성 변경과 마찬가지로 프로세스를 완료하려면 도메인(이 경우 컨트롤 도메인)의 재부트를 수행해야 합니다.

주 - 컨트롤 도메인이 지연된 재구성 모드인 경우 컨트롤 도메인에서 `ldm add-mem` 및 `ldm rm-mem` 명령을 사용하여 무제한 메모리 지정을 수행할 수 있습니다. 하지만 `ldm set-core` 명령을 사용할 경우 컨트롤 도메인에 하나의 코어 지정만 수행할 수 있습니다.

도메인에서 물리적 리소스 관리 제한 사항

물리적 리소스 지정에 다음 제한 사항이 적용됩니다.

- 동일 도메인에서 물리적 메모리와 비물리적 메모리 바인드 또는 물리적 코어와 비물리적 코어 바인드는 불가능합니다.
- 동일 도메인에서 비물리적 메모리와 물리적 코어 바인드 또는 비물리적 코어와 물리적 메모리 바인드는 가능합니다.
- 도메인에 물리적 리소스를 추가할 경우 해당하는 리소스 유형은 물리적 바인드로 제약됩니다.
- 익명 CPU를 도메인에 추가하거나 도메인에서 제거하려고 시도하면 `physical-bindings=core`가 실패합니다.
- 바인드가 해제된 리소스의 경우 리소스의 할당 및 확인은 `ldm bind` 명령을 실행할 때만 가능합니다.
- 도메인에서 물리적 메모리를 제거할 경우 이전에 추가된 동일한 물리적 메모리 블록을 제거해야 합니다.
- 물리적 메모리 범위는 겹칠 수 없습니다.
- 도메인에 물리적 리소스를 지정할 때는 `ldm add-core cid=` 또는 `ldm set-core cid=` 명령만 사용할 수 있습니다.
- `ldm add-mem mblock=` 또는 `ldm set-mem mblock=` 명령을 사용하여 여러 개의 물리적 메모리 블록을 지정하는 경우 다른 바인드와 충돌이 있는지 확인하기 위해 주소와 크기가 즉시 검사됩니다.
- 일부 코어가 지정된 도메인은 해당 코어의 나머지 CPU가 해제되고 사용 가능한 경우 전체 코어를 사용할 수 있습니다.

메모리 동적 재구성 사용

메모리 동적 재구성(DR)은 용량을 기반으로 하며, 임의의 메모리 양을 활성 논리적 도메인에 추가하거나 활성 논리적 도메인에서 제거할 수 있도록 해줍니다.

메모리 DR 기능 사용을 위한 요구사항 및 제한 사항은 다음과 같습니다.

- 모든 도메인에서 메모리 DR 작업을 수행할 수 있습니다. 하지만 한 번에 하나의 메모리 DR 작업만 도메인에서 진행 가능합니다.

- 메모리 DR 기능은 한 번의 작업에 포함되는 메모리 주소 및 크기에 대해 256MB 정렬을 적용합니다. “메모리 정렬” [318]을 참조하십시오.
- 사용 가능한 메모리 풀에서 정렬되지 않은 메모리는 메모리 DR 기능을 사용하여 도메인에 지정할 수 없습니다. “정렬되지 않은 메모리 추가” [319]를 참조하십시오.

메모리 DR 작업을 사용하여 도메인의 메모리를 재구성할 수 없는 경우 메모리를 재구성하려면 먼저 도메인을 중지해야 합니다. 도메인이 컨트롤 도메인인 경우 먼저 지연된 재구성을 시작해야 합니다.

특정 환경에서는 Logical Domains Manager가 요청된 메모리 할당을 8KB 또는 4MB의 배수 중에서 그 다음으로 가장 큰 수로 반올림합니다. 다음 예제에서 `ldm list-domain -1` 명령의 샘플 출력을 보면 제약 조건 값이 실제로 할당된 크기보다 작은 것을 알 수 있습니다.

```
Memory:
Constraints: 1965 M
raddr      paddr5      size
0x1000000  0x291000000 1968M
```

메모리 추가

도메인이 활성 상태인 경우 `ldm add-memory` 명령을 사용하여 도메인에 메모리를 동적으로 추가할 수 있습니다. 지정된 메모리 크기가 도메인의 현재 메모리 크기보다 큰 경우 `ldm set-memory` 명령도 메모리를 동적으로 추가할 수 있습니다.

메모리 제거

도메인이 활성 상태인 경우 `ldm remove-memory` 명령을 사용하여 도메인에서 메모리를 동적으로 제거할 수 있습니다. 지정된 메모리 크기가 도메인의 현재 메모리 크기보다 작은 경우 `ldm set-memory` 명령도 메모리를 동적으로 제거할 수 있습니다.

메모리 제거는 오래 실행되는 작업이 될 수 있습니다. 지정된 도메인에 대해 `ldm list -1` 명령을 실행하여 `ldm remove-memory` 명령 진행을 추적할 수 있습니다.

`ldm remove-memory` 명령을 중단(Ctrl-C 누름)시키거나 `ldm cancel-operation memdr` 명령을 실행하여 진행 중인 제거 요청을 취소할 수 있습니다. 메모리 제거 요청을 취소할 경우 제거 요청의 미해결 부분(즉, 도메인에서 제거될 메모리 양)만 영향을 받습니다.

일부 메모리 DR 요청

동적으로 메모리를 도메인에 추가하거나 도메인에서 제거하는 요청은 부분적으로만 이행될 수 있습니다. 그 결과는 추가하거나 제거할 적합한 메모리가 있는지 여부에 달려있습니다.

주 - 메모리는 도메인에서 제거된 후 다른 도메인에 추가되기 전에 지워집니다.

컨트롤 도메인의 메모리 재구성

메모리 DR 기능을 사용하여 컨트롤 도메인의 메모리를 재구성할 수 있습니다. 컨트롤 도메인에서 메모리 DR 요청을 수행할 수 없는 경우 먼저 지연된 재구성을 시작해야 합니다.

메모리 DR 작업은 오래 실행될 수 있으므로 활성 도메인에서 많은 용량의 메모리 제거를 위해 메모리 DR을 사용하는 것은 적절하지 않을 수 있습니다. 특히, 시스템의 초기 구성 중에는 지연된 재구성을 사용하여 컨트롤 도메인의 메모리를 줄여야 합니다.

컨트롤 도메인의 메모리 줄이기

초기 공장 출하 시 기본 구성에서 컨트롤 도메인의 메모리를 줄이려면 메모리 DR 대신 지연된 재구성을 사용합니다. 이러한 경우 컨트롤 도메인의 호스트 시스템의 모든 메모리를 소유하고 있습니다. 활성 도메인은 요청된 모든 메모리를 추가하거나 더 일반적으로는 해제하지 않으므로 메모리 DR 기능은 이 용도로 적합하지 않습니다. 한편 해당 도메인에서 실행 중인 OS는 요청을 처리하기 위해 최선을 다합니다. 또한 메모리 제거는 오래 실행되는 작업이 될 수 있습니다. 초기 컨트롤 도메인의 메모리 줄이기와 마찬가지로 대량 메모리 작업이 수행되는 경우 이러한 문제는 더 커집니다.

이러한 이유로 다음 단계를 수행하여 지연된 재구성을 사용하십시오.

1. `ldm start-reconf primary` 명령을 사용하여 컨트롤 도메인을 지연된 재구성 모드로 설정합니다.
2. 필요에 따라 컨트롤 도메인이 소유하고 있는 호스트 시스템의 리소스를 분할합니다.
3. 필요한 경우 `ldm cancel-reconf` 명령을 사용하여 2단계의 작업을 실행 취소하고 다시 시작합니다.
4. 컨트롤 도메인을 재부트하여 재구성 변경사항을 적용합니다.

동적 및 지연된 재구성

지연된 재구성이 컨트롤 도메인에서 보류 중인 경우 다른 도메인에 대한 메모리 재구성 요청은 거부됩니다. 지연된 재구성이 컨트롤 도메인에서 보류 중이 아닌 경우 메모리 DR을 지원하지 않는 도메인에 대한 메모리 재구성 요청은 거부됩니다. 해당 도메인의 경우 지연된 재구성 요청으로 요청이 변환됩니다.

메모리 정렬

메모리 재구성 요청은 요청이 적용되는 도메인의 상태에 따라 정렬 요구사항이 달라집니다.

활성 도메인에 대한 메모리 정렬

- 동적 추가 및 제거. 메모리 블록의 주소와 크기는 동적 추가 및 동적 제거의 경우 256MB로 정렬됩니다. 최소 작업 크기는 256MB입니다.
정렬되지 않은 요청이나 바인드된 크기보다 큰 제거 요청은 거부됩니다.
메모리 할당을 조정하려면 다음 명령을 사용합니다.
 - `ldm add-memory`. 이 명령과 함께 `--auto-adj` 옵션을 지정할 경우 추가되는 메모리 용량은 256MB로 정렬되며, 이 경우 도메인에 실제로 추가되는 메모리의 용량이 늘어날 수 있습니다.
 - `ldm remove-memory`. 이 명령과 함께 `--auto-adj` 옵션을 지정할 경우 제거되는 메모리 용량은 256MB로 정렬되며, 이 경우 도메인에서 실제로 제거되는 메모리의 용량이 줄어 들 수 있습니다.
 - `ldm set-memory`. 이 명령은 추가 또는 제거 작업으로 취급됩니다. `--auto-adj` 옵션을 지정할 경우 추가되거나 제거되는 메모리 용량은 앞에서 설명한 대로 256MB로 정렬됩니다. 이 정렬로 도메인의 실제 메모리 크기가 늘어날 수 있습니다.
- 지연된 재구성. 메모리 블록의 주소와 크기는 4MB로 정렬됩니다. 정렬되지 않은 요청을 할 경우 요청이 반올림되어 4MB로 정렬됩니다.

바인드된 도메인에 대한 메모리 정렬

메모리 블록의 주소와 크기는 바인드된 도메인의 경우 4MB로 정렬됩니다. 정렬되지 않은 요청을 할 경우 요청이 반올림되어 4MB로 정렬됩니다. 따라서 도메인의 결과 메모리 크기는 예상한 것보다 늘어날 수 있습니다.

`ldm add-memory`, `ldm set-memory` 및 `ldm remove-memory` 명령의 경우 `--auto-adj` 옵션은 결과 메모리의 크기를 반올림하여 256MB로 정렬합니다. 따라서 결과 메모리는 예상한 것보다 늘어날 수 있습니다.

비활성 도메인에 대한 메모리 정렬

`ldm add-memory`, `ldm set-memory` 및 `ldm remove-memory` 명령의 경우 `--auto-adj` 옵션은 결과 메모리의 크기를 반올림하여 256MB로 정렬합니다. 비활성 도메인에 대한 정렬 요구사항은 없습니다.

다. “바인드된 도메인에 대한 메모리 정렬” [318]에 설명된 제한 사항은 이러한 도메인이 바인드된 후 적용됩니다.

정렬되지 않은 메모리 추가

메모리 DR 기능은 활성 도메인에 동적으로 추가되거나 제거되는 메모리 주소 및 크기에 대해 256MB 메모리 정렬을 적용합니다. 따라서 활성 도메인에서 정렬되지 않은 메모리는 메모리 DR을 사용하여 제거할 수 없습니다.

또한 사용 가능한 메모리 풀에서 정렬되지 않은 메모리는 메모리 DR을 사용하여 활성 도메인에 추가할 수 없습니다.

모든 정렬된 메모리가 할당된 후 `ldm add-memory` 명령을 사용하여 나머지 정렬되지 않은 메모리를 바인드된 도메인이나 활성 도메인에 추가할 수 있습니다. 또한 이 명령을 사용하면 지연된 재구성 작업을 통해 나머지 정렬되지 않은 메모리를 컨트롤 도메인에 추가할 수 있습니다.

다음 예는 2개의 나머지 128MB 메모리 블록을 `primary` 및 `ldom1` 도메인에 추가하는 방법을 보여줍니다. `ldom1` 도메인은 바인드된 상태입니다.

다음 명령은 컨트롤 도메인에서 지연된 재구성 작업을 시작합니다.

```
primary# ldm start-reconf primary
Initiating a delayed reconfiguration operation on the primary domain.
All configuration changes for other domains are disabled until the
primary domain reboots, at which time the new configuration for the
primary domain also takes effect.
```

다음 명령은 컨트롤 도메인에 128MB 메모리 블록 중 하나를 추가합니다.

```
primary# ldm add-memory 128M primary
-----
Notice: The primary domain is in the process of a delayed reconfiguration.
Any changes made to the primary domain will only take effect after it reboots.
-----
```

```
primary# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -ndcv- SP     8     2688M  0.1%  23d 8h 8m

primary# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP     8     2560M  0.5%  23d 8h 9m
ldom1         bound   ----- 5000   1     524M
```

다음 명령은 `ldom1` 도메인에 다른 128MB 메모리 블록을 추가합니다.

```
primary# ldm add-mem 128M ldom1
primary# ldm list
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP     8     2560M  0.1%  23d 8h 9m
ldom1         bound   ----- 5000   1     652M
```

메모리 DR 예

다음 예에서는 메모리 DR 작업을 수행하는 방법을 보여줍니다. 관련 CLI 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 66 활성 도메인에서 메모리 DR 작업

이 예는 활성 도메인인 `ldom1`에 동적으로 메모리를 추가하고 제거하는 방법을 보여줍니다.

`ldm list` 출력은 Memory 필드의 각 도메인에 대한 메모리를 보여줍니다.

```
primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP      4       27392M    0.4%    1d 22h 53m
ldom1         active    -n----   5000    2       2G        0.4%    1d 1h 23m
ldom2         bound     ------ 5001    2       200M
```

메모리를 256MB의 배수로 지정해야 하므로 다음 `ldm add-mem` 명령은 오류와 함께 종료됩니다. 다음 `ldm add-mem` 명령은 `--auto-adj` 옵션을 사용하므로 200M를 추가할 메모리 용량으로 지정하더라도 반올림되어 256MB가 됩니다.

```
primary# ldm add-mem 200M ldom1
The size of memory must be a multiple of 256MB.
```

```
primary# ldm add-mem --auto-adj 200M ldom1
Adjusting request size to 256M.
The ldom1 domain has been allocated 56M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP      4       27392M    5.0%    8m
ldom1         active    -n----   5000    2       2304M    0.5%    1m
ldom2         bound     ------ 5001    2       200M
```

메모리를 256MB의 배수로 지정해야 하므로 `ldm rm-mem` 명령은 오류와 함께 종료됩니다. 동일한 명령에 `--auto-adj` 옵션을 추가하면 메모리 양이 다음 256MB 경계로 반내림되므로 메모리 제거를 성공합니다.

```
primary# ldm rm-mem --auto-adj 300M ldom1
Adjusting requested size to 256M.
The ldom1 domain has been allocated 44M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE      FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary       active    -n-cv-   SP      4       27392M    0.3%    8m
ldom1         active    -n----   5000    2       2G        0.2%    2m
ldom2         bound     ------ 5001    2       200M
```

예 67 바인드된 도메인에서 메모리 DR 작업

이 예는 바인드된 도메인인 `ldom2`에 동적으로 메모리를 추가하고 제거하는 방법을 보여줍니다.

ldm list 출력은 Memory 필드의 각 도메인에 대한 메모리를 보여줍니다. 첫번째 ldm add-mem 명령은 100MB의 메모리를 ldom2 도메인에 추가합니다. 다음 ldm add-mem 명령은 --auto-adj 옵션을 지정하여 추가 112MB의 메모리가 ldom2에 동적으로 추가됩니다.

ldm rm-mem 명령은 ldom2 도메인에서 100MB를 동적으로 제거합니다. 동일 명령에 --auto-adj 옵션을 사용하여 300MB의 메모리를 제거할 경우 메모리 용량이 반내림되어 다음 256MB 경계가 됩니다.

```
primary# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.4%  1d 22h 53m
ldom1         active  -n---- 5000  2     2G      0.4%  1d 1h 23m
ldom2         bound   ------ 5001  2     200M
```

```
primary# ldm add-mem 100M ldom2
```

```
primary# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.5%  1d 22h 54m
ldom1         active  -n---- 5000  2     2G      0.2%  1d 1h 25m
ldom2         bound   ------ 5001  2     300M
```

```
primary# ldm add-mem --auto-adj 100M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 112M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.4%  1d 22h 55m
ldom1         active  -n---- 5000  2     2G      0.5%  1d 1h 25m
ldom2         bound   ------ 5001  2     512M
```

```
primary# ldm rm-mem 100M ldom2
```

```
primary# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  3.3%  1d 22h 55m
ldom1         active  -n---- 5000  2     2G      0.2%  1d 1h 25m
ldom2         bound   ------ 5001  2     412M
```

```
primary# ldm rm-mem --auto-adj 300M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 144M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE    FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active  -n-cv- SP    4    27392M  0.5%  1d 22h 55m
ldom1         active  -n---- 5000  2     2G      0.2%  1d 1h 26m
ldom2         bound   ------ 5001  2     256M
```

예 68 도메인 메모리 크기 설정

이 예는 ldm set-memory 명령을 사용하여 도메인에 메모리를 추가하고 제거하는 방법을 보여줍니다.

ldm list 출력은 Memory 필드의 각 도메인에 대한 메모리를 보여줍니다.

```
primary# ldm list
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv- SP     4    27392M  0.5%  1d 22h 55m
ldom1         active    -n---- 5000   2     2G      0.2%  1d 1h 26m
ldom2         bound     ------ 5001   2     256M
```

다음 `ldm set-mem` 명령은 `primary` 도메인의 크기를 3400MB로 설정하려고 시도합니다. 지정된 값이 256MB 경계에 있지 않다는 오류가 나타납니다. 동일 명령에 `--auto-adj` 옵션을 추가하면 일부 메모리를 자동으로 제거하고 256MB 경계에 있을 수 있습니다. 또한 이 명령은 도메인에서 해당 메모리를 사용 중이므로 요청된 메모리 중 일부를 제거할 수 없다는 경고를 표시합니다.

```
primary# ldm set-mem 3400M primary
An ldm set-mem 3400M command would remove 23992MB, which is not a multiple
of 256MB. Instead, run ldm rm-mem 23808MB to ensure a 256MB alignment.
```

```
primary# ldm set-mem --auto-adj 3400M primary
Adjusting request size to 3.4G.
The primary domain has been allocated 184M more memory
than requested because of memory alignment constraints.
Only 9472M of memory could be removed from the primary domain
because the rest of the memory is in use.
```

다음 `ldm set-mem` 명령은 바인드된 상태인 `ldom2` 도메인의 메모리 크기를 690MB로 설정합니다. 동일 명령에 `--auto-adj` 옵션을 사용할 경우 추가 78MB의 메모리가 `ldom2`에 동적으로 추가되어 256MB 경계에 있게 됩니다.

```
primary# ldm set-mem 690M ldom2
primary# ldm list
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv- SP     4    17920M  0.5%  1d 22h 56m
ldom1         active    -n---- 5000   2     2G      0.6%  1d 1h 27m
ldom2         bound     ------ 5001   2     690M
```

```
primary# ldm set-mem --auto-adj 690M ldom2
Adjusting request size to 256M.
The ldom2 domain has been allocated 78M more memory
than requested because of memory alignment constraints.
```

```
primary# ldm list
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary       active    -n-cv- SP     4    17920M  2.1%  1d 22h 57m
ldom1         active    -n---- 5000   2     2G      0.2%  1d 1h 27m
ldom2         bound     ------ 5001   2     768M
```

리소스 그룹 사용

리소스 그룹은 시스템의 리소스를 확인하는 대체 방법을 제공합니다. 리소스는 프로세서 코어, 메모리, I/O 버스 사이의 기본적인 물리적 관계에 따라 그룹화됩니다. 서로 다른 플랫폼은, 심지어 SPARC T5-2 및 SPARC T5-8과 같이 동일 서버 제품군 내의 다른 플랫폼 구성조차도 하드웨어에 차이점이 반영된 다른 리소스 그룹을 가질 수 있습니다. `ldm list-rsrc-group` 명령을 사용하여 리소스 그룹 정보를 봅니다.

리소스 그룹의 멤버십은 하드웨어 구성에 따라 정적으로 정의됩니다. `ldm remove-core` 및 `ldm remove-memory` 명령을 사용하여 특정 리소스 그룹에서 리소스를 조작할 수 있습니다.

- `remove-core` 하위 명령은 도메인에서 제거할 CPU 코어 수를 지정합니다. `-g` 옵션을 사용하여 리소스 그룹을 지정할 때 제거 대상으로 선택된 코어는 모두 해당 리소스 그룹에서 옵니다.
- `remove-memory` 하위 명령은 지정된 메모리 양을 논리적 도메인에서 제거합니다. `-g` 옵션을 사용하여 리소스 그룹을 지정할 때 제거 대상으로 선택된 메모리는 모두 해당 리소스 그룹에서 옵니다.

이 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예제는 “[리소스 그룹 정보 나열](#)” [331]을 참조하십시오.

리소스 그룹 요구사항 및 제한 사항

리소스 그룹 기능은 SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버, SPARC S7 시리즈 서버 및 Fujitsu M10 서버에서만 사용할 수 있습니다.

리소스 그룹 기능에는 다음 제한 사항이 있습니다.

- UltraSPARC T2, UltraSPARC T2 Plus, SPARC T3, SPARC T4 플랫폼에서 사용할 수 없습니다.
- `ldm list-rsrc-group` 명령은 지원되지 않는 플랫폼에 대한 어떤 정보도 표시하지 않으며 `ldm remove-core` 및 `ldm remove-memory` 명령의 `-g` 변형은 작동하지 않습니다.
- 지원되는 플랫폼에서 `domain-name` 대신 `_sys_`를 지정하면 모든 시스템 메모리가 다른 리소스 그룹의 사용 가능한 메모리로 이동합니다. 이 명령은 지원되지 않는 플랫폼에서 `no-op`입니다.

전원 관리 사용

전원 관리(PM)를 사용으로 설정하려면 먼저 ILOM 펌웨어의 버전 3.0 이상에서 PM 정책을 설정해야 합니다. 이 절에서는 Oracle VM Server for SPARC 소프트웨어에서 PM을 사용하기 위해 필요한 정보를 요약합니다.

PM 기능 및 ILOM 기능에 대한 자세한 내용은 다음을 참조하십시오.

- [부록 A. 전원 관리 사용](#)
- *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide*의 “Monitoring Power Consumption”
- *Oracle Integrated Lights Out Manager(ILOM) 3.0* 기능 업데이트 및 릴리스 노트

동적 리소스 관리 사용

정책을 사용하여 DR 작업을 자동으로 수행할 방법을 결정할 수 있습니다. 이때 가상 CPU의 동적 리소스 관리를 제어하기 위한 정책만 만들 수 있습니다.



주의 - 다음 제한 사항은 CPU 동적 리소스 관리(DRM)에 영향을 줍니다.

- UltraSPARC T2 및 UltraSPARC T2 Plus 플랫폼에서는 PM 탄력적 정책이 설정되어 있으면 DRM을 사용으로 설정할 수 없습니다.
- UltraSPARC T2 및 UltraSPARC T2 Plus 플랫폼에서는 DRM이 사용으로 설정되어 있는 경우 성능 정책에서 탄력적 정책으로의 모든 변경이 지연됩니다.
- 도메인 마이그레이션 작업을 수행하기 전에 CPU DRM을 사용 안함으로 설정했는지 확인합니다. 그렇지 않으면 오류 메시지가 표시됩니다.
- PM 탄력적 정책이 설정된 경우 펌웨어에서 정규화된 활용률을 지원하는 경우에만(8.2.0) DRM을 사용할 수 있습니다.

리소스 관리 정책은 가상 CPU를 논리적 도메인에 자동으로 추가하고 논리적 도메인에서 제거할 수 있는 조건을 지정합니다. 정책은 `ldm add-policy`, `ldm set-policy` 및 `ldm remove-policy` 명령을 사용하여 관리됩니다.

```
ldm add-policy [enable=yes|no] [priority=value] [attack=value] [decay=value]
[elastic-margin=value] [sample-rate=value] [tod-begin=hh:mm[:ss]]
[tod-end=hh:mm[:ss]] [util-lower=percent] [util-upper=percent] [vcpu-min=value]
[vcpu-max=value] name=policy-name domain-name...
ldm set-policy [enable=[yes|no]] [priority=[value]] [attack=[value]] [decay=[value]]
[elastic-margin=[value]] [sample-rate=[value]] [tod-begin=[hh:mm:ss]]
[tod-end=[hh:mm:ss]] [util-lower=[percent]] [util-upper=[percent]] [vcpu-min=[value]]
[vcpu-max=[value]] name=policy-name domain-name...
ldm remove-policy [name=policy-name... domain-name
```

이러한 명령 및 리소스 관리 정책 만들기에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

정책은 `tod-begin` 및 `tod-end` 등록 정보로 지정된 기간 동안 유효합니다. `tod-begin`으로 지정된 시간은 24시간 기간에서 `tod-end`로 지정된 시간보다 이전이어야 합니다. 기본적으로 `tod-begin` 및 `tod-end` 등록 정보에 대한 값은 각각 00:00:00 및 23:59:59입니다. 기본값이 사용될 경우 정책은 항상 유효합니다.

정책에서는 `priority` 등록 정보의 값을 사용하여 동적 리소스 관리(DRM) 정책에 대한 우선 순위를 지정합니다. 우선 순위 값은 단일 도메인에서 DRM 정책 사이의 관계 및 단일 시스템에서 DRM이 사용으로 설정된 도메인 사이의 관계를 결정하는 데 사용됩니다. 숫자 값이 낮을수록 더 높은(좋은) 우선 순위를 나타냅니다. 유효한 값은 1 - 9999입니다. 기본값은 99입니다.

`priority` 등록 정보의 동작은 다음과 같이 해제 CPU 리소스 풀의 가용성에 따라 달라집니다.

- 해제 CPU 리소스를 풀에서 사용할 수 있음. 이 경우 `priority` 등록 정보는 단일 도메인에 대해 둘 이상의 겹치는 정책이 정의된 경우 적용할 DRM 정책을 결정합니다.

- **해제 CPU 리소스를 풀에서 사용할 수 없음.** 이 경우 `priority` 등록 정보는 동일 시스템의 낮은 우선 순위 도메인에서 높은 우선 순위 도메인으로 리소스를 동적으로 이동할 수 있는지 여부를 지정합니다. 도메인의 우선 순위는 해당 도메인에 대해 적용된 DRM 정책으로 지정되는 우선 순위입니다.

예를 들어, 높은 우선 순위의 도메인은 낮은 우선 순위의 DRM 정책을 가진 다른 도메인에서 CPU 리소스를 가져올 수 있습니다. 이 리소스 획득 기능은 DRM 정책이 사용으로 설정된 도메인에만 해당합니다. `priority` 값이 같은 도메인은 이 기능의 영향을 받지 않습니다. 따라서 모든 정책에 대해 기본 우선 순위가 사용될 경우 도메인은 낮은 우선 순위의 도메인에서 리소스를 가져올 수 없습니다. 이 기능을 활용하려면 `priority` 등록 정보 값이 같지 않도록 조정하십시오.

예를 들어, `ldg1` 및 `ldg2` 도메인에는 모두 적용 중인 DRM 정책이 있습니다. `ldg1` 도메인에 대한 `priority` 등록 정보는 `ldg2` 도메인의 `priority` 등록 정보 값(2)보다 우선하는 1입니다. 다음 상황에서 `ldg1` 도메인은 `ldg2` 도메인에서 CPU 리소스를 동적으로 제거하고 자신에게 지정할 수 있습니다.

- `ldg1` 도메인에 다른 CPU 리소스가 필요합니다.
- 해제 CPU 리소스 풀이 모두 소진되었습니다.

정책에서는 `util-high` 및 `util-low` 등록 정보 값을 사용하여 CPU 사용률에 대한 상위 및 하위 임계값을 지정합니다. 사용률이 `util-high`의 값을 초과할 경우 숫자가 `vcpu-min`과 `vcpu-max` 값 사이가 될 때까지 가상 CPU가 도메인에 추가됩니다. 사용률이 `util-low` 값 아래로 떨어지면 숫자가 `vcpu-min`과 `vcpu-max` 값 사이가 될 때까지 가상 CPU가 도메인에서 제거됩니다. `vcpu-min`에 도달하면 더 이상 가상 CPU를 동적으로 제거할 수 없습니다. `vcpu-max`에 도달하면 더 이상 가상 CPU를 동적으로 추가할 수 없습니다.

예 69 리소스 관리 정책 추가

예를 들어, 여러 주에 걸쳐 시스템의 일반적인 사용률을 관찰한 후 리소스 사용을 최적화하기 위한 정책을 설정할 수 있습니다. 가장 높은 사용 시간은 매일 오전 9:00 - 오후 6:00(태평양 표준시)이고, 낮은 사용 시간은 매일 오후 6:00 - 오전 9:00(태평양 표준시)입니다.

이 시스템 사용률 관찰 결과에 따라 전체적인 시스템 사용률을 기준으로 다음과 같이 상위 및 하위 정책을 만들기로 결정합니다.

- 상위: 매일 오전 9:00 - 오후 6:00(태평양 표준시)
- 하위: 매일 오후 6:00 - 오전 9:00(태평양 표준시)

다음 `ldm add-policy` 명령은 `ldom1` 도메인에서 높은 사용률 기간 동안 사용될 `high-usage` 정책을 만듭니다.

다음 `high-usage` 정책은 다음을 수행합니다.

- `tod-begin` 및 `tod-end` 등록 정보를 설정하여 시작 및 종료 시간이 각각 오전 9:00 및 오후 6:00가 되도록 지정합니다.
- `util-lower` 및 `util-upper` 등록 정보를 설정하여 정책 분석을 수행할 하한 및 상한이 각각 25% 및 75%가 되도록 지정합니다.

- vcpu-min 및 vcpu-max 등록 정보를 설정하여 최소 및 최대 가상 CPU 수가 각각 2 및 16이 되도록 지정합니다.
- attack 등록 정보를 설정하여 한 리소스 제어 사이클 중 추가되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- decay 등록 정보를 설정하여 한 리소스 제어 사이클 중 제거되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- priority 등록 정보를 설정하여 이 정책의 우선 순위가 1이 되도록 지정합니다. 우선 순위 1은 다른 정책을 적용할 수 있더라도 이 정책이 강제 적용됨을 의미합니다.
- name 등록 정보를 설정하여 정책 파일의 이름이 high-usage가 되도록 지정합니다.
- 지정되지 않은 등록 정보(enable 및 sample-rate 등)에 대해 기본값을 사용합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
primary# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-upper=75 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=high-usage ldom1
```

다음 ldm add-policy 명령은 ldom1 도메인에서 낮은 사용률 기간 동안 사용될 med-usage 정책을 만듭니다.

다음 med-usage 정책은 다음을 수행합니다.

- tod-begin 및 tod-end 등록 정보를 설정하여 시작 및 종료 시간이 각각 오후 6:00 및 오전 9:00가 되도록 지정합니다.
- util-lower 및 util-upper 등록 정보를 설정하여 정책 분석을 수행할 하한 및 상한이 각각 10% 및 50%가 되도록 지정합니다.
- vcpu-min 및 vcpu-max 등록 정보를 설정하여 최소 및 최대 가상 CPU 수가 각각 2 및 16이 되도록 지정합니다.
- attack 등록 정보를 설정하여 한 리소스 제어 사이클 중 추가되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- decay 등록 정보를 설정하여 한 리소스 제어 사이클 중 제거되는 최대 가상 CPU 수가 1이 되도록 지정합니다.
- priority 등록 정보를 설정하여 이 정책의 우선 순위가 1이 되도록 지정합니다. 우선 순위 1은 다른 정책을 적용할 수 있더라도 이 정책이 강제 적용됨을 의미합니다.
- name 등록 정보를 설정하여 정책 파일의 이름이 high-usage가 되도록 지정합니다.
- 지정되지 않은 등록 정보(enable 및 sample-rate 등)에 대해 기본값을 사용합니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
primary# ldm add-policy tod-begin=18:00 tod-end=09:00 util-lower=10 util-upper=50 \
vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1 name=med-usage ldom1
```

도메인 리소스 나열

이 절에서는 ldm 하위 명령에 대한 구문 사용법을 보여주고, 플래그 및 활용률 통계와 같은 몇 가지 출력 용어를 정의하며, 출력으로 표시되는 것과 유사한 예를 제공합니다.

시스템에서 읽을 수 있는 출력

`ldm list` 명령 출력을 사용하는 스크립트를 만드는 경우 항상 `-p` 옵션을 사용하여 시스템에서 읽을 수 있는 형식의 출력을 생성하십시오.

모든 `ldm` 하위 명령에 대한 구문 사용법을 보려면 다음 명령을 사용하십시오.

```
primary# ldm --help
```

`ldm` 하위 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

플래그 정의

도메인에 대한 출력(`ldm list`)에 다음 플래그가 표시될 수 있습니다. 명령에 대해 긴 구문 분석 가능 옵션(`-l -p`)을 사용할 경우 플래그의 문자(예: `flags=normal,control,vio-service`)가 표시됩니다. 그렇지 않을 경우 문자 약어(예: `-n-cv-`)가 표시됩니다. `list` 플래그 값은 위치에 따라 달라집니다. 다음 값은 왼쪽에서 오른쪽까지 6개의 각 열에 나타날 수 있습니다.

열 1 – 도메인 시작 또는 중지

- `s` – 도메인 시작 또는 중지

열 2 – 도메인 상태

- `n` – 일반
- `t` – 전환
- `d` – 누락된 리소스로 인해 시작할 수 없는 저하된 도메인

열 3 – 재구성 상태

- `d` – 지연된 재구성
- `r` – 메모리 동적 재구성

열 4 – 컨트롤 도메인

- `c` – 컨트롤 도메인

열 5 – 서비스 도메인

- `v` – 가상 I/O 서비스 도메인

열 6 – 마이그레이션 상태

- `s` – 마이그레이션의 소스 도메인
- `t` – 마이그레이션의 대상 도메인
- `e` – 마이그레이션 중 오류 발생

활용률 통계 정의

가상 CPU당 활용률 통계(UTIL)가 `ldm list` 명령의 `g` (-1) 옵션을 통해 표시됩니다. 통계는 가상 CPU가 게스트 운영체제 대신 실행하는 데 걸린 시간의 백분율입니다. 가상 CPU는 하이퍼바이저에 양보한 경우를 제외하고 게스트 운영체제 대신 실행하는 것으로 간주됩니다. 게스트 운영체제가 가상 CPU를 하이퍼바이저에 양보하지 않을 경우 게스트 운영체제에서 CPU 사용률은 항상 100%로 표시됩니다.

논리적 도메인에 대해 보고되는 사용률 통계는 도메인의 가상 CPU에 대한 가상 CPU 사용률의 평균입니다. 정규화된 활용률 통계(NORM)는 가상 CPU가 게스트 OS 대신 실행되는 시간의 백분율입니다. 이 값은 사이클 건너뛰기와 같은 작업을 고려합니다. 정규화된 가상화는 시스템에서 시스템 펌웨어 8.2.0 이상을 실행하는 경우에만 사용할 수 있습니다.

PM이 사이클 건너뛰기 작업을 수행하지 않는 경우 100% 활용률은 100% 정규화된 활용률과 같습니다. PM이 사이클 건너뛰기를 4/8로 조정하면 100% 활용률이 50% 활용률과 같아집니다. 즉, CPU는 실제로 사용 가능한 사이클 수의 반만 사용합니다. 따라서 완전히 사용된 CPU의 정규화된 활용률은 50%입니다. `ldm list` 또는 `ldm list -1` 명령을 사용하면 가상 CPU 및 게스트 OS에 대한 정규화된 활용률을 볼 수 있습니다.

다양한 목록 보기

- 설치된 현재 소프트웨어 버전을 보려면 다음 명령을 사용합니다.

```
primary# ldm -v
```

- 모든 도메인에 대해 짧은 목록을 생성하려면 다음 명령을 사용합니다.

```
primary# ldm list
```

- 모든 도메인에 대해 긴 목록을 생성하려면 다음 명령을 사용합니다.

```
primary# ldm list -1
```

- 모든 도메인의 확장된 목록을 생성하려면 다음 명령을 사용합니다.

```
primary# ldm list -e
```

- 모든 도메인의 구문 분석 가능하고 시스템에서 읽을 수 있는 목록을 생성하려면 다음 명령을 사용합니다.

```
primary# ldm list -p
```

- 다음 *format* 옵션 중 하나 이상을 입력하여 출력을 리소스의 하위 세트로 생성할 수 있습니다. 둘 이상의 형식을 지정할 경우 공백 없이 콤마로 항목을 구분하십시오.

```
primary# ldm list -o resource[,resource...] domain-name
```

- `console` – 출력에 가상 콘솔(vcons) 및 가상 콘솔 집중기(vcc) 서비스가 포함됩니다.

- `core` - 출력에 전체 코어가 할당된 도메인에 대한 정보가 포함됩니다.
- `cpu` - 출력에 가상 CPU(`vcpu`), 물리적 CPU(`pcpu`) 및 코어 ID에 대한 정보가 포함됩니다.
- `crypto` - 암호화 장치 출력에 모듈식 산술 장치(`mau`) 및 CWQ(Control Word Queue)와 같은 기타 지원되는 암호화 장치가 포함됩니다.
- `disk` - 출력에 가상 디스크(`vdisk`) 및 가상 디스크 서버(`vds`)가 포함됩니다.
- `domain-name` - 출력에 변수(`var`), 호스트 ID(`hostid`), 도메인 상태, 플래그, UUID 및 소프트웨어 상태가 포함됩니다.
- `memory` - 출력에 `memory`가 포함됩니다.
- `network` - 출력에 매체 액세스 제어(`mac`) 주소, 가상 네트워크 스위치(`vsw`) 및 가상 네트워크(`vnet`) 장치가 포함됩니다.
- `physio` - 물리적 입/출력에 주변 장치 구성요소 상호 연결(`pci`) 및 네트워크 인터페이스 장치(`niu`)가 포함됩니다.
- `resgmt` - 출력에 동적 리소스 관리(DRM) 정책 정보가 포함되고, 현재 실행 중인 정책이 나타나며, 전체 코어 구성과 관련된 제약 조건이 나열됩니다.
- `serial` - 출력에는 가상 논리적 도메인 채널(`vldc`) 서비스와 가상 논리적 도메인 채널 클라이언트(`vldcc`)가 포함됩니다.
- `stats` - 출력에 리소스 관리 정책과 관련된 통계가 포함됩니다.
- `status` - 출력에 진행 중인 도메인 마이그레이션에 대한 상태가 포함됩니다.

다음 예에서는 지정할 수 있는 출력의 다양한 하위 세트를 보여줍니다.

- 컨트롤 도메인에 대한 CPU 정보를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list -o cpu primary
```

- 게스트 도메인에 대한 도메인 정보를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list -o domain ldm2
```

- 게스트 도메인에 대한 메모리 및 네트워크 정보를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list -o network,memory ldm1
```

- 게스트 도메인에 대한 DRM 정책 정보를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list -o resgmt,stats ldm1
```

- 도메인에 대한 변수 및 해당 값을 표시하려면 다음 명령을 사용합니다.

```
primary# ldm list-variable variable-name domain-name
```

예를 들어, 다음 명령은 `ldg1` 도메인에서 `boot-device` 변수에 대한 값을 표시합니다.

```
primary# ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

- 도메인에 바인드된 리소스를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list-bindings domain-name
```

- SP에 저장된 논리적 도메인 구성을 나열하려면 다음 명령을 사용합니다.

ldm list-config 명령은 서비스 프로세서에 저장된 논리적 도메인 구성을 나열합니다. -r 옵션과 함께 사용하면 이 명령은 컨트롤 도메인에 자동 저장 파일이 존재하는 구성을 나열합니다.

구성에 대한 자세한 내용은 “도메인 구성 관리” [337]를 참조하십시오. 더 많은 예는 ldm(1M) 매뉴얼 페이지를 참조하십시오.

```
primary# ldm list-config
factory-default
3guests
foo [next poweron]
primary
reconfig-primary
```

구성 이름 오른쪽에 있는 레이블은 다음을 의미합니다.

- [current] – 마지막으로 부트된 구성(현재 실행 중인 구성과 일치하는 경우만 즉, 재구성을 시작할 때까지)입니다. 재구성 이후 주석은 [next poweron]으로 변경됩니다.
- [next poweron] – 다음 번 전원을 켜다 켤 때 사용될 구성입니다.
- [degraded] – 구성이 이전에 부트된 구성의 저하된 버전입니다.
- 모든 서버 리소스(바인드 및 바인드 해제된 서버 리소스)를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list-devices -a
```

- 할당할 수 있는 메모리 양을 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list-devices mem
MEMORY
      PA                SIZE
      0x14e000000       2848M
```

- 논리적 도메인에 사용할 수 없는 메모리 부분을 확인하려면 다음 명령을 사용합니다.

```
primary# ldm list-devices -a mem
MEMORY
      PA                SIZE                BOUND
      0x0                57M                _sys_
      0x3900000          32M                _sys_
      0x5900000          94M                _sys_
      0xb700000          393M               _sys_
      0x2400000          192M               _sys_
      0x3000000          255G               primary
      0x3ff00000000      64M                _sys_
      0x3ff40000000      64M                _sys_
      0x3ff80000000      128M               _sys_
      0x800000000000     2G                 ldg1
      0x800800000000     2G                 ldg2
      0x801000000000     2G                 ldg3
```

```

0x80180000000    2G          ldg4
0x80200000000    103G         primary
0x81bc0000000    145G         primary
    
```

- 사용 가능한 서비스를 나열하려면 다음 명령을 사용합니다.

```
primary# ldm list-services
```

제약 조건 나열

Logical Domains Manager에서 제약 조건은 특정 도메인에 지정하려는 하나 이상의 리소스입니다. 사용 가능한 리소스에 따라, 도메인에 추가하도록 요청한 리소스를 모두 받거나 아무 것도 얻지 못합니다. `list-constraints` 하위 명령은 도메인에 지정되도록 요청한 리소스를 나열합니다.

- 하나의 도메인에 대한 제약 조건을 나열하려면 다음 명령을 사용합니다.

```
# ldm list-constraints domain-name
```

- 특정 도메인에 대해 XML 형식으로 제약 조건을 나열하려면 다음 명령을 사용합니다.

```
# ldm list-constraints -x domain-name
```

- 구문 분석 가능한 형식으로 모든 도메인에 대한 제약 조건을 나열하려면 다음 명령을 사용합니다.

```
# ldm list-constraints -p
```

리소스 그룹 정보 나열

`ldm list-rsrc-group` 명령을 사용하여 리소스 그룹에 대한 정보를 표시할 수 있습니다.

다음 명령은 모든 리소스 그룹에 대한 정보를 표시합니다.

```
primary# ldm list-rsrc-group
NAME                CORE MEMORY IO
/SYS/CMU4           12  256G  4
/SYS/CMU5           12  256G  4
/SYS/CMU6           12  128G  4
/SYS/CMU7           12  128G  4
    
```

다른 `ldm list-*` 명령과 마찬가지로 구문 분석 가능한 출력, 상세한 출력과 특정 리소스 그룹 및 도메인에 대한 정보를 표시하는 옵션을 지정할 수 있습니다. 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 예제는 `-l` 옵션을 사용하여 `/SYS/CMU5` 리소스 그룹에 대한 상세한 정보를 표시합니다.

```
primary# ldm list-rsrc-group -l /SYS/CMU5
NAME                CORE MEMORY IO
/SYS/CMU5           12  256G  4
    
```

CORE		
CID		BOUND
192, 194, 196, 198, 200, 202, 208, 210		primary
212, 214, 216, 218		primary
MEMORY		
PA	SIZE	BOUND
0xc00000000000	228M	ldg1
0xc00300000000	127G	primary
0xc1ffc0000000	64M	_sys_
0xd00000000000	130816M	primary
0xd1ffc0000000	64M	_sys_
IO		
DEVICE	PSEUDONYM	BOUND
pci@900	pci_24	primary
pci@940	pci_25	primary
pci@980	pci_26	primary
pci@9c0	pci_27	primary

Perf-Counter 등록 정보 사용

성능 레지스터 액세스 제어 기능을 통해 특정 성능 레지스터 그룹에 대한 도메인의 액세스 권한을 얻고 설정하고 설정 해제할 수 있습니다.

`ldm add-domain` 및 `ldm set-domain` 명령을 사용하여 `perf-counters` 등록 정보의 값을 지정합니다. 다음 재부트 시 게스트 도메인에서 새로운 `perf-counters` 등록 정보 값을 인식합니다. `perf-counters` 값을 지정하지 않으면 `htstrand` 값이 사용됩니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

`perf-counters` 등록 정보에 대해 다음 값을 지정할 수 있습니다.

<code>global</code>	할당된 리소스에서 액세스할 수 있는 전역 성능 카운터에 도메인 액세스 권한을 부여합니다. 한번에 하나의 도메인만 전역 성능 카운터에 액세스할 수 있습니다. 이 값을 단독으로 지정하거나 <code>strand</code> 또는 <code>htstrand</code> 값과 함께 지정할 수 있습니다.
<code>strand</code>	도메인에 할당된 CPU에 있는 스트랜드 성능 카운터에 도메인 액세스 권한을 부여합니다. 이 값은 <code>htstrand</code> 값과 함께 지정할 수 없습니다.
<code>htstrand</code>	<code>strand</code> 값과 똑같이 작동하며 도메인에 할당된 CPU에서 초특권 모드 이벤트를 계측합니다. 이 값은 <code>strand</code> 값과 함께 지정할 수 없습니다.

성능 카운터에 대한 액세스를 모두 사용 안함으로 설정하려면 `perf-counters=`를 지정합니다.

하이퍼바이저에 성능 액세스 기능이 없을 때 `perf-counters` 등록 정보를 설정하려고 시도하면 실패합니다.

ldm list -o domain 및 ldm list -e 명령은 perf-counters 등록 정보의 값을 표시합니다. 성능 액세스 기능이 지원되지 않으면 perf-counters 값이 출력에 표시되지 않습니다.

예 70 도메인을 만들고 성능 레지스터 액세스 지정

global 레지스터 세트에 액세스할 수 있는 ldg0 도메인을 새로 만듭니다.

```
primary# ldm add-domain perf-counters=global ldg0
```

예 71 도메인에 대한 성능 레지스터 액세스 지정

ldg0 도메인에서 global 및 strand 레지스터 세트에 액세스할 수 있도록 지정합니다.

```
primary# ldm set-domain perf-counters=global,strand ldg0
```

예 72 도메인에서 레지스터 세트에 액세스할 수 없도록 지정

ldg0 도메인에서 레지스터 세트에 액세스할 수 없도록 지정합니다.

```
primary# ldm set-domain perf-counters= ldg0
```

예 73 성능 액세스 정보 보기

다음 예제는 ldm list -o domain 명령을 사용하여 성능 액세스 정보를 보는 방법을 보여줍니다.

- 다음 ldm list -o domain 명령은 ldg0 도메인에 global 및 htstrand 성능 값이 지정되었음을 보여줍니다.

```
primary# ldm list -o domain ldg0
NAME      STATE    FLAGS    UTIL
NORM
ldg0      active  -n----   0.0% 0.0%

SOFTSTATE
Solaris running

UUID
062200af-2de2-e05f-b271-f6200fd3eee3

HOSTID
0x84fb315d

CONTROL
failure-policy=ignore
extended-mapin-space=on
cpu-arch=native
rc-add-policy=
```

```

shutdown-group=15
perf-counters=global,htstrand

DEPENDENCY
  master=

PPRIORITY    4000

VARIABLES
  auto-boot?=false
  boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
    /virtualdevices@100/channel@200/disk@0
  network-boot-arguments=dhcp,hostname=solaris,
    file=http://10.129.241.238:5555/cgibin/wanboot-cgi
  pm_boot_policy=disabled=0;tftc=2000;ttr=0;

```

- 다음 `ldm list -p -o domain` 명령은 이전 예제와 동일한 정보를 보여주되, 구문 분석 가능한 형식으로 표시합니다.

```

primary# ldm list -p -o domain ldg0
VERSION 1.12
DOMAIN|name=ldg0|state=active|flags=normal|util=|norm_util=
UUID|uuid=4e8749b9-281b-e2b1-d0e2-ef4dc2ce5ce6
HOSTID|hostid=0x84f97452
CONTROL|failure-policy=reset|extended-mapin-space=on|cpu-arch=native|rc-add-policy=|
shutdown-group=15|perf-counters=global,htstrand
DEPENDENCY|master=
VARIABLES
|auto-boot?=false
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
|pm_boot_policy=disabled=0;tftc=2500000;ttr=0;

```

리소스 관리 문제

도메인에서 많은 수의 CPU를 제거할 경우 실패할 수 있음

게스트 도메인에서 많은 수의 CPU를 제거하려고 시도하면 다음과 같은 오류 메시지가 표시될 수 있습니다.

```

Request to remove cpu(s) sent, but no valid response received
VCPU(s) will remain allocated to the domain, but might
not be available to the guest OS
Resource modification failed

```

이 문제를 방지하려면 도메인에서 CPU를 한 번에 100개 미만으로 제거하십시오.

때때로 동적으로 추가된 메모리 블록을 전체로서만 동적으로 제거할 수 있음

Oracle Solaris OS에서 동적으로 추가된 메모리를 관리하기 위해 메타데이터를 처리하는 방법 때문에, 이전에 동적으로 추가된 메모리를 나중에 제거할 때 적절한 하위 세트가 아닌 전체 블록만 제거할 수 있습니다.

이러한 상황은 다음 예제에 표시된 것처럼 메모리 크기가 작은 도메인이 훨씬 큰 크기로 동적으로 증가할 경우에 발생할 수 있습니다.

```
primary# ldm list ldom1
NAME STATE FLAGS  CONS VCPU MEMORY UTIL UPTIME
ldom1 active -n--  5000 2    2G    0.4% 23h

primary# ldm add-mem 16G ldom1

primary# ldm rm-mem 8G ldom1
Memory removal failed because all of the memory is in use.

primary# ldm rm-mem 16G ldom1

primary# ldm list ldom1
NAME STATE FLAGS  CONS VCPU MEMORY UTIL UPTIME
ldom1 active -n--  5000 2    2G    0.4% 23h
```

이 문제를 해결하려면 `ldm add-mem` 명령을 사용하여 나중에 제거할 수 있는 것보다 큰 청크가 아닌 작은 청크로 메모리를 순차적으로 추가합니다.

이 문제가 발생하면 다음 조치 중 하나를 수행하십시오.

- 도메인을 중지하고 메모리를 제거한 후 도메인을 다시 시작합니다.
- 도메인을 재부트합니다. 그러면 Oracle Solaris OS에서 메모리 관리 메타데이터를 재할당하므로 이전에 추가된 메모리를 작은 청크로 동적으로 제거할 수 있습니다.

도메인 구성 관리

이 장에서는 도메인 구성 관리에 대한 정보를 제공합니다.

이 장에서는 다음 주제를 다룹니다.

- “도메인 구성 관리” [337]
- “사용 가능한 구성 복구 방법” [338]
- “구성 관리 문제” [344]

도메인 구성 관리

도메인 구성은 단일 시스템 내의 모든 도메인과 해당 리소스 지정으로 완전히 설명됩니다. 나중에 사용할 수 있도록 서비스 프로세서(SP)에 구성을 저장할 수 있습니다.

구성을 SP에 저장하면 시스템의 전원을 껐다 켜도 구성이 유지됩니다. 구성을 여러 개 저장하고 다음 번에 전원 켜기를 시도할 때 부트할 구성을 지정할 수 있습니다.

시스템 전원을 켤 때 SP가 선택한 구성을 부트합니다. 시스템이 구성에 지정된 것과 동일한 도메인 세트를 실행하고 동일한 가상화 및 분할 리소스 할당을 사용합니다. 기본 구성은 가장 최근에 저장된 구성입니다. `ldm set-spconfig` 명령 또는 적절한 ILOM 명령을 사용하여 다른 구성을 명시적으로 요청할 수도 있습니다.



주의 - 항상 안정적인 구성을 XML로 SP에 저장하십시오. 이와 같은 방식으로 구성을 저장하면 전원 장애 후 시스템 구성을 복구하고 나중에 사용할 수 있도록 저장할 수 있습니다. “[도메인 구성 저장](#)” [340]을 참조하십시오.

구성을 SP에 저장할 때마다 SP 구성 및 Logical Domains 제약 조건 데이터베이스의 로컬 복사본이 컨트롤 도메인에 저장됩니다. 이 로컬 복사본을 부트 세트라고 합니다. 부트 세트는 시스템의 전원을 껐다 켤 때 해당 Logical Domains 제약 조건 데이터베이스를 로드하는 데 사용됩니다.

SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버, SPARC S7 시리즈 서버 및 Fujitsu M10 서버에서는 컨트롤 도메인의 부트 세트가 구성의 마스터 복사본입니다. 시작 시 Logical Domains Manager가 모든 구성을 SP

와 자동으로 동기화합니다. 따라서 SP의 구성이 항상 컨트롤 도메인에 저장된 부트 세트와 동일하게 유지됩니다.

주 - 부트 세트에는 중요한 시스템 데이터가 포함되므로 컨트롤 도메인의 파일 시스템은 디스크 미러링 또는 RAID와 같은 기술을 사용하여 디스크 오류의 영향을 줄일 수 있습니다.

physical domain(물리적 도메인)은 단일 Oracle VM Server for SPARC 인스턴스가 관리하는 리소스의 범위입니다. 지원되는 SPARC T-Series 서버 및 SPARC S7 시리즈 서버와 마찬가지로 물리적 도메인은 전체 물리적 시스템일 수 있습니다. 또는 지원되는 SPARC M-Series 서버와 마찬가지로 전체 시스템일 수도 있고, 시스템의 하위 세트일 수도 있습니다.

사용 가능한 구성 복구 방법

Oracle VM Server for SPARC는 다음과 같은 구성 복구 방법을 지원합니다.

- 자동 저장 방법 - SP에서 구성을 사용할 수 없을 경우 사용됩니다.
다음과 같은 경우 중 하나에서 이 상황이 발생할 수 있습니다.
 - 저장된 구성을 보관하는 하드웨어가 교체된 경우
 - 최신 구성 변경사항을 SP에 저장하지 못했거나 예상치 않은 전원 켜다 켜기 동작이 발생하여 구성이 최신이 아닌 경우
- `ldm add-domain` 방법 - 도메인 하위 세트의 구성을 복원해야 할 경우 사용됩니다.
- `ldm init-system` 방법 - 최후의 방법으로만 사용되어야 합니다. SP의 구성과 컨트롤 도메인의 자동 저장 정보가 손실된 경우에만 이 방법을 사용하십시오.

자동 저장을 사용하여 구성 복원

도메인 구성을 변경할 때마다 현재 구성의 복사본이 컨트롤 도메인에 자동으로 저장됩니다. 이 자동 저장 작업은 구성을 명시적으로 SP에 저장하지 않습니다.

다음 경우에도 자동 저장 작업이 즉시 발생합니다.

- 새 구성이 SP에 명시적으로 저장되지 않은 경우
- 영향을 받는 도메인이 재부트될 때까지 구성이 변경되지 않은 경우

이 자동 저장 작업은 SP에 저장된 구성이 손실된 경우 구성을 복구할 수 있습니다. 또한 이 작업으로 인해 현재 구성이 SP에 명시적으로 저장되지 않은 경우 시스템의 전원을 켜다 켜 후 구성을 복구할 수 있습니다. 이러한 상황에서 Logical Domains Manager는 해당 구성이 다음 부트용으로 표시된 구성보다 최신인 경우 다시 시작 시 구성을 복원할 수 있습니다.

주 - 전원 관리, FMA, ASR 이벤트로 자동 저장 파일이 업데이트되지는 않습니다.

자동 저장 파일을 새 구성이나 기존 구성으로 자동/수동으로 복원할 수 있습니다. 기본적으로 자동 저장 구성이 실행 중인 구성보다 최신인 경우 Logical Domains 로그에 메시지가 기록됩니다. 따라서 `ldm add-spconfig -r` 명령을 사용하여 기존 구성을 수동으로 업데이트하거나 자동 저장 데이터를 기반으로 새 구성을 만들어야 합니다. 이 명령을 사용한 후에 수동 복구를 완료하려면 전원을 껐다 켜야 합니다.

주 - 지연된 재구성이 보류 중인 경우 구성 변경사항이 즉시 자동 저장됩니다. 그 결과, `ldm list-config -r` 명령을 실행하면 자동 저장 구성이 현재 구성보다 최신인 것으로 나타납니다.

`ldm *-spconfig` 명령을 사용하여 구성을 관리하고 자동 저장 파일을 수동으로 복구하는 방법은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

부트할 구성을 선택하는 방법은 “[서비스 프로세서와 함께 Oracle VM Server for SPARC 사용](#)” [355]을 참조하십시오. [ldm\(1M\)](#) 매뉴얼 페이지에 설명된 `ldm set-spconfig` 명령을 사용할 수도 있습니다.

자동 복구 정책

자동 복구 정책은 컨트롤 도메인에 자동으로 저장된 구성이 실행 중인 구성보다 최신인 경우 구성의 복구 처리 방법을 지정합니다. `ldmd` SMF 서비스의 `autorecovery_policy` 등록 정보를 설정하여 자동 복구 정책을 지정합니다. 이 등록 정보에는 다음 값이 사용될 수 있습니다.

- `autorecovery_policy=1` – 자동 저장 구성이 실행 중인 구성보다 최신일 때 경고 메시지를 기록합니다. 이러한 메시지는 `ldmd` SMF 로그 파일에 기록됩니다. 구성 복구를 수동으로 수행해야 합니다. 이것이 기본 정책입니다.
- `autorecovery_policy=2` – 자동 저장 구성이 실행 중인 구성보다 최신일 때 통지 메시지를 표시합니다. 이 통지 메시지는 Logical Domains Manager를 다시 시작한 후 처음 `ldm` 명령을 실행할 때 `ldm` 명령의 출력에 인쇄됩니다. 구성 복구를 수동으로 수행해야 합니다.
- `autorecovery_policy=3` – 자동 저장 구성이 실행 중인 구성보다 최신일 때 구성을 자동으로 업데이트합니다. 이렇게 하면 다음에 전원을 껐다 켜는 동안 사용될 SP 구성이 겹쳐쓰입니다. 이 구성이 사용 가능하려면 한번 더 전원을 껐다 켜야 합니다. 이 구성은 컨트롤 도메인에 저장된 최신 구성으로 업데이트됩니다. 이 작업은 현재 실행 중인 구성에 영향을 주지 않습니다. 다음 번에 전원을 껐다 켜는 동안 사용될 구성에만 영향을 줍니다. 또한 최신 구성이 SP에 저장되었으며 다음 번에 시스템의 전원을 껐다 켤 때 부트될 것임을 알리는 메시지가 기록됩니다. 이러한 메시지는 `ldmd` SMF 로그 파일에 기록됩니다.

▼ 자동 복구 정책을 수정하는 방법

1. 컨트롤 도메인에 로그인합니다.
2. 관리자로 전환합니다.

Oracle Solaris 11.3의 경우, [Securing Users and Processes in Oracle Solaris 11.3](#)의 1장, “About Using Rights to Control Users and Processes”를 참조하십시오.

3. `autorecovery_policy` 등록 정보 값을 확인합니다.

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy
```

4. `ldmd` 서비스를 중지합니다.

```
# svcadm disable ldmd
```

5. `autorecovery_policy` 등록 정보 값을 변경합니다.

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=value
```

예를 들어, 자동 복구를 수행할 정책을 설정하려면 등록 정보 값을 3으로 설정합니다.

```
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3
```

6. `ldmd` 서비스를 새로고침하고 다시 시작합니다.

```
# svcadm refresh ldmd  
# svcadm enable ldmd
```

예 74 자동 복구 정책을 로그에서 자동 복구로 수정

다음 예제는 `autorecovery_policy` 등록 정보의 현재 값을 확인하고 새 값으로 변경하는 방법을 보여줍니다. 이 등록 정보의 원래 값은 1이며, 자동 저장 변경사항이 기록됨을 의미합니다. `svcadm` 명령을 사용하여 `ldmd` 서비스를 중지 후 다시 시작하고, `svccfg` 명령을 사용하여 등록 정보 값을 확인하고 설정할 수 있습니다.

```
# svccfg -s ldmd listprop ldmd/autorecovery_policy  
ldmd/autorecovery_policy integer 1  
# svcadm disable ldmd  
# svccfg -s ldmd setprop ldmd/autorecovery_policy=3  
# svcadm refresh ldmd  
# svcadm enable ldmd
```

도메인 구성 저장

단일 도메인 또는 시스템의 모든 도메인에 대한 도메인 구성을 저장할 수 있습니다.

명명된 물리적 리소스를 제외하고, 다음 방법은 실제 바인딩을 보존하지 않습니다. 그러나 해당 바인딩을 만드는 데 사용된 제약 조건은 보존합니다. 구성을 저장 및 복원한 후 도메인은 동일한 가상 리소스를 보유하지만 반드시 동일한 물리적 리소스에 바인드되는 것은 아닙니다. 명명된 물리적 리소스는 관리자가 지정한 대로 바인드됩니다.

- 단일 도메인에 대한 구성을 저장하려면 도메인의 제약 조건을 포함하는 XML 파일을 만듭니다.

```
# ldm list-constraints -x domain-name >domain-name.xml
```

다음 예제는 ldg1 도메인의 제약 조건을 포함하는 XML 파일 ldg1.xml을 만드는 방법을 보여줍니다.

```
# ldm list-constraints -x ldg1 >ldg1.xml
```

- 시스템의 모든 도메인에 대한 구성을 저장하려면 모든 도메인의 제약 조건을 포함하는 XML 파일을 만듭니다.

```
# ldm list-constraints -x >file.xml
```

다음 예제는 시스템의 모든 도메인에 대한 제약 조건을 포함하는 XML 파일 config.xml을 만드는 방법을 보여줍니다.

```
# ldm list-constraints -x >config.xml
```

도메인 구성 복원

이 절에서는 게스트 도메인 및 컨트롤(primary) 도메인에 대한 XML 파일에서 도메인 구성을 복원하는 방법에 대해 설명합니다.

- 게스트 도메인에 대한 도메인 구성을 복원하려면 [XML 파일에서 도메인 구성을 복원하는 방법\(ldm add-domain\) \[341\]](#)에 설명된 대로 `ldm add-domain -i` 명령을 사용합니다. XML 파일에 primary 도메인의 제약 조건을 저장할 수는 있지만 해당 파일을 이 명령의 입력으로 사용할 수 없습니다.
- primary 도메인에 대한 도메인 구성을 복원하려면 `ldm init-system` 명령 및 XML 파일의 리소스 제약 조건을 사용하여 primary 도메인을 재구성합니다. 또한 `ldm init-system` 명령을 사용하여 XML 파일에 설명된 다른 도메인을 재구성할 수 있습니다. 단, 해당 도메인은 구성이 완료될 때 비활성 상태로 유지될 수 있습니다. [XML 파일에서 도메인 구성을 복원하는 방법\(ldm init-system\) \[342\]](#)을 참조하십시오.

▼ XML 파일에서 도메인 구성을 복원하는 방법(ldm add-domain)

이 절차는 게스트 도메인에 대해서만 수행할 수 있으며 컨트롤(primary) 도메인에 대해서는 수행할 수 없습니다. primary 도메인 또는 XML 파일에 설명된 다른 도메인에 대한 구성을 복원하려면 [XML 파일에서 도메인 구성을 복원하는 방법\(ldm init-system\) \[342\]](#)을 참조하십시오.

1. 입력으로 만든 XML 파일을 사용하여 도메인을 만듭니다.

```
# ldm add-domain -i domain-name.xml
```

2. 도메인을 바인드합니다.

```
# ldm bind-domain [-fq] domain-name
```

`-f` 옵션은 잘못된 백엔드 장치가 감지된 경우에도 강제로 도메인을 바인딩합니다. `-q` 옵션은 명령이 더 빠르게 실행되도록 백엔드 장치의 검증을 사용 안함으로 설정합니다.

3. 도메인을 시작합니다.

```
# ldm start-domain domain-name
```

예 75 XML 파일에서 단일 도메인 복원

다음 예제는 단일 도메인을 복원하는 방법을 보여줍니다. 먼저, XML 파일에서 ldg1 도메인을 복원합니다. 그 다음, 복원한 ldg1 도메인을 바인딩하고 다시 시작합니다.

```
# ldm add-domain -i ldg1.xml
# ldm bind ldg1
# ldm start ldg1
```

▼ XML 파일에서 도메인 구성을 복원하는 방법(ldm init-system)

이 절차에서는 XML 파일과 함께 ldm init-system 명령을 사용하여 이전에 저장된 구성을 다시 만드는 방법을 설명합니다.



주의 - ldm init-system 명령은 물리적 I/O 명령이 사용된 구성을 제대로 복원하지 못할 수 있습니다. 해당 명령은 ldm add-io, ldm set-io, ldm remove-io, ldm create-vf 및 ldm destroy-vf입니다. 자세한 내용은 [Oracle VM Server for SPARC 3.4 릴리스 노트](#)의 "ldm init-system 명령이 물리적 I/O 변경을 수행한 도메인 구성을 올바르게 복원하지 못함"을 참조하십시오.

시작하기 전에 ldm list-constraints -x 명령을 실행하여 XML 구성 파일을 만들었어야 합니다. 해당 파일은 하나 이상의 도메인 구성을 설명해야 합니다.

1. primary 도메인에 로그인합니다.
2. 시스템이 factory-default 구성인지 확인합니다.

```
primary# ldm list-config | grep "factory-default"
factory-default [current]
```

시스템이 factory-default 구성이 아닌 경우 [출하 시 기본 구성을 복원하는 방법 \[375\]](#)을 참조하십시오.

3. 관리자로 전환합니다.
Oracle Solaris 11.3의 경우, [Securing Users and Processes in Oracle Solaris 11.3](#)의 1장, "About Using Rights to Control Users and Processes"를 참조하십시오.
4. XML 파일에서 도메인 구성을 복원합니다.

```
# ldm init-system [-frs] -i filename.xml
```

구성을 적용하려면 primary 도메인을 재부트해야 합니다. -r 옵션은 구성 후 primary 도메인을 재부트합니다. -r 옵션을 지정하지 않을 경우 수동으로 재부트를 수행해야 합니다.

-s 옵션은 가상 서비스 구성(vds, vcc, vsw)만 복원하고 재부트 없이 수행할 수 있습니다.

-f 옵션은 factory-default 구성 검사를 건너뛰고 시스템에 이미 구성된 사항에 관계없이 작업을 계속합니다. -f 옵션은 주의해서 사용하십시오. ldm init-system 명령은 시스템이 factory-default 구성 상태라고 가정하므로 XML 파일에 의해 지정된 변경사항을 바로 적용합니다. 시스템이 factory-default 이외의 구성일 때 -f 옵션을 사용하면 XML 파일에 지정된 대로 시스템이 구성되지 않습니다. XML 파일과 초기 구성의 변경사항 조합에 따라 하나 이상의 변경사항이 시스템에 적용되지 못할 수 있습니다.

primary 도메인은 파일에 지정된 대로 재구성됩니다. XML 파일에 구성이 있는 모든 비primary 도메인은 재구성되지만 비활성 상태로 유지됩니다.

예 76 XML 구성 파일에서 도메인 복원

다음 예제는 ldm init-system 명령을 사용하여 factory-default 구성에서 primary 도메인 및 시스템의 모든 구성을 복원하는 방법을 보여줍니다.

- primary 도메인 복원. -r 옵션을 사용하여 구성이 완료된 후 primary 도메인을 재부트합니다. primary.xml 파일은 이전에 저장한 XML 도메인 구성을 포함합니다.

```
primary# ldm init-system -r -i primary.xml
```

- 시스템의 모든 도메인 복원. 시스템의 도메인을 config.xml XML 파일의 구성으로 복원합니다. config.xml 파일은 이전에 저장한 XML 도메인 구성을 포함합니다. primary 도메인은 ldm init-system 명령에 의해 자동으로 다시 시작됩니다. 다른 모든 도메인은 복원되지만 바인드 후 다시 시작되지 않습니다.

```
# ldm init-system -r -i config.xml
```

시스템을 재부트한 후 다음 명령이 ldg1 및 ldg2 도메인을 바인드하고 다시 시작합니다.

```
# ldm bind ldg1
# ldm start ldg1
# ldm bind ldg2
# ldm start ldg2
```

서비스 프로세서 연결 문제 해결

SPARC T7 시리즈 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에서 ILOM 상호 연결은 ldmd 서비스와 SP 간 통신에 사용됩니다.

ldm 명령을 사용하여 도메인 구성을 관리하려고 시도할 때 SP와 통신 오류 때문에 실패하면 해당 플랫폼에 적용되는 다음 복구 단계를 수행합니다.

- **SPARC T7 시리즈 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버:** ILOM 상호 연결 상태를 확인하고 ilomconfig-interconnect 서비스를 다시 사용으로 설정합니다. [ILOM 상호 연결 구성을 확인하는 방법 \[40\]](#) 및 [ILOM 상호 연결 서비스를 다시 사용으로 설정하는 방법 \[40\]](#)을 참조하십시오.

- **SPARC T3, SPARC T4, SPARC T5, SPARC M5 및 SPARC M6 서버:** `ldmd` 서비스를 다시 시작합니다.

```
primary# svcadm enable ldmd
```

이 단계로 통신을 복원하지 못하면 SP를 다시 시작합니다.

구성 관리 문제

`init-system`이 저장된 XML 파일에서 게스트 도메인의 이름이 지정된 코어 제약 조건을 복원하지 않음

도메인에 이름이 지정된 코어 리소스를 지정한 경우 `ldm init-system` 명령을 사용하면 해당 도메인에 이름이 지정된 리소스를 재지정하지 못할 수 있습니다. `ldm init-system` 명령이 `primary` 도메인에 대한 지연된 재구성을 시작하고 지연된 재구성마다 가상 CPU 작업 하나만 수행할 수 있기 때문에 이런 일이 발생할 수 있습니다. 따라서 저장된 XML 파일에서 게스트 도메인의 이름이 지정된 CPU 코어 제약 조건을 복원하기 위한 이 명령이 실패합니다.

해결 방법: 다음 단계를 수행합니다.

1. 기본 도메인에 대한 XML 파일을 만듭니다.

```
primary# ldm list-constraints -x primary > primary.xml
```

2. 게스트 도메인에 대한 XML 파일을 만듭니다.

```
primary# ldm list-constraints -x domain-name[, domain-name][,...] > guest.xml
```

3. 시스템을 껐다 켜고 출하 시 기본 구성으로 부트합니다.

4. `primary` 도메인에 XML 구성을 적용합니다.

```
primary# ldm init-system -r -i primary.xml
```

5. 게스트 도메인에 XML 구성을 적용합니다.

```
primary# ldm init-system -f -i guest.xml
```

◆◆◆ 17 장

하드웨어 오류 처리

이 장에서는 Oracle VM Server for SPARC의 하드웨어 오류 처리 방법에 대한 정보를 제공합니다.

이 장에서는 다음 주제를 다룹니다.

- “하드웨어 오류 처리 개요” [345]
- “FMA를 사용하여 결함이 있는 리소스 차단 또는 구성 해제” [345]
- “결함이 있거나 누락된 리소스 감지 후 도메인 복구” [347]
- “도메인을 저하됨으로 표시” [350]
- “I/O 리소스를 비워짐으로 표시” [350]

하드웨어 오류 처리 개요

Oracle VM Server for SPARC 소프트웨어는 SPARC T5, SPARC M5 및 SPARC S7 시리즈 서버부터 SPARC 엔터프라이즈급 플랫폼에 대한 다음 RAS 기능을 추가합니다.

- **FMA(Fault Management Architecture)** 차단. FMA가 결함이 있는 CPU 또는 메모리 리소스를 감지하면 Oracle VM Server for SPARC가 해당 요소를 차단 목록에 지정합니다. 차단 목록에 지정된 결함이 있는 리소스는 FMA가 손상 복구된 것으로 표시할 때까지 도메인에 재지정될 수 없습니다.
- 복구 모드. 결함이 있거나 누락된 리소스로 인해 부트될 수 없는 도메인 구성을 자동으로 복구합니다.

Fujitsu M10 플랫폼도 복구 모드를 지원합니다. 결함이 있는 리소스의 차단을 지원하지 않지만 Fujitsu M10 플랫폼 자동 교체 기능이 유사한 기능을 제공합니다.

FMA를 사용하여 결함이 있는 리소스 차단 또는 구성 해제

FMA는 결함이 있는 리소스를 감지할 때 Logical Domains Manager에 연락합니다. 그러면 Logical Domains Manager는 실행 중인 모든 도메인에서 해당 리소스의 사용을 중지하려고

시도합니다. 나중에 결함이 있는 리소스가 도메인에 지정될 수 없도록 FMA는 차단 목록에 리소스를 추가합니다.

Logical Domains Manager는 CPU 및 메모리 리소스에 대해서만 차단을 지원하며 I/O 리소스에 대해서는 차단을 지원하지 않습니다.

결함이 있는 리소스가 사용되고 있지 않을 경우 Logical Domains Manager는 `ldm list-devices` 출력에 표시되는 사용 가능한 리소스 목록에서 해당 리소스를 제거합니다. 이 단계에서 해당 리소스는 나중에 도메인에 재지정될 수 없도록 내부적으로 "차단됨"으로 표시됩니다.

결함이 있는 리소스가 사용되고 있을 경우 Logical Domains Manager는 리소스를 비우려고 시도합니다. 실행 중인 도메인에서 서비스 중단이 발생하지 않도록 Logical Domains Manager는 먼저 CPU 또는 메모리 동적 재구성을 사용하여 결함이 있는 리소스를 비우려고 시도합니다. 코어를 대상으로 사용할 수 있을 경우 Logical Domains Manager는 결함이 있는 코어를 다시 매핑합니다. 이와 같은 "실시간 비우기"를 성공할 경우 결함이 있는 리소스는 내부적으로 차단됨으로 표시되며, 나중에 도메인에 지정되지 않도록 `ldm list-devices` 출력에 표시되지 않습니다.

실시간 비우기를 실패할 경우 Logical Domains Manager는 결함이 있는 리소스를 내부적으로 "비우기 보류"로 표시합니다. 그러면 리소스는 영향을 받는 게스트 도메인이 재부트되거나 중지될 때까지 실행 중인 도메인에서 계속 사용되므로 `ldm list-devices` 출력에 일반적으로 표시됩니다.

영향을 받는 게스트 도메인이 중지되거나 재부트되면 Logical Domains Manager는 결함이 있는 리소스를 비우려고 시도하고 나중에 리소스가 지정될 수 없도록 내부적으로 차단됨으로 표시합니다. 해당 장치는 `ldm` 출력에 표시되지 않습니다. 보류 중인 비우기가 완료된 후 Logical Domains Manager는 게스트 도메인을 시작하려고 시도합니다. 단, 사용 가능한 리소스가 부족하여 게스트 도메인을 시작할 수 없을 경우 게스트 도메인이 "저하됨"으로 표시되고 사용자 개입을 통해 수동 복구가 수행될 수 있도록 다음과 같은 경고 메시지가 기록됩니다.

```
primary# ldm ls
NAME          STATE   FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary      active  -n-cv-  UART  368   2079488M 0.1%  0.0%  16h 57m
gd0          bound   -d- - -  5000   8
```

```
warning: Could not restart domain gd0 after completing pending evacuation.
The domain has been marked degraded and should be examined to see
if manual recovery is possible.
```

시스템의 전원을 껐다 켜면 FMA는 아직 결함이 있는 상태인 리소스에 대해 비우기 요청을 반복하고 Logical Domains Manager는 결함이 있는 리소스를 비우고 내부적으로 차단됨으로 표시하여 해당 요청을 처리합니다.

FMA 차단이 지원되기 전의 버전에서는 결함이 있는 리소스로 인해 패닉이 발생한 게스트 도메인에서 패닉-재부트가 계속 반복될 수도 있습니다. 게스트 도메인이 재부트될 때 리소스 비우기 및 차단을 사용하면 이와 같은 패닉-재부트가 반복되지 않도록 방지하고 나중에 결함이 있는 리소스가 사용되지 않도록 할 수 있습니다.

결함이 있거나 누락된 리소스 감지 후 도메인 복구

전원이 켜질 때 SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버, SPARC S7 시리즈 서버 또는 Fujitsu M10 서버가 결함이 있거나 누락된 리소스를 감지할 경우 Logical Domains Manager는 사용 가능한 나머지 리소스를 사용하여 구성된 도메인을 복구하려고 시도합니다. 복구가 수행되는 동안 시스템(또는 SPARC M5, SPARC M6 및 SPARC M7 시리즈 서버의 물리적 도메인)은 복구 모드 상태입니다. 복구 모드는 기본적으로 사용으로 설정됩니다. [“복구 모드 제어” \[350\]](#)를 참조하십시오.

다음과 같은 경우 마지막으로 선택된 전원 켜기 구성을 부트할 수 없으면 전원이 켜질 때 시스템 펌웨어가 factory-default 구성으로 되돌아갑니다.

- 구성의 각 PCIe 스위치 내 I/O 토폴로지가 마지막으로 선택된 전원 켜기 구성의 I/O 토폴로지와 일치하지 않는 경우
- 마지막으로 선택된 전원 켜기 구성의 CPU 리소스 또는 메모리 리소스가 더 이상 시스템에 존재하지 않는 경우

복구 모드가 사용으로 설정된 경우 Logical Domains Manager는 마지막으로 선택된 전원 켜기 구성에서 모든 활성 도메인과 바인드된 도메인을 복구합니다. 결과로 나타나는 실행 중인 구성을 저하된 구성이라고 합니다. 저하된 구성은 SP에 저장되며 새 SP 구성이 저장되거나 물리적 도메인의 전원을 껐다 켤 때까지 활성 구성 상태로 유지됩니다.

주 - 물리적 도메인의 경우 복구 후 저하된 구성을 활성화하기 위해 전원을 껐다 켤 필요가 없습니다. 이미 실행 중인 구성이기 때문입니다.

물리적 도메인의 전원을 껐다 켜면 시스템 펌웨어는 먼저 마지막으로 선택된 원래 전원 켜기 구성을 부트하려고 시도합니다. 그 동안에 누락되거나 결함이 있는 하드웨어가 교체된 경우 시스템은 원래 일반 구성을 부트할 수 있습니다. 마지막으로 선택된 전원 켜기 구성을 부트할 수 없을 경우 다음으로 펌웨어는 연관되어 있는 저하된 구성(있을 경우)을 부트하려고 시도합니다. 저하된 구성을 부트할 수 없거나 해당 구성이 없을 경우 factory-default 구성이 부트되며 복구 모드가 호출됩니다.

복구 작업은 다음 순서로 작동합니다.

- **컨트롤 도메인.** Logical Domains Manager가 CPU, 메모리 및 I/O 구성과 해당 가상 I/O 서비스를 복원하여 컨트롤 도메인을 복구합니다.

복구 가능한 모든 도메인에 필요한 CPU 또는 메모리 양이 사용 가능한 나머지 양보다 클 경우 다른 도메인의 크기에 비례하여 CPU, 코어 또는 메모리 수가 줄어듭니다. 예를 들어, 각 도메인에 25%의 CPU 및 메모리가 지정된 4-도메인 시스템에서 결과로 나타나는 저하된 구성은 계속해서 각 도메인에 25%의 CPU 및 메모리를 지정합니다. primary 도메인이 원래 최대 2개의 코어(16개의 가상 CPU)와 8GB 메모리로 구성된 경우 컨트롤 도메인 크기가 줄어들지 않습니다.

다른 도메인에 지정된 루트 컴플렉스 및 PCIe 장치가 컨트롤 도메인에서 제거됩니다. 컨트롤 도메인 소유의 루트 컴플렉스에 있는 가상 기능이 다시 만들어집니다. 컨트롤 도메인에

지정되어 있는 누락된 루트 컴플렉스, PCIe 장치, 물리적 기능 또는 가상 기능이 비워짐으로 표시됩니다. 그런 다음 Logical Domains Manager가 컨트롤 도메인을 재부트하여 변경사항을 활성화합니다.

- **루트 도메인.** 컨트롤 도메인이 재부트된 후 Logical Domains Manager가 루트 도메인을 복구합니다. 필요한 경우 다른 복구 가능 도메인에 비례하여 CPU 및 메모리 양이 줄어듭니다. 루트 컴플렉스가 시스템에 더 이상 실제로 존재하지 않을 경우 비워짐으로 표시됩니다. 이 루트 컴플렉스는 복구 작업 중 도메인에 구성되지 않습니다. 루트 도메인에 지정된 루트 컴플렉스 중 하나 이상을 사용할 수 있을 경우 루트 도메인이 복구됩니다. 사용 가능한 루트 컴플렉스가 없을 경우 루트 도메인이 복구되지 않습니다. Logical Domains Manager는 루트 도메인을 부트하고 루트 도메인 소유의 물리적 기능에 가상 기능을 다시 만듭니다. 또한 루트 도메인에서 임대된 PCIe 슬롯을 제거합니다. 누락된 PCIe 슬롯, 물리적 기능 및 가상 기능이 비워짐으로 표시됩니다. 가능한 경우 도메인이 제공하는 가상 I/O 서비스가 다시 만들어집니다.
- **I/O 도메인.** Logical Domains Manager가 모든 I/O 도메인을 복구합니다. 시스템에서 누락된 모든 PCIe 슬롯 및 가상 기능이 비워짐으로 표시됩니다. 필요한 I/O 장치가 없을 경우 도메인이 복구되지 않으며 다른 도메인이 해당 CPU 및 메모리 리소스를 사용할 수 있습니다. 가능한 경우 도메인이 제공하는 가상 I/O 서비스가 다시 만들어집니다.
- **게스트 도메인.** 게스트 도메인은 도메인을 사용하는 서비스 도메인 중 하나 이상이 복구된 경우에만 복구됩니다. 게스트 도메인을 복구할 수 없을 경우 다른 게스트 도메인이 해당 CPU 및 메모리 리소스를 사용할 수 있습니다.

가능한 경우 원래 구성에 지정된 것과 동일한 수의 CPU와 메모리 양이 도메인에 할당됩니다. 해당 수의 CPU 또는 메모리 양을 사용할 수 없을 경우 사용 가능한 나머지 리소스를 소비하기 위해 비례적으로 해당 리소스가 줄어듭니다. 도메인에 이름이 지정된 리소스를 지정했고 나중에 복구 모드에서 복구되는 경우 이름이 지정된 리소스를 해당 도메인에 재지정하려고 시도하지 않습니다.

주 - 시스템이 복구 모드인 경우 `ldm list-*` 명령만 실행할 수 있습니다. 기타 모든 `ldm` 명령은 복구 작업이 완료될 때까지 사용 안함으로 설정됩니다.

Logical Domains Manager는 바인드된 도메인 및 활성 도메인만 복구하려고 시도합니다. 바인드가 해제된 도메인의 기존 리소스 구성은 새 구성에 그대로 복사됩니다.

복구 작업 중에는 이전에 부트된 구성에서보다 적은 양의 리소스를 사용할 수 있습니다. 따라서 Logical Domains Manager가 이전에 구성된 도메인 중 일부만 복구할 수도 있습니다. 또한 복구된 도메인에 원래 구성의 모든 리소스가 포함되지 않을 수 있습니다. 예를 들어, 복구된 바인드된 도메인의 I/O 리소스가 이전 구성에서보다 적을 수 있습니다. I/O 장치가 더 이상 존재하지 않거나 상위 서비스 도메인을 복구할 수 없을 경우 도메인이 복구되지 않을 수 있습니다.

복구 모드는 Logical Domains Manager SMF 로그 `/var/svc/log/ldoms-ldmd:default.log`에 단계를 기록합니다. Logical Domains Manager가 복구를 시작하고 컨트롤 도메인을 재부트할 때, 그리고 복구가 완료될 때 시스템 콘솔에 메시지가 기록됩니다.



주의 - 복구된 도메인이 제대로 작동하지 않을 수도 있습니다. OS 인스턴스 또는 응용 프로그램을 실행하는 데 필요한 리소스가 도메인에 포함되어 있지 않을 수도 있습니다. 예를 들어, 복구된 도메인에 네트워크 리소스만 포함되고 디스크 리소스가 포함되지 않을 수 있습니다. 또는 복구된 도메인에 응용 프로그램을 실행하는 데 필요한 파일 시스템이 누락될 수 있습니다. 도메인에 다중 경로가 지정된 I/O를 사용하면 누락된 I/O 리소스의 영향이 줄어듭니다.

복구 모드 하드웨어 및 소프트웨어 요구사항

- 하드웨어 요구사항 - 복구 모드 기능은 SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버, SPARC S7 시리즈 서버 및 Fujitsu M10 서버에서 지원됩니다.
- 펌웨어 요구사항 - SPARC T5 서버, SPARC M5 서버 및 SPARC M6 서버에는 최소 9.1.0.a 버전의 시스템 펌웨어가 필요합니다. SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버에는 최소 9.4.3 버전의 시스템 펌웨어가 필요합니다. SPARC S7 시리즈 서버에 대한 시스템 펌웨어의 릴리스된 모든 버전입니다. Fujitsu M10 서버에는 최소 XCP2230 버전의 시스템 펌웨어가 필요합니다.
- 소프트웨어 요구사항 - PCIe 슬롯을 임대한 비_{primary} 루트 도메인에서 최소한 Oracle Solaris 10 1/13 OS 또는 Oracle Solaris 11.2 OS를 실행해야 합니다.

저하된 구성

각 물리적 도메인의 저하된 구성은 하나만 SP에 저장될 수 있습니다. 저하된 구성이 이미 있을 경우 새로 만들어진 저하된 구성으로 바꿉니다.

저하된 구성과 직접 상호 작용할 수 없습니다. 필요한 경우 시스템 펌웨어는 다음 전원 켜기 구성의 저하된 버전을 투명하게 부트합니다. 이 투명성을 통해 시스템은 누락된 리소스가 다시 나타나는 경우 전원을 껐다 켜 후 원래 구성을 부트할 수 있습니다. 활성 구성이 저하된 구성인 경우 `ldm list-sponfig` 출력에서 `[degraded]`로 표시됩니다.

활성 구성이 저하된 구성인 동안에는 자동 저장 기능이 사용 안함으로 설정됩니다. 저하된 구성이 활성 상태인 동안 SP에 새 구성을 저장하면 새 구성은 저하되지 않은 일반 구성입니다.

주 - 이전에 누락되었지만 전원을 껐다 켜 후 다시 나타난 리소스는 일반 구성 콘텐츠에 영향을 끼치지 않습니다. 단, 나중에 복구 모드를 트리거한 구성을 선택할 경우 SP는 모든 하드웨어를 사용할 수 있는 저하되지 않은 원래 구성을 부트합니다.

복구 모드 제어

`ldmd/recovery_mode` SMF 등록 정보는 복구 모드 동작을 제어합니다. 복구 모드는 기본적으로 사용으로 설정됩니다.

`ldmd/recovery_mode` 등록 정보가 제공되지 않거나 `auto`로 설정된 경우 복구 모드가 사용으로 설정됩니다.

`ldmd/recovery_mode` 등록 정보가 `never`로 설정된 경우 Logical Domains Manager는 작업을 수행하지 않고 복구 모드를 종료하며 물리적 도메인이 `factory-default` 구성을 실행합니다.

주 - 시스템 펌웨어가 사용으로 설정되지 않은 복구 모드를 요청할 경우 다음 명령을 실행하여 요청이 생성된 후 복구 모드를 사용으로 설정하십시오.

```
primary# svccfg -s ldmd setprop ldmd/recovery_mode = astring: auto
primary# svcadm refresh ldmd
primary# svcadm restart ldmd
```

시스템이 변경되지 않은 경우(`factory-default` 구성인 경우)에만 이 시나리오에서 즉시 복구 모드가 시작됩니다.

복구 모드를 사용으로 설정하는 것 외에도, 복구 동안 루트 도메인의 부트 시간 초과 값을 지정할 수 있습니다. 기본적으로 `ldmd/recovery_mode_boot_timeout` 등록 정보 값은 30분입니다. 유효한 값은 5분부터 시작합니다.

도메인을 저하됨으로 표시

FMA 리소스 차단으로 도메인이 시작 리소스 부족 상태로 유지되는 경우 도메인이 저하됨으로 표시됩니다. 그러면 도메인은 바인드 상태로 유지되어 도메인에 지정된 나머지 리소스가 다른 도메인에 재할당되지 않도록 설정됩니다.

I/O 리소스를 비워짐으로 표시

복구 모드를 통해 누락된 것으로 감지된 I/O 리소스는 `ldm list` 출력에서 별표(*)가 붙어 비워짐으로 표시됩니다.

◆◆◆ 18 장

기타 관리 작업 수행

이 장에서는 Oracle VM Server for SPARC 소프트웨어 사용에 대한 정보 및 이전 장에서 설명되지 않은 작업을 제공합니다.

이 장에서는 다음 주제를 다룹니다.

- “CLI에서 이름 입력” [351]
- “/etc/system 파일에서 등록 정보 값 업데이트” [352]
- “부하가 높은 도메인을 중지할 때 시간 초과가 발생할 수 있음” [353]
- “Oracle VM Server for SPARC와 함께 Oracle Solaris OS 작동” [354]
- “서비스 프로세서와 함께 Oracle VM Server for SPARC 사용” [355]
- “도메인 종속성 구성” [356]
- “CPU 및 메모리 주소를 매핑하여 오류 발생 위치 확인” [360]
- “UUID(Universally Unique Identifier) 사용” [362]
- “가상 도메인 정보 명령 및 API” [362]
- “논리적 도메인 채널 사용” [363]
- “대량의 도메인 부트” [366]
- “Oracle VM Server for SPARC 시스템의 정상 종료 및 전원 꺾다 켜기” [367]
- “Logical Domains 변수 지속성” [367]
- “인터럽트 제한 조정” [369]
- “도메인 I/O 종속성 나열” [371]
- “Logical Domains Manager 데몬 사용으로 설정” [372]
- “자동 저장 구성 데이터 저장 및 복원” [373]
- “출하 시 기본 구성과 도메인 사용 안함으로 설정” [374]

CLI에서 이름 입력

일반적으로 Logical Domains Manager 이름은 최대 256자일 수 있습니다.

다음 절에서는 Logical Domains Manager CLI의 이름 지정 제한 사항에 대해 설명합니다.

- 변수 이름

- 첫번째 문자는 문자, 숫자 또는 슬래시(/)여야 합니다.
- 후속 문자는 문자, 숫자 또는 문장 부호여야 합니다.
- 가상 디스크 백엔드, 가상 스위치 장치 이름 및 경로 파일 이름에 사용되는 파일 이름입니다.
이 이름에는 문자, 숫자 또는 문장 부호만 들어 있어야 합니다.
- 구성 이름
도메인 구성 이름(또는 SP 구성 이름)의 길이는 SP에서 제한합니다. 이 제한은 현재 약 69 자이지만 플랫폼마다 다를 수 있습니다.
너무 긴 SP 구성 이름을 지정하면 다음 오류가 발생합니다.

```
primary# ldm add-sconfig \  
test567890123456789212345678931234567894123456789512345678961234567897  
Error: Operation failed because an invalid configuration name was given
```

- 가상 장치 서비스 및 클라이언트 이름
가상 장치 이름은 OpenBoot PROM에서 사용하는 `devalias` 등록 정보를 만드는 데 사용됩니다. 하지만 OpenBoot PROM은 31자를 초과하는 `devalias` 이름을 지원하지 않습니다.
31자를 초과하는 가상 장치 이름을 지정하는 경우 명령이 성공하지만 해당 `devalias` 등록 정보는 생성되지 않습니다. 이 명령은 다음 경고도 실행합니다.

```
primary# ldm add-vds primary-vds012345678901234567890 primary  
Warning: Device name primary-vds012345678901234567890 is too long to create devalias
```

- 하드웨어 경로 이름
이 이름은 물리적 리소스에 대한 경로입니다. 이 이름은 `ldm add-vsan` 명령의 `iport`와 `ldm list-rsrc-group` 명령의 리소스 그룹을 지정하는 데 사용됩니다.
- 기타 모든 이름
 - 첫번째 문자는 문자 또는 숫자여야 합니다.
 - 후속 문자는 문자, 숫자 또는 `-_+#.!:~()` 문자여야 합니다.

/etc/system 파일에서 등록 정보 값 업데이트

/etc/system 파일을 수동으로 변경하지 마십시오. 이 파일은 /etc/system.d 디렉토리의 파일에 조정 등록 정보 값을 지정할 때 재부트 시 자동으로 생성됩니다.

▼ 조정 등록 정보 값을 추가하거나 수정하는 방법

1. 기존 /etc/system 파일과 /etc/system.d 파일에서 조정 등록 정보 이름을 검색합니다.

예를 들어, `vds:vd_volume_force_slice` 등록 정보에 값을 지정하려면 등록 정보가 이미 설정되었는지 여부를 확인합니다.

```
# grep 'vds:vd_volume_force_slice' /etc/system /etc/system.d/*
```

2. 등록 정보 값을 업데이트합니다.

- `/etc/system.d` 파일 중 하나에서 등록 정보를 발견하면 기존 파일에서 등록 정보 값을 업데이트합니다.
- `/etc/system` 파일에서 등록 정보를 발견하거나 등록 정보를 찾을 수 없으면 다음 예제와 같은 이름을 사용하여 `/etc/system.d` 디렉토리에 파일을 만듭니다.

```
/etc/system.d/com.company-name:ldoms-config
```

부하가 높은 도메인을 중지할 때 시간 초과가 발생할 수 있음

도메인 종료 작업이 완료되기 전에 `ldm stop-domain` 명령 시간 초과가 발생할 수 있습니다. 이 문제가 발생하면 Logical Domains Manager가 다음과 유사한 오류를 반환합니다.

```
LDom ldg8 stop notification failed
```

하지만 이 경우에도 도메인 종료 요청은 계속 처리될 수 있습니다. 도메인 상태를 확인하려면 `ldm list-domain` 명령을 사용하십시오. 예를 들어, 다음과 같습니다.

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU MEMORY  UTIL  UPTIME
ldg8          active s---- 5000   22   3328M  0.3% 1d 14h 31m
```

위 목록에서는 도메인이 활성 상태임을 보여 주지만 `s` 플래그를 통해 도메인 중지가 진행 중임을 알 수 있습니다. 이는 일시적인 상태입니다.

다음 예에서는 도메인이 이제 중지되었음을 보여줍니다.

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU MEMORY  UTIL  UPTIME
ldg8          bound  ----- 5000   22   3328M
```

`ldm stop` 명령은 `shutdown` 명령을 사용하여 도메인을 중지합니다. 종료 시퀀스의 실행은 대개 `ldm stop -q` 명령을 실행하여 수행할 수 있는 빠른 중지보다 훨씬 오래 걸립니다. [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

긴 종료 시퀀스로 인해 다음과 같은 시간 초과 메시지가 생성될 수 있습니다.

```
domain-name stop timed out. The domain might still be in the process of shutting down.
Either let it continue, or specify -f to force it to stop.
```

이 종료 시퀀스가 실행되는 동안 도메인에 대해 `s` 플래그도 표시됩니다.

Oracle VM Server for SPARC와 함께 Oracle Solaris OS 작동

이 절에서는 Logical Domains Manager가 만든 구성이 인스턴스화된 후 Oracle Solaris OS를 사용할 때 변경되는 동작에 대해 설명합니다.

Oracle Solaris OS가 시작된 후 OpenBoot 펌웨어를 사용할 수 없음

Oracle Solaris OS가 시작된 후에는 OpenBoot 펌웨어를 사용할 수 없습니다. 메모리에서 제거되었기 때문입니다.

Oracle Solaris OS에서 `ok` 프롬프트에 도달하려면 Oracle Solaris OS `halt` 명령을 사용하여 도메인을 정지해야 합니다.

서버의 전원을 껐다 켜

Oracle VM Server for SPARC 소프트웨어를 실행하는 시스템에서 유지 관리를 수행할 때마다 서버의 전원을 껐다 켜야 하므로 먼저 SP에 현재 논리적 도메인 구성을 저장해야 합니다.

SP에 현재 도메인 구성을 저장하려면 다음 명령을 사용하십시오.

```
# ldm add-config config-name
```

Oracle Solaris OS 중단 결과

다음과 같이 Oracle Solaris OS 중단을 시작할 수 있습니다.

1. 입력 장치가 `keyboard`로 설정된 경우 L1-A 키 시퀀스를 누릅니다.
2. 가상 콘솔이 `telnet` 프롬프트에 있을 때 `send break` 명령을 입력합니다.

해당 중단을 시작하면 Oracle Solaris OS가 다음 프롬프트를 실행합니다.

```
c)ontinue, s)ync, r)eset, h)alt?
```

이러한 유형의 중단이 발생한 후 시스템에서 수행할 작업을 나타내는 문자를 입력합니다.

컨트롤 도메인 재부트 결과

reboot 및 shutdown -i 5 명령을 사용하여 컨트롤(primary) 도메인을 재부트할 수 있습니다.

- reboot:
 - 다른 도메인이 구성되지 않음. 정상 종료 없이 컨트롤 도메인을 재부트하고 전원을 끄지 않습니다.
 - 다른 도메인이 구성됨. 정상 종료 없이 컨트롤 도메인을 재부트하고 전원을 끄지 않습니다.
- shutdown -i 5:
 - 다른 도메인이 구성되지 않음. 정상 종료 후에 호스트 전원을 끄고 SP에서 다시 전원이 켜질 때까지 꺼진 상태로 유지합니다.
 - 다른 도메인이 구성됨. 정상 종료와 함께 재부트하고 전원을 끄지 않습니다.

일부 경우에 Oracle VM Server for SPARC 시스템의 명령줄에서 uadmin 1 0 명령을 실행하면 이후 재설정 후에도 시스템이 ok 프롬프트로 돌아가지 않습니다. 이러한 잘못된 동작은 auto-reboot? 변수가 true로 설정된 경우에만 표시됩니다. auto-reboot?가 false로 설정된 경우에는 예상된 동작이 발생합니다. 이 문제를 해결하려면 uadmin 2 0 명령을 사용하거나 항상 auto-reboot?를 false로 설정하여 실행합니다.

루트 도메인 역할이 있는 도메인의 재부트 결과에 대한 자세한 내용은 “PCIe 끝점이 구성된 상태로 루트 도메인 재부트” [140]를 참조하십시오.

서비스 프로세서와 함께 Oracle VM Server for SPARC 사용

이 절에서는 Logical Domains Manager와 함께 ILOM(Integrated Lights Out Manager) SP (서비스 프로세서)를 사용하는 것과 관련된 정보에 대해 설명합니다. ILOM 소프트웨어 사용에 대한 자세한 내용은 사용 중인 특정 플랫폼에 대한 문서(<http://www.oracle.com/technetwork/documentation/sparc-tseries-servers-252697.html>)를 참조하십시오.

기존 ILOM 명령에 추가 config 옵션을 사용할 수 있습니다.

```
-> set /HOST/bootmode config=config-name
```

이 옵션을 통해 다음 전원 켜기에 대한 구성을 factory-default 출하시 구성을 비롯한 다른 구성으로 설정할 수 있습니다.

호스트의 전원을 켜거나 끌 때 명령을 호출할 수 있습니다. 그러면 다음 번 호스트 재설정 또는 전원 켜기에 적용됩니다.

논리적 도메인 구성을 재설정하려면 옵션을 factory-default로 설정합니다.

```
-> set /HOST/bootmode config=factory-default
```

ldm add-config 명령을 사용하여 Logical Domains Manager로 만들어진 후 SP(서비스 프로세서)에 저장된 다른 구성을 선택할 수도 있습니다. Logical Domains Manager ldm add-config 명령에서 지정한 이름을 사용하여 ILOM bootmode 명령을 통해 해당 구성을 선택할 수 있습니다. 예를 들어, 이름이 ldm-config1인 구성을 저장했다고 가정합니다.

```
-> set /HOST/bootmode config=ldm-config1
```

이제 시스템의 전원을 껐다 켜서 새 구성을 로드해야 합니다.

ldm add-config 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

도메인 종속성 구성

Logical Domains Manager를 사용하여 도메인 간의 종속성 관계를 설정할 수 있습니다. 종속된 하나 이상의 도메인을 가진 도메인을 마스터 도메인이라고 합니다. 다른 도메인에 종속된 도메인을 슬레이브 도메인이라고 합니다.

master 등록 정보를 설정하여 슬레이브 도메인마다 최대 4개의 마스터 도메인을 지정할 수 있습니다. 예를 들어, pine 슬레이브 도메인은 다음과 같이 쉼표로 구분된 목록에서 4개의 마스터 도메인을 지정합니다.

```
# ldm add-domain master=alpha,beta,gamma,delta pine
```

alpha, beta, gamma, delta 마스터 도메인은 모두 실패 정책으로 stop을 지정합니다.

각 마스터 도메인은 마스터 도메인 실패 시 슬레이브 도메인에 발생하는 동작을 지정할 수 있습니다. 예를 들어, 마스터 도메인이 실패할 경우 슬레이브 도메인에 패닉이 발생하도록 할 수 있습니다. 슬레이브 도메인에 여러 개의 마스터 도메인이 있을 경우 각 마스터 도메인은 동일한 실패 정책을 가져야 합니다. 따라서 실패하는 첫번째 마스터 도메인에 따라 모든 슬레이브 도메인에 대해 정의된 실패 정책이 트리거됩니다.

마스터 도메인의 실패 정책을 제어하려면 failure-policy 등록 정보를 다음 값 중 하나로 설정하십시오.

- ignore - 모든 슬레이브 도메인을 무시합니다.
- panic - 모든 슬레이브 도메인에 패닉이 발생합니다. ldm panic 명령을 실행하는 것과 비슷합니다.
- reset - 모든 슬레이브 도메인을 즉시 중지한 후 다시 시작합니다. ldm stop -f 명령 후 ldm start 명령을 실행하는 것과 비슷합니다.
- stop - 모든 슬레이브 도메인을 중지합니다. ldm stop -f 명령을 실행하는 것과 비슷합니다.

이 예에서 마스터 도메인은 다음과 같이 실패 정책을 지정합니다.

```
primary# ldm set-domain failure-policy=ignore apple
primary# ldm set-domain failure-policy=panic lemon
```

```
primary# ldm set-domain failure-policy=reset orange
primary# ldm set-domain failure-policy=stop peach
primary# ldm set-domain failure-policy=stop alpha
primary# ldm set-domain failure-policy=stop beta
primary# ldm set-domain failure-policy=stop gamma
primary# ldm set-domain failure-policy=stop delta
```

이 방식을 사용하여 도메인 간의 명시적 종속성을 만들 수 있습니다. 예를 들어, 게스트 도메인은 암시적으로 서비스 도메인에 종속되어 가상 장치를 제공합니다. 게스트 도메인이 종속된 서비스 도메인이 작동 및 실행 중이 아닐 경우 게스트 도메인의 I/O가 차단됩니다. 게스트 도메인을 서비스 도메인의 슬레이브로 정의하여 서비스 도메인의 작동이 중지될 때 게스트 도메인에 발생하는 동작을 지정할 수 있습니다. 종속성이 설정되지 않은 경우 게스트 도메인은 서비스 도메인이 다시 작동할 때까지 기다립니다.

주 - Logical Domains Manager에서는 종속성 주기를 만드는 도메인 관계를 만들 수 없습니다. 자세한 내용은 “종속성 주기” [358]를 참조하십시오.

도메인 종속성 XML 예는 [Oracle VM Server for SPARC 3.4 개발자 설명서](#)의 “도메인 정보 (ldom_info) 리소스”를 참조하십시오.

도메인 종속성 예

다음 예에서는 도메인 종속성을 구성하는 방법을 보여줍니다.

예 77 도메인 종속성을 사용하여 실패 정책 구성

첫번째 명령은 `twizzle`이라는 마스터 도메인을 만듭니다. 이 명령은 `failure-policy=reset`을 사용하여 `twizzle` 도메인이 실패할 경우 슬레이브 도메인이 재설정되도록 지정합니다. 두번째 명령은 `primary`라는 마스터 도메인을 수정합니다. 이 명령은 `failure-policy=reset`을 사용하여 `primary` 도메인이 실패할 경우 슬레이브 도메인이 재설정되도록 지정합니다. 세번째 명령은 2개의 마스터 도메인(`twizzle` 및 `primary`)에 종속된 `chocktaw`라는 슬레이브 도메인을 만듭니다. 슬레이브 도메인은 `master=twizzle,primary`를 사용하여 마스터 도메인을 지정합니다. `twizzle` 또는 `primary` 도메인이 실패할 경우 `chocktaw` 도메인이 재설정됩니다.

```
primary# ldm add-domain failure-policy=reset twizzle
primary# ldm set-domain failure-policy=reset primary
primary# ldm add-domain master=twizzle,primary chocktaw
```

예 78 도메인을 수정하여 마스터 도메인 지정

이 예에서는 `ldm set-domain` 명령을 통해 `orange` 도메인을 수정하여 `primary`를 마스터 도메인으로 지정하는 방법을 보여줍니다. 두번째 명령은 `ldm set-domain` 명령을 사용하여 `orange` 및 `primary`를 `tangerine` 도메인의 마스터 도메인으로 지정합니다. 세번째 명령은 이러한 모든 도메인에 대한 정보를 나열합니다.

```

primary# ldm set-domain master=primary orange
primary# ldm set-domain master=orange,primary tangerine
primary# ldm list -o domain
NAME          STATE      FLAGS    UTIL
primary       active    -n-cv-   0.2%

SOFTSTATE
Solaris running

HOSTID
0x83d8b31c

CONTROL
failure-policy=ignore

DEPENDENCY
master=

-----
NAME          STATE      FLAGS    UTIL
orange        bound     -
HOSTID
0x84fb28ef

CONTROL
failure-policy=ignore

DEPENDENCY
master=primary

-----
NAME          STATE      FLAGS    UTIL
tangerine     bound     -
HOSTID
0x84f948e9

CONTROL
failure-policy=ignore

DEPENDENCY
master=orange,primary

```

예 79 구문 분석이 가능한 도메인 목록 표시

다음에서는 구문 분석이 가능한 출력이 포함된 목록 예를 보여줍니다.

```
primary# ldm list -o domain -p
```

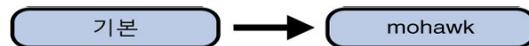
종속성 주기

Logical Domains Manager에서는 종속성 주기를 만드는 도메인 관계를 만들 수 없습니다. 종속성 주기는 슬레이브 도메인이 자체적으로 종속되거나 마스터 도메인이 슬레이브 도메인 중 하나에 종속되도록 하는 2개 이상의 도메인 간 관계입니다.

Logical Domains Manager는 종속성을 추가하기 전에 종속성 주기가 존재하는지 여부를 확인합니다. Logical Domains Manager는 슬레이브 도메인부터 시작하여 경로 끝에 도달할 때까지 마스터 배열에 따라 지정된 모든 경로를 검색합니다. 해당 경로에서 발견된 종속성 주기는 오류로 보고됩니다.

다음 예에서는 종속성 주기가 만들어질 수 있는 방법을 보여줍니다. 첫번째 명령은 마스터 도메인을 primary로 지정하는 mohawk라는 슬레이브 도메인을 만듭니다. 따라서 mohawk는 다음 다이어그램과 같이 종속성 체인에서 primary에 종속됩니다.

그림 26 단일 도메인 종속성



두번째 명령은 마스터 도메인을 counter로 지정하는 primary라는 슬레이브 도메인을 만듭니다. 따라서 mohawk는 다음 다이어그램과 같이 종속성 체인에서 counter에 종속되는 primary에 종속됩니다.

그림 27 다중 도메인 종속성



세번째 명령은 다음 다이어그램과 같이 종속성 주기가 생성되도록 counter 도메인과 mohawk 도메인 간에 종속성을 만들려고 시도합니다.

그림 28 도메인 종속성 주기



ldm set-domain 명령은 다음 오류 메시지와 함께 실패합니다.

```

# ldm add-domain master=primary mohawk
# ldm set-domain master=counter primary
# ldm set-domain master=mohawk counter
Dependency cycle detected: LDom "counter" indicates "primary" as its master
  
```

CPU 및 메모리 주소를 매핑하여 오류 발생 위치 확인

이 절에서는 Oracle Solaris FMA(Fault Management Architecture)가 보고하는 정보와 결합이 있는 것으로 표시된 논리적 도메인 리소스를 상관시키는 방법에 대해 설명합니다.

FMA는 물리적 CPU 번호와 관련하여 CPU 오류를 보고하고 물리적 메모리 주소와 관련하여 메모리 오류를 보고합니다.

논리적 도메인에서 오류가 발생한 위치와 도메인 내 해당 가상 CPU 번호 또는 실제 메모리 주소를 확인하려면 매핑을 수행해야 합니다.

CPU 매핑

지정된 물리적 CPU 번호에 해당하는 도메인 및 도메인 내 가상 CPU 번호를 찾을 수 있습니다.

먼저 다음 명령을 사용하여 모든 도메인에 대해 구문 분석이 가능한 긴 목록을 생성하십시오.

```
primary# ldm list -l -p
```

목록의 vcpu 섹션에서 pid 필드가 물리적 CPU 번호인 항목을 찾습니다.

- 해당 항목이 발견되면 항목이 나열된 도메인에 CPU가 있으며 항목의 vid 필드에 도메인 내 가상 CPU 번호가 지정된 것입니다.
- 해당 항목이 발견되지 않으면 CPU가 도메인에 없는 것입니다.

메모리 매핑

지정된 PA(물리적 메모리 주소)에 해당하는 도메인 및 도메인 내 실제 메모리 주소를 찾을 수 있습니다.

먼저 모든 도메인에 대해 구문 분석이 가능한 긴 목록을 생성하십시오.

```
primary# ldm list -l -p
```

목록의 MEMORY 섹션에서 PA가 포함 범위 pa 에서 $(pa + size - 1)$ 사이에 속한 행, 즉 $pa \leq PA \leq (pa + size - 1)$ 인 행을 찾습니다. pa 및 $size$ 는 행의 해당 필드에 있는 값을 나타냅니다.

- 항목이 발견되면 항목이 나열된 도메인에 PA가 있으며 $ra + (PA - pa)$ 로 도메인 내 해당 실제 주소가 지정된 것입니다.

- 해당 항목이 발견되지 않으면 PA가 도메인에 없는 것입니다.

CPU 및 메모리 매핑 예

예 80 도메인의 구성 결정

다음 명령은 논리적 도메인 구성의 구문 분석이 가능한 긴 목록을 생성합니다.

```
primary# ldm list -l -p
VERSION 1.6
DOMAIN|name=primary|state=active|flags=normal,control,vio-service|
cons=SP|ncpu=4|mem=1073741824|util=0.6|uptime=64801|
softstate=Solaris running
VCPU
|vid=0|pid=0|util=0.9|strand=100
|vid=1|pid=1|util=0.5|strand=100
|vid=2|pid=2|util=0.6|strand=100
|vid=3|pid=3|util=0.6|strand=100
MEMORY
|ra=0x80000000|pa=0x80000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
|dev=pci@7c0|alias=bus_b
...
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|
ncpu=2|mem=805306368|util=29|uptime=903|
softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x80000000|pa=0x48000000|size=805306368
...
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|
ncpu=3|mem=1073741824|util=35|uptime=775|
softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x80000000|pa=0x78000000|size=1073741824
...
```

예 81 물리적 CPU 번호에 해당하는 가상 CPU 확인

논리적 도메인 구성은 예 80. “도메인의 구성 결정”에 나와 있습니다. 이 예에서는 물리적 CPU 번호 5에 해당하는 도메인 및 가상 CPU와 물리적 주소 0x7e816000에 해당하는 도메인 및 실제 주소를 확인하는 방법에 대해 설명합니다.

목록의 vCPU 항목에서 pid 필드가 5인 항목을 통해 논리적 도메인 ldg1에서 다음 항목을 찾을 수 있습니다.

```
|vid=1|pid=5|util=29|strand=100
```

따라서 물리적 CPU 번호 5는 도메인 ldg1에 있으며 도메인 내 가상 CPU 번호는 1임을 알 수 있습니다.

목록의 MEMORY 항목을 통해 도메인 ldg2에서 다음 항목을 찾을 수 있습니다.

```
ra=0x8000000|pa=0x78000000|size=1073741824
```

여기서는 $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$, 즉 $pa \leq PA \leq (pa + size - 1)$ 입니다. 따라서 PA는 도메인 ldg2에 있으며 해당 실제 주소는 $0x8000000 + (0x7e816000 - 0x78000000) = 0xe816000$ 임을 알 수 있습니다.

UUID(Universally Unique Identifier) 사용

각 도메인에는 UUID(Universally Unique Identifier)가 지정됩니다. UUID는 도메인이 만들어 질 때 지정됩니다. 레거시 도메인의 경우 UUID는 ldm 데몬이 초기화될 때 지정됩니다.

주 - ldm migrate-domain -f 명령을 사용하여 이전 버전의 Logical Domains Manager가 실행되는 대상 시스템으로 도메인을 마이그레이션할 경우 UUID가 손실됩니다. 이전 버전의 Logical Domains Manager가 실행되는 소스 시스템에서 도메인을 마이그레이션할 때 마이그레이션의 일부로 도메인에 새 UUID가 지정됩니다. 그렇지 않은 경우 UUID가 마이그레이션됩니다.

ldm list -l, ldm list-bindings 또는 ldm list -o domain 명령을 실행하여 도메인의 UUID를 확인할 수 있습니다. 다음 예에서는 ldg1 도메인의 UUID를 보여줍니다.

```
primary# ldm create ldg1
primary# ldm ls -l ldg1
NAME          STATE          FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1          inactive      -----
UUID
6c908858-12ef-e520-9eb3-f1cd3dbc3a59

primary# ldm ls -l -p ldg1
VERSION 1.6
DOMAIN|name=ldg1|state=inactive|flags=|cons=|ncpu=|mem=|util=|uptime=
UUID|uuid=6c908858-12ef-e520-9eb3-f1cd3dbc3a59
```

가상 도메인 정보 명령 및 API

virtinfo 명령을 사용하여 실행 중인 가상 도메인에 대한 정보를 수집할 수 있습니다. 가상 도메인 정보 API를 사용하여 가상 도메인 관련 정보를 수집할 프로그램을 만들 수도 있습니다.

다음 목록에서는 명령 또는 API를 사용하여 가상 도메인에 대해 수집할 수 있는 몇 가지 정보를 보여줍니다.

- 도메인 유형(구현, 컨트롤, 게스트, I/O, 서비스, 루트)
- Virtual Domain Manager로 확인된 도메인 이름
- 도메인의 UUID(Universally Unique Identifier)
- 도메인의 컨트롤 도메인에 대한 네트워크 노드 이름
- 도메인이 실행 중인 새시 일련 번호

`virtinfo` 명령에 대한 자세한 내용은 `virtinfo(1M)` 매뉴얼 페이지를 참조하십시오. API에 대한 자세한 내용은 `libv12n(3LIB)` 및 `v12n(3EXT)` 매뉴얼 페이지를 참조하십시오.

논리적 도메인 채널 사용

Oracle VM Server for SPARC에서는 논리적 도메인 채널(LDC)을 사용하여 콘솔, 가상 I/O 및 컨트롤 트래픽과 같은 모든 통신을 구현합니다. LDC는 두 끝점 사이에 통신을 사용으로 설정하는 데 사용되는 방법입니다. 일반적으로 각 끝점은 서로 다른 도메인에 있지만 끝점은 동일한 도메인에서 루프백 통신을 사용으로 설정할 수 있습니다.

이 소프트웨어 및 시스템 펌웨어는 컨트롤 도메인과 게스트 도메인에 사용할 수 있는 LDC 끝점의 대형 풀을 제공합니다. 이 LDC 끝점 풀은 SPARC T4 서버, SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버, SPARC S7 시리즈 서버 및 Fujitsu M10 서버용으로만 제공됩니다.

LDC 끝점 풀 지원에 필요한 시스템 펌웨어는 8.5.2(SPARC T4 서버의 경우), 9.2.1(SPARC T5 서버, SPARC M5 서버 및 SPARC M6 서버의 경우), 9.4.3(SPARC T7 시리즈 서버 및 SPARC M7 시리즈 서버의 경우), 릴리스된 모든 버전(SPARC S7 시리즈 서버의 경우), XCP2240(Fujitsu M10 서버의 경우)입니다.

지원되는 플랫폼이나 UltraSPARC T2, UltraSPARC T2 Plus 또는 SPARC T3 플랫폼에서 이전 버전의 시스템 펌웨어를 실행하는 경우에도 다음과 같이 LDC 끝점 제한이 적용됩니다.

- **UltraSPARC T2** 서버 – LDC 끝점 512개
- **UltraSPARC T2 Plus** 서버 – LDC 끝점 768개
- **SPARC T3** 서버 – LDC 끝점 768개
- **SPARC T4** 서버 – LDC 끝점 768개
- **SPARC T5** 서버 – LDC 끝점 768개
- **SPARC T7** 시리즈 서버 – LDC 끝점 768개
- **SPARC M5** 서버 – LDC 끝점 768개
- **SPARC M6** 서버 – LDC 끝점 768개

- SPARC M7 시리즈 서버 – LDC 끝점 768개
- Fujitsu M10 서버 – LDC 끝점 768개

또한 가상 I/O 데이터 통신 및 다른 도메인의 Logical Domains Manager 컨트롤 모두에 사용되는 LDC 끝점 수는 잠재적으로 매우 크기 때문에 이러한 제한이 컨트롤 도메인에서 문제가 될 수 있습니다.

서비스를 추가하거나 도메인을 바인드하려고 시도할 때 LDC 끝점 수가 단일 도메인에서 제한을 초과할 경우, 다음과 비슷한 오류 메시지와 함께 작업이 실패합니다.

13 additional LDCs are required on guest primary to meet this request,
but only 9 LDCs are available

다음 지침을 통해 LDC 끝점 사용을 위해 제대로 계획하고 컨트롤 도메인의 LDC 기능 오버플로가 발생 가능한 이유를 설명할 수 있습니다.

- 컨트롤 도메인은 구성된 기타 도메인 수에 관계없이 하이퍼바이저, FMA(Fault Management Architecture) 및 SP(시스템 프로세서)에서 다양한 통신 목적에 따라 약 15개의 LDC 끝점을 사용합니다. 컨트롤 도메인이 사용하는 LDC 끝점의 수는 플랫폼 및 사용되는 소프트웨어의 버전에 따라 달라집니다.
- Logical Domains Manager는 컨트롤 트래픽을 위해 자신을 포함하여 모든 도메인에 대해 LDC 끝점을 컨트롤 도메인에 할당합니다.
- 컨트롤 도메인의 각 가상 I/O 서비스는 해당 서비스의 모든 연결된 클라이언트에 대해 1개의 LDC 끝점을 사용합니다. 각 도메인에는 가상 네트워크, 가상 디스크 및 가상 콘솔이 최소한 하나씩 있어야 합니다.

다음 공식은 이러한 지침에 따라 컨트롤 도메인에서 필요한 LDC 끝점의 수를 결정합니다.

$$15 + \text{number-of-domains} + (\text{number-of-domains} \times \text{number-of-virtual-services}) = \text{total-LDC-endpoints}$$

*number-of-domains*는 컨트롤 도메인을 포함한 총 도메인 수이고, *number-of-virtual-services*는 이 도메인에서 서비스하는 총 가상 I/O 장치 수입니다.

다음 예는 1개의 컨트롤 도메인과 8개의 추가 도메인이 있을 때 공식을 사용하여 LDC 끝점 수를 결정하는 방법을 보여 줍니다.

$$15 + 9 + (8 \times 3) = 48 \text{개 LDC 끝점}$$

다음 예에는 45개의 게스트 도메인이 있으며, 각 도메인에는 5개의 가상 디스크, 2개의 가상 네트워크 및 1개의 가상 콘솔이 포함되어 있습니다. 계산하면 다음 결과를 얻게 됩니다.

$$15 + 46 + 45 \times 8 = 421 \text{개 LDC 끝점}$$

플랫폼에서 지원되는 LDC 끝점 수에 따라 Logical Domains Manager에서 구성을 수락하거나 거부할 수 있습니다.

컨트롤 도메인에 LDC 끝점이 부족할 경우 서비스 도메인 또는 I/O 도메인을 만들어 게스트 도메인에 가상 I/O 서비스를 제공할 수 있습니다. 이 작업을 통해 LDC 끝점을 컨트롤 도메인 대신 I/O 도메인 및 서비스 도메인에서 만들 수 있습니다.

게스트 도메인에도 LDC 끝점이 부족할 수 있습니다. 이 상황은 `inter-vnet-link` 등록 정보가 `on`으로 설정되어 추가 LDC 끝점이 게스트 도메인에 지정되고 서로 직접 연결되기 때문에 발생할 수 있습니다.

다음 공식은 `inter-vnet-link=off`일 때 게스트 도메인에서 필요한 LDC 끝점 수를 결정합니다.

$$2 + \text{number-of-vnets} + \text{number-of-vdisks} = \text{total-LDC-endpoints}$$

2는 가상 콘솔과 컨트롤 트래픽을 나타내며, `number-of-vnets`는 게스트 도메인에 지정된 총 가상 네트워크 장치 수이고, `number-of-vdisks`는 게스트 도메인에 지정된 총 가상 디스크 수입니다.

다음 예는 `inter-vnet-link=off`이고 2개의 가상 디스크와 2개의 가상 네트워크가 있을 때 공식을 사용하여 게스트 도메인당 LDC 끝점 수를 결정하는 방법을 보여 줍니다.

$$2 + 2 + 2 = 6 \text{개 LDC 끝점}$$

다음 공식은 `inter-vnet-link=on`일 때 게스트 도메인에서 필요한 LDC 끝점 수를 결정합니다.

$$2 + [[(\text{number-of-vnets-from-vswX} \times \text{number-of-vnets-in-vswX})] \dots] + \text{number-of-vdisks} = \text{total-LDC-endpoints}$$

2는 가상 콘솔과 컨트롤 트래픽을 나타내며, `number-of-vnets-from-vswX`는 `vswX` 가상 스위치에서 게스트 도메인에 지정된 총 가상 네트워크 장치 수이고, `number-of-vnets-in-vswX`는 `vswX` 가상 스위치의 총 가상 네트워크 장치 수이며, `number-of-virtual-disks`는 게스트 도메인에 지정된 총 가상 디스크 수입니다.

다음 예는 `inter-vnet-link=on`이고 2개의 가상 디스크와 2개의 가상 스위치가 있을 때 공식을 사용하여 게스트 도메인당 LDC 끝점 수를 결정하는 방법을 보여 줍니다. 첫번째 가상 스위치에는 8개의 가상 네트워크가 있고 이 중 4개를 도메인에 지정합니다. 두번째 가상 스위치는 8개의 모든 가상 네트워크를 도메인에 지정합니다.

$$2 + (4 \times 8) + (8 \times 8) + 2 = 100 \text{개 LDC 끝점}$$

Oracle VM Server for SPARC 3.4 소프트웨어를 사용하여 만든 가상 네트워크 장치는 `inter-vnet-link=auto`를 기본적으로 사용합니다. 이 기능은 숫자가 임계값을 초과하는 경우 자동으로 `inter-vnet-links`를 해제합니다. 하지만 `inter-vnet-link=on`으로 만든 가상 네트워크 장치는 LDC 채널 수를 줄이기 위해 `inter-vnet-link=off`를 변경하도록 명시적으로 수정해야 합니다. 자세한 내용은 “[Vnet 간 LDC 채널](#)” [225]을 참조하십시오.

`inter-vnet-link=off`를 설정하여 가상 네트워크 장치가 있는 도메인에서 LDC 끝점 수를 줄일 수 있습니다. 그러나 서비스 도메인에는 여전히 각 가상 네트워크 장치에 LDC 연결이 필요하므로 가상 스위치가 있는 서비스 도메인에는 `off` 등록 정보 값이 영향을 미치지 않습니다. 이 등

록 정보가 off로 설정된 경우 LDC 채널이 vnet 간 통신에 사용되지 않습니다. 대신 가상 네트워크 장치 및 가상 스위치 장치 간 통신에 대해서만 LDC 채널이 지정됩니다. [1dm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

주 - Vnet 간 링크 지정을 사용 안함으로 설정해도 LDC 끝점 수가 줄어들지만 게스트-게스트 네트워크 성능에 부정적인 영향을 줄 수 있습니다. 이러한 성능 저하는 모든 게스트 간 통신 트래픽이 한 게스트 도메인에서 다른 게스트 도메인으로 직접 이동하지 않고 가상 스위치를 거치기 때문에 발생합니다.

대량의 도메인 부트

서버에 따라 다음 개수의 도메인을 부트할 수 있습니다.

- 물리적 분할 영역당 Fujitsu M10 서버의 경우 최대 256개
- 물리적 도메인당 SPARC M7 시리즈 서버의 경우 최대 128개
- 물리적 도메인당 SPARC M6 서버의 경우 최대 128개
- 물리적 도메인당 SPARC M5 서버의 경우 최대 128개
- SPARC T7 시리즈 서버의 경우 최대 128개
- SPARC T5 서버의 경우 최대 128개
- SPARC T4 서버의 경우 최대 128개
- SPARC T3 서버의 경우 최대 128개
- UltraSPARC T2 Plus 서버의 경우 최대 128개
- UltraSPARC T2 서버의 경우 최대 64개

할당되지 않은 가상 CPU가 있는 경우, 가상 I/O 요청 처리를 지원하도록 서비스 도메인에 이를 지정할 수 있습니다. 32개를 초과하는 도메인을 만들 때는 서비스 도메인에 4~8개의 가상 CPU를 할당합니다. 최대 도메인 구성에 따라 서비스 도메인에 단일 CPU만 포함된 경우 도메인을 구성 및 사용할 때 단일 CPU에 불필요한 스트레스를 주지 마십시오. 가상 스위치(vsw) 서비스는 시스템에서 사용 가능한 모든 네트워크 어댑터에 분산되어야 합니다. 예를 들어, Sun SPARC Enterprise T5240 서버에서 128개의 도메인을 부트할 경우 각각 32개의 가상 네트워크(vnet) 인스턴스를 지원하는 4개의 vsw 서비스를 만듭니다. vsw 서비스당 32개보다 많은 vnet 인스턴스를 지정하면 서비스 도메인에서 하드 중단이 발생할 수 있습니다.

최대 구성을 실행하려면 시스템에서 게스트 도메인을 지원하는 데 적합한 메모리 양이 필요합니다. 메모리 양은 플랫폼 및 OS에 따라 달라집니다. 플랫폼 설명서 [Oracle Solaris 10 8/11 Installation Guide: Planning for Installation and Upgrade](#) 및 [Installing Oracle Solaris 11.3 Systems](#)를 참조하십시오.

도메인에서 사용되는 vsw 서비스가 여러 도메인에 있는 여러 가상 네트워크에 서비스를 제공할 경우 게스트 도메인에서 메모리 및 스왑 공간 사용이 늘어납니다. 이 증가는 vsw에 연결된 모든 vnet 인스턴스 간의 피어 투 피어 연결 때문입니다. 서비스 도메인은 추가 메모리를 확보함

로써 이점을 얻을 수 있습니다. 64개를 초과하는 도메인을 실행할 때의 권장 최소값은 4GB입니다. 10개 이하의 도메인으로 시작하고, 현재 일괄 처리가 부트될 때까지 기다린 후 다음 일괄 처리를 시작하십시오. 도메인에 운영체제를 설치할 때도 동일한 조건이 적용됩니다. Vnet 간 링크를 사용 안함으로 설정하여 링크 수를 줄일 수 있습니다. [Oracle VM Server for SPARC 3.4 관리 설명서](#)의 “Vnet 간 LDC 채널”을 참조하십시오.

Oracle VM Server for SPARC 시스템의 정상 종료 및 전원 켜다 켜기

SP에 마지막으로 구성을 저장한 후 구성을 변경한 경우에는 Oracle VM Server for SPARC 시스템 전원을 끄거나 전원을 켜다 켜기 전에 보존하려는 최신 구성을 저장할지 확인하십시오.

▼ 여러 활성 도메인이 연결된 시스템 전원 끄는 방법

1. 모든 비I/O 도메인을 종료, 중지 및 바인드 해제합니다.
2. 모든 활성 I/O 도메인을 종료, 중지 및 바인드 해제합니다.
3. 도메인을 초기 상태 5로 변경합니다.

```
primary# shutdown -i 5
```

shutdown 명령을 사용하는 대신 `init 5` 명령을 사용할 수도 있습니다.

▼ 시스템 전원 켜다 켜는 방법

1. 시스템의 전원을 끕니다.
[여러 활성 도메인이 연결된 시스템 전원 끄는 방법 \[367\]](#)을 참조하십시오.
2. SP를 사용하여 시스템의 전원을 켭니다.

Logical Domains 변수 지속성

변수 업데이트는 재부트 후에도 지속되지만 변수 업데이트가 컨트롤 도메인에서 OpenBoot 펌웨어로부터 시작되지 않았거나 이후 구성을 SP에 저장하지 않은 경우, 전원을 켜다 켜기 때는 지속되지 않습니다.

다음 조건에 유의하십시오.

- 컨트롤 도메인을 재부트할 때, 바인드된 게스트 도메인이 없고 진행 중인 지연된 재구성이 없으면, SP에서 시스템 전원을 켜다 켜집니다.
- 컨트롤 도메인이 재부트될 때, 게스트 도메인이 바인드되었거나 활성 상태이면(또는 컨트롤 도메인이 지연된 재구성을 수행하는 중인 경우), SP가 시스템 전원을 켜다 켜지 않습니다.

도메인에 대한 Logical Domains 변수는 다음 방법을 사용하여 지정할 수 있습니다.

- OpenBoot 프롬프트에서
- Oracle Solaris OS `eeprom(1M)` 명령 사용
- Logical Domains Manager CLI (`ldm`) 사용
- 제한된 방식으로 SP(서비스 프로세서)에서 `bootmode` 명령 사용. 이 방법은 특정 변수에 대해 `factory-default` 구성에서만 사용할 수 있습니다.

이러한 방법을 사용하여 수행된 변수 업데이트는 도메인 재부트 시에도 항상 지속되어야 합니다. 변수 업데이트는 또한 SP에 저장된 이후의 도메인 구성에도 항상 적용됩니다.

Oracle VM Server for SPARC 3.4 소프트웨어에서는 변수 업데이트가 예상한 대로 지속되지 않는 일부 경우가 존재합니다.

- 모든 변수 업데이트 방법은 해당 도메인의 재부트 시에도 지속됩니다. 하지만 이후 논리적 도메인 구성을 SP에 저장하지 않으면 시스템을 켜다 켜 때 지속되지 않습니다.
컨트롤 도메인에서 OpenBoot 펌웨어 명령 또는 `eeprom` 명령을 사용하여 수행한 업데이트는 시스템 전원을 켜다 켜 때도 지속됩니다. 즉, 이후에 새 논리적 도메인 구성을 SP에 저장하지 않아도 업데이트가 지속됩니다. `eeprom` 명령은 SPARC T4로 시작하는 서버에서 이 동작을 지원합니다.

Logical Domains 변수 변경이 문제가 될 경우 다음 중 하나를 수행합니다.

- 시스템에 `ok` 프롬프트를 표시하고 변수를 업데이트합니다.
- Logical Domains Manager가 사용 안함으로 설정된 상태에서 변수를 업데이트합니다.

```
# svcadm disable ldmd
update variables
# svcadm enable ldmd
```

- 라이브 업그레이드를 실행할 때 다음 단계를 수행합니다.

```
# svcadm disable -t ldmd
# luactivate be3
# init 6
```

논리적 도메인에서 시간 또는 날짜를 수정할 경우(예: `ntpdate` 명령 사용), 도메인 재부트 시에 변경사항이 지속되지만 호스트의 전원을 켜다 켜 때는 지속되지 않습니다. 시간 변경사항이 지속되도록 하려면 시간 변경을 포함하는 구성을 SP에 저장하고 해당 구성으로부터 부트합니다.

이러한 문제 해결을 위해 버그 ID 15375997, 15387338, 15387606, 15415199가 제공되었습니다.

인터럽트 제한 조정

하드웨어는 한정된 수의 인터럽트를 제공하므로 Oracle Solaris는 각 장치가 사용할 수 있는 인터럽트 수를 제한합니다. 기본 제한은 표준 시스템 구성 요구에 맞아야 하지만, 특정 시스템 구성에 따라 이 값을 조정해야 할 수도 있습니다.

주 - 이러한 제한 사항은 SPARC M7 시리즈 서버와 SPARC T7 시리즈 서버에 적용되지 않습니다.

PCIe 버스에서 I/O 가상화를 사용으로 설정하면 각 I/O 도메인에 인터럽트 하드웨어 리소스가 지정됩니다. 이들 중 한정된 수의 리소스만 각 도메인에 할당되므로 인터럽트 할당 문제가 발생할 수 있습니다. 이 상황은 UltraSPARC T2, UltraSPARC T2 Plus, SPARC T3, SPARC T4, SPARC T5, SPARC M5, SPARC M6 플랫폼에만 영향을 줍니다.

Oracle Solaris 콘솔에 표시된 다음 경고는 I/O 장치 드라이버를 연결하는 동안 인터럽트 공급이 소진되었음을 의미합니다.

```
WARNING: ddi_intr_alloc: cannot fit into interrupt pool
```

특히, 시스템이 여러 논리적 도메인으로 분할된 경우와 너무 많은 I/O 장치가 게스트 도메인에 지정된 경우 제한 조정이 필요합니다. Oracle VM Server for SPARC는 인터럽트 총수를 더 작은 세트로 나누어 게스트 도메인에 지정합니다. 너무 많은 I/O 장치가 게스트 도메인에 지정된 경우 각 장치에 기본 인터럽트 제한을 제공하기에는 인터럽트 공급량이 너무 작을 수 있습니다. 따라서 게스트 도메인이 모든 드라이버를 완전히 연결하기 전에 인터럽트 공급량을 소진합니다.

일부 드라이버는 Oracle Solaris OS에서 자동으로 인터럽트를 조정할 수 있도록 선택적 콜백 루틴을 제공합니다. 이러한 드라이버에는 기본 제한이 적용되지 않습니다.

`::irmpools` 및 `::irmreqs` MDB 매크로를 사용하여 인터럽트가 어떻게 사용되는지 확인합니다. `::irmpools` 매크로는 전체 인터럽트 공급량을 풀로 나눈 값을 보여줍니다. `::irmreqs` 매크로는 각 풀에 매핑된 장치를 보여줍니다. 각 장치에 대해 `::irmreqs`는 선택적 콜백 루틴에 의해 기본 제한이 강제 적용되는지 여부, 각 드라이버가 요청한 인터럽트 수, 각 드라이버가 소유한 인터럽트 수를 보여줍니다.

이 매크로에 연결을 실패한 드라이버에 대한 정보는 표시되지 않지만 이 정보를 바탕으로 어느 정도까지 기본 제한을 조정할 수 있는지 계산할 수 있습니다. 콜백 루틴을 제공하지 않고 여러 개의 인터럽트를 사용하는 장치의 경우 기본 제한을 조정하여 더 적은 인터럽트를 사용하도록 강제할 수 있습니다. 해당 장치에서 기본 제한을 줄이면 다른 장치에서 사용할 인터럽트를 확보할 수 있습니다.

기본 제한을 조정하려면 `/etc/system` 파일에서 `ddi_msix_alloc_limit` 등록 정보를 1 - 8 범위의 값으로 설정합니다. 그런 다음 시스템을 재부트하여 변경사항을 적용합니다.

`/etc/system` 등록 정보 값을 정확히 만들거나 업데이트하는 방법은 “[/etc/system 파일에서 등록 정보 값 업데이트](#)” [352]를 참조하십시오.

성능을 최대화하려면 큰 값을 지정하여 시작했다가 시스템이 경고 없이 성공적으로 부트할 때까지 증분 값을 조금씩 줄입니다. `::irmpools` 및 `::irmreqs` 매크로를 사용하여 값 조정이 모든 연결된 드라이버에 미치는 영향을 측정합니다.

예를 들어, 게스트 도메인에서 Oracle Solaris OS를 부트하는 동안 다음 경고가 발생한다고 가정합니다.

```
WARNING: emlxs3: interrupt pool too full.
WARNING: ddi_intr_alloc: cannot fit into interrupt pool
```

`::irmpools` 및 `::irmreqs` 매크로는 다음 정보를 보여줍니다.

```
# echo "::irmpools" | mdb -k
ADDR          OWNER      TYPE      SIZE  REQUESTED  RESERVED
00000400016be970 px#0      MSI-X    36    36          36

# echo "00000400016be970::irmreqs" | mdb -k
ADDR          OWNER      TYPE      CALLBACK  NINTRS  NREQ  NAVAIL
00001000143acaa8 emlxs#0    MSI-X    No        32      8     8
00001000170199f8 emlxs#1    MSI-X    No        32      8     8
000010001400ca28 emlxs#2    MSI-X    No        32      8     8
0000100016151328 igb#3     MSI-X    No        10      3     3
0000100019549d30 igb#2     MSI-X    No        10      3     3
0000040000e0f878 igb#1     MSI-X    No        10      3     3
000010001955a5c8 igb#0     MSI-X    No        10      3     3
```

이 예제에서 기본 제한은 장치당 8개 인터럽트이며, 마지막 `emlxs3` 장치까지 시스템에 연결하기에 인터럽트가 부족합니다. 모든 `emlxs` 인스턴스가 같은 방법으로 작동한다고 가정하면, `emlxs3`은 8개 인터럽트를 요청했을 것입니다.

총 풀 크기인 36개 인터럽트에서 모든 `igb` 장치에서 사용된 12개 인터럽트를 빼면 `emlxs` 장치에 24개 인터럽트를 사용할 수 있습니다. 24개 인터럽트를 4로 나누면 장치당 6개 인터럽트가 모든 `emlxs` 장치를 같은 성능으로 연결할 수 있을 것입니다. 따라서 다음 조정이 `/etc/system` 파일에 추가됩니다.

```
set ddi_msix_alloc_limit = 6
```

`/etc/system` 등록 정보 값을 정확히 만들거나 업데이트하는 방법은 [“/etc/system 파일에서 등록 정보 값 업데이트” \[352\]](#)를 참조하십시오.

시스템이 경고 없이 성공적으로 부트하면 `::irmpools` 및 `::irmreqs` 매크로는 다음 업데이트된 정보를 보여줍니다.

```
primary# echo "::irmpools" | mdb -k
ADDR          OWNER      TYPE      SIZE  REQUESTED  RESERVED
00000400018ca868 px#0      MSI-X    36    36          36

# echo "00000400018ca868::irmreqs" | mdb -k
ADDR          OWNER      TYPE      CALLBACK  NINTRS  NREQ  NAVAIL
0000100016143218 emlxs#0    MSI-X    No        32      8     6
0000100014269920 emlxs#1    MSI-X    No        32      8     6
000010001540be30 emlxs#2    MSI-X    No        32      8     6
00001000140cbe10 emlxs#3    MSI-X    No        32      8     6
00001000141210c0 igb#3     MSI-X    No        10      3     3
0000100017549d38 igb#2     MSI-X    No        10      3     3
0000040001ceac40 igb#1     MSI-X    No        10      3     3
000010001acc3480 igb#0     MSI-X    No        10      3     3
```

도메인 I/O 종속성 나열

한 도메인의 I/O 작업에 종종 서비스 도메인이나 I/O 도메인 같은 다른 도메인이 관여하기도 합니다. 예를 들어, 서비스 도메인이 가상 장치를 내보낼 수도 있고 루트 도메인이 물리적 장치에 직접 액세스를 제공할 수도 있습니다.

서비스 도메인이나 루트 도메인의 가동이 중단되면 종속 도메인에도 서비스 중단이 발생하므로 이러한 암시적 I/O 종속성을 유념해 두십시오.

`ldm list-dependencies` 명령을 사용하여 도메인 간의 I/O 종속성을 볼 수 있습니다. 도메인 종속성을 나열하는 것 외에도, 특정 도메인의 종속 항목을 표시하도록 출력을 바꿀 수 있습니다.

다음 목록은 `ldm list-dependencies` 명령을 사용하여 볼 수 있는 I/O 종속성의 유형을 보여줍니다.

VDISK	가상 디스크 서버가 내보낸 가상 디스크 백엔드에 가상 디스크를 연결할 때 만들어진 종속성
VNET	가상 네트워크 장치를 가상 스위치에 연결할 때 만들어진 종속성
IOV	SR-IOV 가상 기능을 SR-IOV 물리적 기능과 연관시킬 때 만들어진 종속성

다음 `ldm list-dependencies` 명령은 도메인 종속성 정보를 볼 수 있는 몇 가지 방법을 보여줍니다.

- 상세한 도메인 종속성 정보를 표시하려면 `-l` 옵션을 사용합니다.

```
primary# ldm list-dependencies -l
DOMAIN          DEPENDENCY      TYPE      DEVICE
primary
svcdom
ldg0             primary         VDISK     primary-vds0/vdisk0
                VNET           primary-vsw0/vnet0
                svcdom         VDISK     svcdom-vds0/vdisk1
                VNET           svcdom-vsw0/vnet1
ldg1             primary         VDISK     primary-vds0/vdisk0
                VNET           primary-vsw0/vnet0
                IOV           /SYS/MB/NET0/IOVNET.PF0.VF0
                svcdom         VDISK     svcdom-vds0/vdisk1
                VNET           svcdom-vsw0/vnet1
                IOV           /SYS/MB/NET2/IOVNET.PF0.VF0
```

- 종속 항목을 종속성에 따라 그룹화하여 상세한 정보를 표시하려면 `-l` 및 `-r` 옵션을 모두 사용합니다.

```
primary# ldm list-dependencies -r -l
DOMAIN          DEPENDENT      TYPE      DEVICE
```

primary	ldg0	VDISK	primary-vds0/vdisk0
		VNET	primary-vsw0/vnet0
	ldg1	VDISK	primary-vds0/vdisk0
		VNET	primary-vsw0/vnet0
svcdom	ldg0	IOV	/SYS/MB/NET0/IOVNET.PF0.VF0
		VDISK	svcdom-vds0/vdisk1
	ldg1	VNET	svcdom-vsw0/vnet1
		VDISK	svcdom-vds0/vdisk1
		VNET	svcdom-vsw0/vnet1
		IOV	/SYS/MB/NET2/IOVNET.PF0.VF0

Logical Domains Manager 데몬 사용으로 설정

Logical Domains Manager 데몬 `ldmd`는 Oracle VM Server for SPARC 소프트웨어 패키지를 설치할 때 자동으로 사용으로 설정됩니다. 데몬이 사용으로 설정된 경우 논리적 도메인을 만들고 수정 및 제어할 수 있습니다.

SPARC T7 시리즈 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에서 ILOM 상호 연결 서비스는 `ldmd` 데몬과 SP(서비스 프로세서) 간의 통신을 가능하게 해줍니다. 기본적으로 `ilomconfig-interconnect` 서비스는 사용으로 설정되어 있습니다. ILOM 상호 연결 서비스가 사용으로 설정되어 있는지 확인하려면 [ILOM 상호 연결 구성을 확인하는 방법 \[40\]](#)을 참조하십시오.



주의 - `ilomconfig-interconnect` 서비스를 사용 안함으로 설정하지 마십시오. 이 서비스를 사용 안함으로 설정하면 논리적 도메인과 OS의 올바른 작동을 방해할 수 있습니다.

▼ Logical Domains Manager 데몬을 사용으로 설정하는 방법

이 절차에 따라 `ldmd` 데몬이 사용 안함으로 설정된 경우 사용으로 설정합니다.

1. `svcadm` 명령을 사용하여 **Logical Domains Manager** 데몬 `ldmd`를 사용으로 설정합니다.

```
# svcadm enable ldmd
```

`svcadm` 명령에 대한 자세한 내용은 `svcadm(1M)` 매뉴얼 페이지를 참조하십시오.

2. **Logical Domains Manager**가 실행 중인지 확인합니다.

`ldm list` 명령은 현재 시스템에 정의된 모든 도메인을 나열해야 합니다. 특히 `primary` 도메인이 나열되고 `active` 상태여야 합니다. 다음 샘플 출력은 `primary` 도메인만 시스템에 정의되어 있음을 보여줍니다.

```
# ldm list
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary      active ---c-  SP    64    3264M  0.3%  19d 9m
```

자동 저장 구성 데이터 저장 및 복원

다음 절에서는 자동 저장 구성 디렉토리 및 제약 조건 데이터베이스 파일을 저장하고 복원하는 방법을 설명합니다.

자동 저장 구성 디렉토리 저장 및 복원

컨트롤 도메인에 운영체제를 재설치하기 전에 자동 저장 구성 디렉토리를 저장하고 복원할 수 있습니다. 컨트롤 도메인에 운영체제를 재설치할 때마다 `/var/share/ldomsmanager/autosave-autosave-name` 디렉토리에 있는 도메인 자동 저장 구성 데이터를 저장하고 복원해야 합니다.

`tar` 또는 `cpio` 명령을 사용하여 전체 디렉토리 내용을 저장하고 복원할 수 있습니다.

주 - 각 자동 저장 디렉토리에는 관련 구성의 마지막 SP 구성 업데이트에 대한 시간 기록이 있습니다. 자동 저장 파일을 복원할 때 시간 기록이 동기화되지 않을 수 있습니다. 이 경우 복원된 자동 저장 구성이 이전 상태, 즉 `[newer]` 또는 `up to date`로 표시됩니다.

자동 저장 구성에 대한 자세한 내용은 [Oracle VM Server for SPARC 3.4 관리 설명서](#)의 “[도메인 구성 관리](#)”를 참조하십시오.

▼ 자동 저장 디렉토리를 저장하고 복원하는 방법

1. 자동 저장 디렉토리를 저장합니다.

```
# cd /
# tar -cvpf autosave.tar var/share/ldomsmanager/autosave-*
```

2. (선택사항) 깔끔한 복원 작업을 위해 기존 자동 저장 디렉토리를 제거합니다.

때때로 자동 저장 디렉토리에 잉여 파일이 포함될 수 있습니다. 아마 이전 구성에서 남은 파일로, SP에 다운로드된 구성을 손상시킬 수 있습니다. 이 경우, 이 예제에 표시된 대로 복원 작업 전에 자동 저장 디렉토리를 정리합니다.

```
# cd /
# rm -rf var/share/ldomsmanager/autosave-*
```

3. 자동 저장 디렉토리를 복원합니다.

다음 명령은 `/var/share/ldomsmanager` 디렉토리에 파일 및 디렉토리를 복원합니다.

```
# cd /  
# tar -xvpf autosave.tar
```

Logical Domains 제약 조건 데이터베이스 파일 저장 및 복원

컨트롤 도메인의 운영체제를 업그레이드할 때마다 `/var/share/ldomsmanager/ldom-db.xml` Logical Domains 제약 조건 데이터베이스 파일을 저장하고 복원해야 합니다.

또한 컨트롤 도메인의 파일 데이터를 파괴할 수 있는 다른 작업(예: 디스크 교체)을 수행할 때 `/var/share/ldomsmanager/ldom-db.xml` 파일을 저장하고 복원하십시오.

주 - `/var/share/ldomsmanager` 디렉토리는 모든 부트 환경 간에 공유됩니다. `/var/opt/SUNWldm`은 역호환성 목적상 `/var/share/ldomsmanager`에 대한 심볼릭 링크입니다.

출하 시 기본 구성과 도메인 사용 안함으로 설정

플랫폼이 하나의 운영체제만 호스트하는 단일 시스템으로 나타나는 초기 구성을 출하 시 기본 구성이라고 합니다. 논리적 도메인을 사용 안함으로 설정하려는 경우 이 구성을 복원하려는 것입니다. 그러면 시스템이 다른 도메인에 지정된 모든 리소스(CPU, 메모리, I/O)에 대한 액세스 권한을 되찾습니다.

이 절에서는 모든 게스트 도메인을 제거하고, 모든 도메인 구성을 제거하고, 구성을 출하 시 기본값으로 되돌리는 방법을 설명합니다.

▼ 모든 게스트 도메인을 제거하는 방법

1. 모든 도메인을 중지합니다.

```
primary# ldm stop-domain -a
```

2. `primary` 도메인을 제외한 모든 도메인의 바인드를 해제합니다.

```
primary# ldm unbind-domain ldom
```

주 - I/O 도메인이 컨트롤 도메인에 필요한 서비스를 제공하는 경우 바인드를 해제할 수 없습니다. 이 경우, 이 단계를 건너뛰십시오.

3. `primary` 도메인을 제외한 모든 도메인을 삭제합니다.

```
primary# ldm remove-domain -a
```

▼ 모든 도메인 구성을 제거하는 방법

1. 서비스 프로세서(SP)에 저장된 모든 도메인 구성을 나열합니다.

```
primary# ldm list-config
```

2. `factory-default` 구성을 제외한, 이전에 SP에 저장된 모든 구성(`config-name`)을 제거합니다. 각 구성에 대해 다음 명령을 사용합니다.

```
primary# ldm rm-config config-name
```

이전에 SP에 저장된 모든 구성을 제거한 후 컨트롤 도메인(`primary`)을 재부트할 때 사용할 그 다음 도메인은 `factory-default`입니다.

▼ 출하 시 기본 구성을 복원하는 방법

1. 출하 시 기본 구성을 선택합니다.

```
primary# ldm set-config factory-default
```

2. 컨트롤 도메인을 중지합니다.

```
primary# shutdown -i5 -g0 -y
```

3. 시스템의 전원을 껐다 켜서 출하 시 기본 구성을 로드합니다.

```
-> stop /SYS
-> start /SYS
```

▼ Logical Domains Manager를 사용 안함으로 설정하는 방법

Logical Domains Manager를 사용 안함으로 설정하면 실행 중인 도메인이 중지되지 않지만, 새 도메인을 만들거나 기존 도메인의 구성을 변경하거나 도메인의 상태를 모니터링하는 것이 불가능합니다.



주의 - Logical Domains Manager를 사용 안함으로 설정하면 이 작업으로 인해 오류 보고 및 전원 관리와 같은 일부 서비스가 사용 안함으로 설정됩니다. 오류 보고의 경우 `factory-default` 구성 상태에서 컨트롤 도메인을 재부트하여 오류 보고를 복원할 수 있습니다. 단, 전원 관리를 다시 사용으로 설정할 수 없습니다. 게다가, 일부 시스템 관리나 모니터링 도구는 Logical Domains Manager에 의존합니다.

- 컨트롤 도메인에서 **Logical Domains Manager**를 사용 안함으로 설정합니다.

```
primary# svcadm disable ldmd
```

▼ 서비스 프로세서에서 출하 시 기본 구성을 복원하는 방법

서비스 프로세서에서 출하 시 기본 구성을 복원할 수 있습니다.

1. 서비스 프로세서에서 출하 시 기본 구성을 복원합니다.

```
-> set /HOST/bootmode config=factory-default
```

2. 시스템의 전원을 껐다 켜서 출하 시 기본 구성을 로드합니다.

```
-> reset /SYS
```



전원 관리 사용

이 부록에는 Oracle VM Server for SPARC 시스템에서 전원 관리를 사용하는 방법에 대한 정보가 포함되어 있습니다.

전원 관리 사용

PM(전원 관리)을 사용으로 설정하려면 먼저 Oracle ILOM(Integrated Lights Out Manager) 3.0 펌웨어에서 PM 정책을 설정해야 합니다. 이 절에서는 Oracle VM Server for SPARC 소프트웨어에서 PM을 사용하기 위해 필요한 정보를 요약합니다.

ILOM에 대한 자세한 내용은 다음을 참조하십시오.

- *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide*의 “Monitoring Power Consumption”
- *Oracle Integrated Lights Out Manager(ILOM) 3.0* 기능 업데이트 및 릴리스 노트

전원 정책은 특정 시점의 시스템 전원 사용을 제어합니다. 기본 플랫폼에서 PM 기능을 구현했다면 다음 전원 정책이 지원됩니다.

- **사용 안함.** 시스템이 사용 가능한 모든 전원을 사용하도록 허용합니다.
- **성능.** 성능에 거의 영향을 끼치지 않는 다음 PM 기능을 하나 이상 사용으로 설정합니다.
 - CPU 코어 자동 사용 안함
 - CPU 클럭 사이클 건너뛰기
 - CPU DVFS(동적 전압 및 주파수 조정)
 - 일관성 링크 조정
 - Oracle Solaris PAD(Power Aware Dispatcher)
- **탄력적.** 성능 절에 설명된 PM 기능을 사용하여 시스템 전원 사용을 현재 사용률 수준으로 조정합니다. 예를 들어, 사용률이 줄어들면 리소스의 전원 상태가 감소합니다.

전원 관리 기능

PM 기능은 다음과 같습니다.

- **CPU 코어 자동 사용 안함.** 해당 코어의 모든 하드웨어 스트랜드(스트랜드)가 도메인에 바인드되어 있지 않으면 탄력적 또는 성능 정책이 적용 중인 경우 Logical Domains Manager가 자동으로 CPU 코어를 사용 안함으로 설정합니다. 이 기능은 UltraSPARC T2, UltraSPARC T2 Plus, SPARC T3 및 SPARC T4 플랫폼에 대해서만 사용할 수 있습니다.
- **CPU 클럭 사이클 건너뛰기.** 탄력적 정책이 적용 중인 경우 Logical Domains Manager가 도메인에 바인드된 다음 CPU 리소스에서 명령을 실행하는 클럭 사이클 수를 자동으로 조정합니다.
 - 프로세서(Oracle Solaris 10 또는 Oracle Solaris 11 OS를 실행하는 도메인의 SPARC T3 또는 SPARC T4)
 - 코어(Oracle Solaris 10 OS를 실행하는 도메인의 SPARC M5만)
 - 코어 쌍(Oracle Solaris 10 OS를 실행하는 SPARC T5 또는 SPARC M6만)
 - SCC(SPARC 캐시 클러스터)(Oracle Solaris 10 OS를 실행하는 도메인의 SPARC T7 시리즈 서버, SPARC M7 서버 및 SPARC S7 시리즈 서버만)

프로세서, 코어, 코어 쌍 또는 SCC 에 바인드된 스트랜드가 없는 경우에도 Logical Domains Manager가 사이클 건너뛰기를 적용합니다.

- **CPU DVFS(동적 전압 및 주파수 조정).** 탄력적 정책이 적용 중인 경우 Logical Domains Manager가 Oracle Solaris 10 OS를 실행하는 도메인에 바인드된 프로세서 또는 SCC의 클럭 주파수를 자동으로 조정합니다. 또한 Logical Domains Manager는 바인드된 스트랜드가 없는 SPARC T5, SPARC M5, SPARC M6 프로세서의 클럭 주파수를 줄입니다. SPARC T7 시리즈 서버에서는 SCC의 클럭 주파수를 줄입니다. 이 기능은 SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에서만 사용할 수 있습니다.
- **일관성 링크 조정.** 탄력적 정책이 적용 중인 경우 Logical Domains Manager는 하이퍼바이저가 사용 중인 일관성 링크 수를 자동으로 조정하도록 합니다. 이 기능은 SPARC T5-2 시스템에서만 사용할 수 있습니다.
- **전원 제한.** SPARC T3 서버, SPARC T4 서버, SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에서 전원 제한을 설정하여 시스템의 전원 소비를 제한할 수 있습니다. 전원 소비가 전원 제한보다 클 경우 PM은 전원을 줄이는 기술을 사용합니다. ILOM 서비스 프로세서(SP)를 사용하여 전원 제한을 설정할 수 있습니다.

다음 문서를 참조하십시오.

- *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide*
- *Oracle Integrated Lights Out Manager(ILOM) 3.0 기능 업데이트 및 릴리스 노트*

ILOM 인터페이스를 사용하여 전원 제한, 유예 기간 및 위반 작업을 설정할 수 있습니다. 전원 제한이 유예 기간보다 오랫동안 초과될 경우 위반 작업이 수행됩니다.

현재 전원 소비가 전원 제한을 초과할 경우 CPU의 전원 상태를 줄이기 위한 시도가 수행됩니다. 전원 소비가 전원 제한 아래로 낮아지면 이러한 리소스의 전원 상태는 다시 높아질 수 있습니다. 시스템에 탄력적 정책이 적용 중인 경우 리소스의 전원 상태 증가는 사용자 레벨로 결정됩니다.

- **Solaris PAD(Power Aware Dispatcher)**. Oracle Solaris 11.1 OS를 실행하는 게스트 도메인은 SPARC T5 서버, SPARC T7 시리즈 서버, SPARC M5 서버, SPARC M6 서버, SPARC M7 시리즈 서버 및 SPARC S7 시리즈 서버에서 PAD(Power Aware Dispatcher)를 사용하여 유휴 상태이거나 활용률이 낮은 리소스의 전원 소비를 최소화합니다. Logical Domains Manager 대신 PAD가 CPU 또는 SCC 클럭 사이클 건너뛰기 레벨 및 DVFS 레벨을 조정합니다.

ILOM 3.0 펌웨어 CLI를 사용하여 전원 정책을 구성하는 방법은 *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide*의 “Monitoring Power Consumption”을 참조하십시오.

전원 소비 데이터 보기

PM(전원 관리) 관찰성 모듈 및 `ldmpower` 명령을 통해 도메인에 대한 CPU 스레드 전원 소비 데이터를 볼 수 있습니다.

`ldmd/pm_observability_enabled` SMF(서비스 관리 기능) 등록 정보가 `true`로 설정되어 있으므로 PM 관찰성 모듈은 기본적으로 사용으로 설정됩니다. [ldmd\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

`ldmpower` 명령에는 전원 소비 보고 데이터를 사용자정의할 수 있는 다음 옵션 및 피연산자가 있습니다.

```
ldmpower [-ehiprstvx | -o hours | -m minutes] | -c resource [-l domain-name[,domain-name[,...]]]
          [interval [count]]
```

옵션에 대한 자세한 내용은 [ldmpower\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

이 명령을 권한 없는 사용자로 실행하려면 `LDoms Power Mgmt Observability` 권한 프로파일이 지정되어야 합니다. 이미 `LDoms Management` 또는 `LDoms Review` 권한 프로파일이 지정된 경우 `ldmpower` 명령을 실행할 수 있는 권한을 자동으로 가지게 됩니다.

Oracle VM Server for SPARC의 권한 사용 방법에 대한 자세한 내용은 “[Logical Domains Manager 프로파일 콘텐츠](#)” [30]를 참조하십시오.

사용자 또는 역할에 이러한 권한 프로파일을 직접 지정할 수 있습니다. 역할에 지정할 경우 해당 역할이 사용자에게 지정됩니다. 이러한 프로파일 중 하나를 사용자에게 직접 지정할 경우 `ldmpower` 명령을 사용하여 CPU 스레드 전원 소비 데이터를 보려면 `pfexec` 명령 또는 프로파일 셸(예: `pfbash` 또는 `pfksh`)을 사용해야 합니다. “[권한을 사용하여 논리적 도메인의 관리 위임](#)” [27]을 참조하십시오.

다음 예에서는 PM 관찰성 모듈을 사용으로 설정하고 도메인에 지정된 CPU에 대한 전원 소비 데이터를 수집하는 방법을 보여줍니다.

예 82 전원 관리 관찰성 모듈 사용으로 설정

다음 명령은 ldmd/pm_observability_enabled 등록 정보가 현재 false로 설정되어 있는 경우 등록 정보를 true로 설정하여 PM 관찰성 모듈을 사용으로 설정합니다.

```
# svccfg -s ldmd setprop ldmd/pm_observability_enabled=true
# svcadm refresh ldmd
# svcadm restart ldmd
```

예 83 프로파일 셸을 통해 역할 및 권한 프로파일을 사용하여 CPU 스레드 전원 소비 데이터 수집

- 다음 예에서는 ldmpower 명령을 실행하도록 허용하는 LDoms Power Mgmt Observability 권한 프로파일과 함께 ldmpower 역할을 만드는 방법을 보여줍니다.

```
primary# roleadd -P "LDoms Power Mgmt Observability" ldmpower
primary# passwd ldmpower
New Password:
Re-enter new Password:
passwd: password successfully changed for ldmpower
```

이 명령은 사용자 sam에게 ldmpower 역할을 지정합니다.

```
primary# usermod -R ldmpower sam
```

사용자 sam은 ldmpower 역할을 맡고 ldmpower 명령을 사용할 수 있습니다. 예를 들어, 다음과 같습니다.

```
$ id
uid=700299(sam) gid=1(other)
$ su ldmpower
Password:
$ pfexec ldmpower
Processor Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 75         84         86
gdom1   47         24         19
gdom2   10         24         26
```

- 다음 예에서는 권한 프로파일을 사용하여 ldmpower 명령을 실행하는 방법을 보여줍니다. 사용자에게 권한 프로파일을 지정합니다.

```
primary# usermod -P +"LDoms Power Mgmt Observability" sam
```

다음 명령은 사용자가 sam이며 All, Basic Solaris User 및 LDoms Power Mgmt Observability 권한 프로파일이 적용되는지 확인하는 방법을 보여줍니다.

```
$ id
uid=702048(sam) gid=1(other)
$ profiles
All
Basic Solaris User
LDoms Power Mgmt Observability
$ pfexec ldmpower
Processor Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 75          84          86
gdom1   47          24          19
gdom2   10          24          26
```

예 84 프로세서 전원 소비 데이터 보기

다음 예에서는 ldmpower를 사용하여 도메인에 대한 프로세서 전원 소비 데이터를 보고하는 방법을 보여줍니다.

- 다음 명령은 모든 도메인에 대한 15초, 30초 및 60초 간격 평균 프로세서 전원 소비 데이터를 보여줍니다.

```
primary# ldmpower
Processor Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 75          84          86
gdom1   47          24          19
gdom2   10          24          26
```

- 다음 명령은 모든 도메인 primary, gdom1 및 gdom2에 대해 추정된 전원 소비 데이터를 보여줍니다.

```
primary# ldmpower -x
System Power Consumption in Watts
DOMAIN 15_SEC_AVG 30_SEC_AVG 60_SEC_AVG
primary 585/57.47% 701/68.96% 712/70.22%
gdom1   132/12.97%  94/9.31%  94/9.30%
gdom2   298/29.27% 218/21.47% 205/20.22%
```

- 다음 명령은 gdom2 및 gdom5 도메인에 대한 순간 프로세서 전원 소비 데이터를 보여줍니다. 데이터를 10초마다 5회 보고합니다.

```
primary# ldmpower -it1 gdom2,gdom5 10 5
Processor Power Consumption in Watts
DOMAIN          TIMESTAMP          INSTANT
```

gdom2	2013.05.17 11:14:45	13
gdom5	2013.05.17 11:14:45	24
gdom2	2013.05.17 11:14:55	18
gdom5	2013.05.17 11:14:55	26
gdom2	2013.05.17 11:15:05	9
gdom5	2013.05.17 11:15:05	16
gdom2	2013.05.17 11:15:15	15
gdom5	2013.05.17 11:15:15	19
gdom2	2013.05.17 11:15:25	12
gdom5	2013.05.17 11:15:25	18

- 다음 명령은 모든 도메인에 대한 최근 12시간 동안의 평균 전원 소비 데이터를 보여줍니다. 데이터는 마지막으로 요청된 매시간 계산부터 시작하여 1시간 간격으로 표시됩니다.

```
primary# ldmpower -eto 12
```

```
Per domain MINIMUM and MAXIMUM power consumption ever recorded:
```

primary	2013.05.17 08:53:06	3	Min Processors
primary	2013.05.17 08:40:44	273	Max Processors
gdom1	2013.05.17 09:56:35	2	Min Processors
gdom1	2013.05.17 08:53:06	134	Max Processors
gdom2	2013.05.17 10:31:55	2	Min Processors
gdom2	2013.05.17 08:56:35	139	Max Processors
primary	2013.05.17 08:53:06	99	Min Memory
primary	2013.05.17 08:40:44	182	Max Memory
gdom1	2013.05.17 09:56:35	13	Min Memory
gdom1	2013.05.17 08:53:06	20	Max Memory
gdom2	2013.05.17 10:31:55	65	Min Memory
gdom2	2013.05.17 08:56:35	66	Max Memory

```
Processor Power Consumption in Watts
```

```
12 hour's worth of data starting from 2013.05.16 23:17:02
```

DOMAIN	TIMESTAMP	1 HOUR AVG
primary	2013.05.17 09:37:35	112
gdom1	2013.05.17 09:37:35	15
gdom2	2013.05.17 09:37:35	26
primary	2013.05.17 10:37:35	96
gdom1	2013.05.17 10:37:35	12
gdom2	2013.05.17 10:37:35	21
primary	2013.05.17 11:37:35	85

gdom1	2013.05.17 11:37:35	11
gdom2	2013.05.17 11:37:35	23
...		

용어 해설

이 용어 해설에서는 Oracle VM Server for SPARC 설명서의 용어, 약어 및 머리글자어를 정의합니다.

A

API Application Programming Interface의 머리글자어로, 응용 프로그래밍 인터페이스입니다.

ASN Abstract Syntax Notation의 머리글자어로, 추상 구문 표기법입니다.

auditing(감사) 시스템에 대한 변경사항을 추적하고 변경 사용자를 식별합니다.

auditreduce 감사 추적 파일에서 감사 레코드를 병합 및 선택하는 명령입니다([auditreduce\(1M\)](#) 매뉴얼 페이지 참조).

authorization (권한 부여) Oracle Solaris OS 권한을 사용하여 작업을 수행하고 데이터에 액세스할 수 있는 권한을 가진 사용자를 확인하는 방법입니다.

B

bge Broadcom BCM57xx 장치의 Broadcom 기가비트 이더넷 드라이버.

BSM Basic Security Module의 머리글자어로, 기본 보안 모듈입니다.

bsmconv BSM을 사용으로 설정하는 명령입니다([bsmconv\(1M\)](#) 매뉴얼 페이지 참조).

bsmunconv BSM을 사용 안함으로 설정하는 명령입니다([bsmunconv\(1M\)](#) 매뉴얼 페이지 참조).

C

CMT Chip Multithreading의 머리글자어로, 칩 멀티스레딩입니다.

compliance (준수)	시스템 구성이 미리 정의된 보안 프로파일을 준수하는지 여부를 확인합니다.
configuration (구성)	서비스 프로세서에 저장된 논리적 도메인 구성의 이름입니다.
constraints (제약 조건)	Logical Domains Manager에서 제약 조건은 특정 도메인에 지정하려는 하나 이상의 리소스입니다. 사용 가능한 리소스에 따라, 도메인에 추가하도록 요청한 리소스를 모두 받거나 아무 것도 얻지 못합니다.
control domain(컨트롤 도메인)	Logical Domains Manager를 사용하여 다른 논리적 도메인 및 서비스를 만들고 관리하는 권한 있는 도메인입니다.
CWQ	Control Word Queue: 암호화 장치.

D

DHCP	Dynamic Host Configuration Protocol, 동적 호스트 구성 프로토콜.
DIO	Direct I/O의 머리글자어로, 직접 I/O입니다.
DMA	직접 메모리 액세스는 CPU 관여 없이 메모리와 장치(예: 네트워크 카드) 간에 직접 데이터를 전송하는 기능입니다.
DMP	Dynamic Multipathing의 머리글자어로, 동적 다중 경로 제어(Veritas)입니다.
domain(도메인)	logical domain(논리적 도메인) 을 참조하십시오.
DPS	Data Plane Software의 머리글자어입니다.
DR	Dynamic Reconfiguration의 머리글자어로, 동적 재구성입니다.
drd	Logical Domains Manager용 Oracle Solaris OS 동적 재구성 데몬입니다(drd (1M) 매뉴얼 페이지 참조).
DRM	Dynamic Resource Management의 머리글자어로, 동적 리소스 관리입니다.
DS	Domain Services의 머리글자어로, 도메인 서비스 모듈입니다.
DVD	Digital Versatile Disc의 머리글자어로, 디지털 다기능 디스크입니다.

E

EFI	Extensible Firmware Interface의 머리글자어입니다.
------------	--

ETM Encoding Table Management의 머리글자어로, 인코딩 테이블 관리 모듈입니다.

F

FC_AL Fiber Channel Arbitrated Loop, 광섬유 채널 중재 루프.

FMA Fault Management Architecture의 머리글자어입니다.

fmd Oracle Solaris OS 결함 관리자 데몬입니다(**fmd(1M)** 매뉴얼 페이지 참조).

fmthard 하드 디스크에 레이블을 채우는 명령어입니다(**fmthard(1M)** 매뉴얼 페이지 참조).

format 디스크 분할 및 유지 관리 유틸리티(**format(1M)** 매뉴얼 페이지 참조)

G

GB Gigabit의 약어로, 기가비트입니다.

GLDv3 Generic LAN Driver 버전 3

guest domain(게스트 도메인) I/O 및 서비스 도메인에서 서비스를 사용하고 컨트롤 도메인에 의해 관리됩니다.

H

hardening(강화) 보안을 향상하도록 Oracle Solaris OS 구성을 수정합니다.

hypervisor(하이퍼바이저) 운영체제와 하드웨어 계층 사이에 위치한 펌웨어 계층입니다.

I

I/O 내부 디스크 및 PCIe 컨트롤러와 같은 입/출력 장치와 여기에 연결된 어댑터 및 장치입니다.

I/O domain(I/O 도메인) 물리적 I/O 장치에 직접 소유권과 직접 액세스를 보유한 도메인으로, 이러한 장치를 가상 장치 형태로 다른 논리적 도메인과 공유합니다.

IB	Infiniband의 약어입니다.
IDE	Integrated Drive Electronics의 머리글자어입니다.
IDR	Interim Diagnostics Release의 머리글자어입니다.
ILOM	Integrated Lights Out Manager의 머리글자어로, 운영체제와 별도로 서버를 관리할 수 있도록 해주는 하드웨어 및 지원 소프트웨어의 전용 시스템입니다.
ioctl	입/출력 제어 호출.
IPMP	Internet Protocol Network Multipathing, 인터넷 프로토콜 네트워크 다중 경로 제어.
 K	
kaio	커널 비동기 입/출력.
KB	Kilobyte의 약어로, 킬로바이트입니다.
KU	Kernel Update의 머리글자어로, 커널 업데이트입니다.
 L	
LAN	Local-Area Network의 머리글자어로, 근거리 통신망입니다.
LDAP	Lightweight Directory Access Protocol, 경량 디렉토리 액세스 프로토콜.
LDC	Logical Domain Channel의 머리글자어로, 논리적 도메인 채널입니다.
ldm	Logical Domains Manager 유틸리티(ldm(1M) 매뉴얼 페이지 참조)
ldmd	Logical Domains Manager 데몬입니다.
lofi	루프백 파일입니다.
logical domain(논리적 도메인)	별도의 논리적 리소스 그룹으로 구성된 가상 컴퓨터로, 단일 컴퓨터 시스템 안에 자체 운영체제와 신원을 가지고 있습니다. 도메인이라고도 합니다.
Logical Domains Manager	논리적 도메인을 만들고 관리하거나 도메인에 리소스를 할당하기 위한 CLI입니다.

M

MAC	매체 액세스 제어 주소로, Logical Domains Manager가 자동으로 지정하거나 사용자가 수동으로 지정할 수 있습니다.
MAU	Modular Arithmetic Unit의 머리글자어로, 모듈식 산술 장치입니다.
MB	Megabyte의 약어로, 메가바이트입니다.
MD	서버 데이터베이스의 시스템 설명.
mem, memory	메모리 단위 – 기본 크기인 바이트를 사용하거나 기가바이트(G), 킬로바이트(K), 메가바이트(M)를 지정합니다. 게스트 도메인에 할당할 수 있는 서버의 가상화된 메모리입니다.
metadb	Solaris Volume Manager 메타 장치 상태 데이터베이스의 복제본을 만들고 삭제하는 명령입니다(metadb(1M) 매뉴얼 페이지 참조).
metaset	디스크 세트를 구성하는 명령입니다(metaset(1M) 매뉴얼 페이지 참조).
mhd	다중 호스트 디스크 제어 작업을 수행하는 명령입니다(mhd(7i) 매뉴얼 페이지 참조).
MIB	Management Information Base의 머리글자어입니다.
minimizing (최소화)	필요한 최소 수의 코어 Oracle Solaris OS 패키지를 설치합니다.
MMF	Multimode Fiber의 머리글자어로, 다중 모드 광섬유입니다.
MMU	Memory Management Unit의 머리글자어로, 메모리 관리 장치입니다.
mpgroup	가상 디스크 장애 조치용 다중 경로 그룹 이름.
mtu	Maximum Transmission Unit의 머리글자어로, 최대 전송 단위입니다.

N

NIS	Network Information Services의 머리글자어로, 네트워크 정보 서비스입니다.
NIU	Network Interface Unit의 머리글자어로, 네트워크 인터페이스 장치(Oracle's Sun SPARC Enterprise T5120 및 T5220 서버)입니다.
NTS	Network Terminal Server의 머리글자어로, 네트워크 터미널 서버입니다.
NVRAM	Non-Volatile Random-Access Memory의 머리글자어입니다.

nxge NIU 10Gb 이더넷 어댑터용 드라이버.

O

OID 객체 식별자로, MIB에서 각 객체를 고유하게 식별하는 번호 시퀀스입니다.

OVF Open Virtualization Format의 머리글자어로, 개방형 가상화 형식입니다.

P

P2V Oracle VM Server for SPARC Physical-to-Virtual 변환 도구. [Oracle VM Server for SPARC 3.3 Administration Guide](#)의 19 장, “Oracle VM Server for SPARC Physical-to-Virtual Conversion Tool”를 참조하십시오.

PA Physical Address의 머리글자어로, 물리적 주소입니다.

PCI Peripheral component interconnect 버스.

PCI-X PCI Extended 버스의 약어입니다.

PCIe PCI EXPRESS 버스의 약어입니다.

pcpu Physical CPU의 약어로, 물리적 CPU입니다.

physical domain(물리적 도메인) 단일 Oracle VM Server for SPARC 인스턴스가 관리하는 리소스의 범위입니다. 지원되는 SPARC T-Series 서버 및 SPARC S-Series 서버와 마찬가지로 물리적 도메인은 전체 물리적 시스템일 수 있습니다. 또는 지원되는 SPARC M-Series 서버와 마찬가지로 전체 시스템일 수도 있고, 시스템의 하위 세트일 수도 있습니다.

physical function(물리적 기능) SR-IOV 사양에 정의된 대로 SR-IOV 기능을 지원하는 PCI 함수입니다. 물리 함수는 SR-IOV 기능 구조를 포함하며 SR-IOV 기능을 관리하는 데 사용됩니다. 물리 함수는 다른 PCIe 장치처럼 검색, 관리 및 조작할 수 있는 완전형 PCIe 함수입니다. 물리 함수에는 전체 구성 리소스가 있으며, PCIe 장치를 구성하고 제어하는 데 사용할 수 있습니다.

physio 물리적 입/출력입니다.

PICL Platform Information and Control Library, 플랫폼 정보 및 컨트롤 라이브러리.

picld PICL 데몬(picld(1M) 매뉴얼 페이지 참조)

PM 가상 CPU 및 메모리의 전원 관리.

praudit 감사 추적 파일의 내용을 인쇄하는 명령입니다(praudit(1M) 매뉴얼 페이지 참조).

PRI	Priority의 약어로, 우선 순위입니다.
R	
RA	Real Address의 머리글자어로, 실 주소입니다.
RAID	Redundant Array of Inexpensive Disks의 머리글자어로, 독립적인 디스크를 하나의 논리적 장치에 결합할 수 있도록 해줍니다.
RPC	Remote Procedure Call의 머리글자어로, 원격 프로시저 호출입니다.
S	
SAN	Storage Area Network의 머리글자어로, 저장 영역 네트워크입니다.
SASL	Simple Authentication and Security Layer, 단순 인증 및 보안 계층.
SAX	Simple API for XML(XML용 단순 API) 구문 분석기로, XML 문서를 순회합니다. SAX 구문 분석기는 이벤트 기반이며 대부분 데이터 스트리밍에 사용됩니다.
SCSA	Sun Common SCSI Architecture의 머리글자어로, Sun 공통 SCSI 아키텍처입니다.
SCSI HBA	SCSI Host Bus Adapter의 머리글자어로, 호스트 버스 어댑터입니다.
service domain(서비스 도메인)	다른 논리적 도메인에 가상 스위치, 가상 콘솔 커넥터, 가상 디스크 서버와 같은 장치를 제공하는 논리적 도메인입니다.
SMA	System Management Agent의 머리글자어입니다.
SMF	Service Management Facility의 머리글자어로, 서비스 관리 기능입니다.
SMI	Structure of Management Information의 머리글자어로, MIB에 사용할 관리 객체를 정의하고 그룹화합니다.
SNMP	Simple Network Management Protocol, 단순 네트워크 관리 프로토콜.
SP(서비스 프로세서)	SP는 시스템 컨트롤러(SC)라고도 하며, 물리적 컴퓨터를 모니터링하고 실행합니다.
SR-IOV	Single root I/O virtualization, 단일 루트 입/출력 가상화.
SSH	Secure Shell의 약어로, 보안 셸입니다.

ssh	보안 셸 명령(ssh(1) 매뉴얼 페이지 참조)
sshd	보안 셸 데몬(sshd(1M) 매뉴얼 페이지 참조)
SunVTS	Sun Validation Test Suite, Sun 검증 테스트 모음.
svcadm	서비스 인스턴스 조작(svcadm(1M) 매뉴얼 페이지 참조)
system controller (SC, 시스템 컨트롤러)	서비스 프로세서를 참조하십시오.
T	
TLS	Transport Layer Security의 머리글자어로, 전송 계층 보안입니다.
U	
UDP	User Datagram Protocol의 머리글자어로, 사용자 다이어그램 프로토콜입니다.
unicast(유니캐스트)	단일 발신자와 단일 수신자 사이에 발생하는 네트워크 통신입니다.
uscsi	사용자 SCSI 명령 인터페이스(uscsi(7) 매뉴얼 페이지 참조)
UTP	Unshielded Twisted Pair의 머리글자어입니다.
V	
var	Variable의 약어로, 변수입니다.
VBSC	Virtual blade system controller, 가상 블레이드 시스템 컨트롤러.
vcc, vconscn	특정 포트 범위의 가상 콘솔 집중기 서비스를 게스트 도메인에 지정합니다.
vcons, vconsole	시스템 레벨 메시지에 액세스하기 위한 가상 콘솔입니다. 특정 포트의 컨트롤 도메인에서 vconscn 서비스에 연결합니다.
vcpu	가상 중앙 처리 장치. 서버의 각 코어는 가상 CPU로 표현됩니다.
vdcc	Virtual Disk Client의 머리글자어로, 가상 디스크 클라이언트입니다.

vdisk	가상 디스크는 여러 유형의 물리적 장치, 볼륨, 파일과 연관된 일반 블록 장치입니다.
vds, vdiskserver	가상 디스크 서버에서 가상 디스크를 논리적 도메인으로 가져올 수 있습니다.
vdsdev, vdiskserverdevice	가상 디스크 서버가 가상 디스크 서버 장치를 내보냅니다. 장치는 전체 디스크, 디스크의 슬라이스, 파일 또는 디스크 볼륨일 수 있습니다.
virtual function (가상 기능)	물리 함수와 연관된 PCI 함수입니다. 가상 함수는 하나 이상의 물리적 리소스를 물리 함수와 공유하거나, 동일한 물리 함수로 연관된 다른 가상 함수와 공유하는 경량 PCIe 함수입니다. 가상 함수는 자체 동작을 위한 구성 리소스만 가질 수 있습니다.
vldc	가상 논리적 도메인 채널 서비스.
vldcc	가상 논리적 도메인 채널 클라이언트.
vnet	가상 네트워크 장치는 가상 이더넷 장치를 구현하고 가상 네트워크 스위치(vswitch)를 사용하여 시스템의 다른 vnet 장치와 통신합니다.
VNIC	가상 네트워크 인터페이스 카드로, 물리적 네트워크 장치를 가상 인스턴스화한 것이며 물리적 네트워크 장치에서 만들어 영역에 지정할 수 있습니다.
vNTS	Virtual network terminal service, 가상 네트워크 터미널 서비스.
vntsd	도메인 콘솔용 Oracle Solaris OS 가상 네트워크 터미널 서버 데몬입니다(vntsd(1M) 매뉴얼 페이지 참조).
volfs	볼륨 관리 파일 시스템(volfs(7FS) 매뉴얼 페이지 참조)
vsw, vswitch	가상 네트워크 장치를 외부 네트워크에 연결하고 장치 사이에 패킷을 전환하는 가상 네트워크 스위치입니다.
VTOC	Volume table of contents, 볼륨 목차.
VxDMP	Veritas Dynamic Multipathing의 약어입니다.
VxVM	Veritas Volume Manager의 약어입니다.
X	
XFP	eXtreme Fast Path의 머리글자어입니다.
XML	Extensible Markup Language의 머리글자어입니다.
XMPP	Extensible Messaging and Presence Protocol, 확장성 메시징 및 프레즌스 프로토콜.

Z

ZFS Zettabyte File System의 머리글자어로, 제타바이트 파일 시스템입니다.

zpool ZFS 저장소 풀([zpool\(1M\) 매뉴얼 페이지 참조](#))

ZVOL ZFS Volume Emulation Driver, ZFS 볼륨 에뮬레이션 드라이버.

색인

번호와 기호

- `/etc/system` 파일
 - 업데이트, 352
- `cancel-reconf` 하위 명령, 300
- CD 또는 DVD 이미지
 - 서비스 도메인에서 게스트 도메인으로 내보내기, 182
 - 여러 번 내보내기, 182
- CD 이미지
 - 내보내기, 181
- CLI 살펴볼 내용 명령줄 인터페이스
- CPU 동적 리소스 관리, 311
- CPU 리소스
 - 할당, 302
- CPU 매핑, 360
- CPU 및 메모리 주소 매핑
 - 문제 해결, 360
- CPU 전원 관리, 311
- CPU 전체 코어
 - 기존 도메인 구성, 308
 - 도메인 구성, 306
 - 도메인 만들기, 307
 - 시스템 재바인드, 311
 - 시스템 재부트, 311
 - 컨트롤 도메인 구성, 308
- CPU 코어 사용 안함, 378
- CPU 클럭 사이클 건너뛰기, 378
- CPU 할당, 302
- CPU DR, 304, 309
- CPU DVFS(동적 전압 및 주파수 조정), 378
- daemons
 - `drd`, 300
- DIO(직접 I/O)
 - 계획, 139
- DR 살펴볼 내용 동적 재구성(DR)
 - DVD 이미지
 - 내보내기, 181
 - dynamic reconfiguration daemon (`drd`), 300
 - FMA 살펴볼 내용 FMA(Fault Management Architecture)
 - FMA(Fault Management Architecture)
 - 차단 목록, 345
 - `format`
 - 가상 디스크, 185
 - GLD 준수(Oracle Solaris 10)
 - 네트워크 장치, 239
 - I/O 가상화
 - PCIe 버스에 대해 사용으로 설정, 154
 - 사용으로 설정, 83
 - I/O 도메인, 63, 73, 135
 - PCIe SR-IOV 가상 기능 사용, 73
 - PCIe 버스, 63
 - SR-IOV 가상 기능을 지정하여 만들기, 73
 - SR-IOV 가상 기능을 지정하여 부트, 97
 - 끝점 장치를 지정하여 만들기, 135
 - 마이그레이션 제한 사항, 63
 - 만들기 지침, 64
 - I/O 리소스
 - 비워짐으로 표시, 350
 - ILOM 상호 연결 구성
 - 확인, 40
 - ILOM 상호 연결 서비스
 - 사용으로 설정, 40
 - InfiniBand SR-IOV
 - 요구사항, 103
 - inter-vnet-link
 - PVLAN, 256
 - IPMP
 - Oracle VM Server for SPARC 환경에서 구성, 243

- 가상 네트워크 장치를 그룹으로 구성, 243, 245
- 서비스 도메인에서 구성, 246
- ISO 이미지
 - 내보내기, 181
 - 서비스 도메인에서 게스트 도메인으로 내보내기, 183
- JumpStart
 - 사용하여 게스트 도메인에 Oracle Solaris 10 OS 설치, 50
- LDC 살피볼 내용 LDC(논리적 도메인 채널)
- LDC(논리적 도메인 채널), 22, 363
- ldmd 살피볼 내용 Logical Domains Manager 데몬
- lofi
 - 파일 및 볼륨을 가상 디스크로 내보내기, 176
- Logical Domains 제약 조건 데이터베이스
 - 복원, 374
 - 저장, 374
- Logical Domains Manager, 20, 22
 - 데몬(ldmd), 23
 - 사용 안함으로 설정, 375
- Logical Domains Manager 데몬
 - 사용으로 설정, 372
- MAC 주소
 - 도메인에 지정, 237
 - 수동으로 지정, 237
 - 자동 지정 알고리즘, 238
 - 자동으로 지정, 237
 - 중복 감지, 238
 - 지정, 237
- NAT
 - Oracle Solaris 10 시스템에서 구성, 242
 - Oracle Solaris 11 시스템에서 구성, 240
 - 가상 스위치 및 서비스 도메인 구성, 240
- Oracle Solaris 10 네트워킹, 221
- Oracle Solaris 11 네트워킹, 218
- Oracle Solaris 11 네트워킹 관련 기능 차이점, 277
- Oracle Solaris OS
 - Oracle VM Server for SPARC와 함께 작동, 354
 - 게스트 도메인에 설치, 46
 - DVD에서, 47
 - ISO 파일에서, 48
 - 네트워크 인터페이스 이름(Oracle Solaris 11) 찾기, 235
 - 중단, 354
- Oracle VM Server for SPARC
 - 문제 해결, 25
 - 서비스 프로세서와 함께 사용, 355
 - Oracle VM Server for SPARC MIB, 25
 - Oracle VM Server for SPARC용, 25
 - PCIe 버스, 63
 - I/O 가상화 사용으로 설정, 83
 - 하드웨어 변경, 141
 - PCIe SR-IOV 가상 기능 살피볼 내용 가상 기능 계획, 84
 - physical-bindings 제약 조건
 - 제거, 313
 - PM(전원 관리), 378, 378, 378, 378, 378, 379
 - primary 도메인, 22
 - PVLAN
 - inter-vnet-link, 256
 - 마이그레이션 제한 사항, 257
 - 만들기, 258
 - 업데이트, 258
 - 요구사항, 257
 - 정보 나열, 259
 - 제거, 258
 - 제한 사항, 257
 - PVLAN(전용 VLAN)
 - 사용, 256
 - ro 옵션
 - 가상 디스크 백엔드, 168
 - SCSI 및 가상 디스크, 185
 - SCSI 및 가상 SCSI HBA, 214
 - slice 옵션
 - 가상 디스크 백엔드, 169
 - Solaris PAD(Power Aware Dispatcher), 379
 - Solaris Volume Manager
 - 가상 디스크에서 사용, 191
 - 사용, 192
 - SR-IOV, 73
 - 기능 유형, 74
 - 동적, 81
 - 동적 요구사항, 81, 81
 - 요구사항, 76
 - 이더넷 장치 관련 등록 정보, 86
 - 정적, 80
 - 정적 요구사항, 80
 - 제한 사항, 79
 - SR-IOV 가상 기능 살피볼 내용 가상 기능
 - SSL 인증서

- 마이그레이션, 297
 - SUNwldm 패키지, 23
 - UUID(universally unique identifiers), 362
 - virtinfo
 - 가상 도메인 정보, 362
 - VLAN
 - Oracle Solaris 10 게스트 도메인에서 지정, 255
 - Oracle Solaris 10 서비스 도메인에서 지정, 254
 - Oracle Solaris 11 게스트 도메인에서 지정, 254
 - Oracle Solaris 11 서비스 도메인에서 지정, 253
 - 사용, 253
 - 지정, 253
 - VLAN 태그 지정
 - 사용, 252
 - VLAN ID(VID), 253
 - Vnet 간 LDC 채널, 225
 - VNIC
 - SR-IOV 가상 기능 만들기, 99
 - VxVM
 - 가상 디스크에서 사용, 191
 - 사용, 193
 - ZFS
 - 가상 디스크 및, 186
 - 가상 디스크에서 사용, 192
 - 디스크 이미지 저장, 186
 - ZFS 볼륨
 - 가상 디스크 백엔드를 여러 번 내보내기, 165
 - 단일 슬라이스 디스크로 내보내기, 174
 - 디스크 이미지 저장, 187
 - 전체 디스크로 내보내기, 173
 - ZFS 파일
 - 디스크 이미지 저장, 187
 - ZFS 풀
 - 서비스 도메인에서 구성, 186
- ㄱ
- 가상 기능, 84
 - I/O 도메인 만들기, 100, 100
 - I/O 도메인 이더넷 네트워크 부트, 97
 - I/O 도메인에 InfiniBand 추가, 107
 - I/O 도메인에 광 섬유 채널 추가, 125
 - I/O 도메인에 이더넷 추가, 95
 - I/O 도메인에서 InfiniBand 제거, 109
 - I/O 도메인에서 광 섬유 채널 제거, 126
 - I/O 도메인에서 제거, 96
 - InfiniBand, 103
 - InfiniBand 나열, 112
 - InfiniBand 만들기, 103
 - InfiniBand 삭제, 106
 - 게스트 도메인에서 광 섬유 채널 액세스, 127
 - 광 섬유 채널, 115
 - 광 섬유 채널 등록 정보 수정, 124
 - 광 섬유 채널 만들기, 118
 - 광 섬유 채널 삭제, 122
 - 광 섬유 채널 요구사항, 115, 116
 - 광 섬유 채널 제한 사항, 116
 - 루트 도메인에 InfiniBand 추가, 110
 - 루트 도메인에서 InfiniBand 제거, 111
 - 이더넷, 85, 86
 - 이더넷 VNIC 만들기, 99
 - 이더넷 등록 정보 수정, 93
 - 이더넷 만들기, 86, 86
 - 이더넷 삭제, 86, 91
 - 장치 관련 광 섬유 채널 등록 정보, 116
 - 가상 기능에 대한 광 섬유 채널 WWN(world-wide name)
 - 할당, 117
 - 가상 네트워크, 218
 - 성능 최대화, 222, 223
 - 가상 네트워크 장치, 225
 - 물리적 네트워크 대역폭의 양 제어, 231
 - 가상 네트워크 터미널 서버 데몬(vntsd)과 , 24
 - 가상 네트워크 터미널 서버 도메인(vntsd)
 - 사용으로 설정, 39
 - 가상 도메인 정보
 - API, 362
 - virtinfo, 362
 - 가상 디스크, 161
 - format 명령 및, 185
 - SCSI 및, 185
 - Solaris Volume Manager에서 사용, 191
 - VxVM에서 사용, 191
 - ZFS에서 사용, 186, 192
 - 관리, 164
 - 다중 경로, 176, 178
 - 다중 경로 구성, 178
 - 단일 슬라이스 디스크로 백엔드 내보내기, 167
 - 디스크 식별자, 163
 - 문제, 193

- 물리적 디스크 슬라이스에서 내보내기, 171
- 물리적 디스크에서 내보내기, 170
- 백엔드, 169
- 백엔드 `excl` 옵션, 168
- 백엔드 `ro` 옵션, 168
- 백엔드 `slice` 옵션, 169
- 백엔드 내보내기, 165
- 백엔드 옵션, 168
- 볼륨 관리자에서 사용, 190
- 시간 초과, 178, 184
- 시간 초과 옵션 수정, 166
- 옵션 수정, 166
- 장치 이름, 163
- 전체 디스크로 백엔드 내보내기, 167
- 제거, 166
- 추가, 165
- 표시, 166
- 가상 스위치, 223
 - Oracle Solaris 11 도메인에 대한 외부 연결을 제공 하도록 구성, 241
 - 기본 인터페이스로 구성, 38
 - 도메인에 대한 외부 연결을 제공하기 위해 Oracle Solaris 10 서비스 도메인에 구성, 243
- 가상 입/출력, 23
- 가상 장치
 - I/O, 24
 - 가상 네트워크(`vnet`), 24
 - 가상 디스크 서비스(`vds`), 24
 - 가상 디스크 클라이언트(`vdc`), 24
 - 가상 스위치(`vsw`), 24
 - 가상 콘솔 집중기(`vcc`), 24
- 가상 장치 식별자, 234
- 가상 컴퓨터, 22
- 가상 CPU
 - 해당하는 물리적 CPU 번호 확인, 361
- 가상 SCSI HBA
 - SCSI 및, 214
 - 관리, 201
 - 다중 경로, 205
 - 다중 경로 구성, 208
 - 모양, 204
 - 시간 초과, 203, 213
 - 식별자, 200
 - 장치 이름, 200
 - 존재 여부 확인, 203
- 가상 SCSI HBA 및 가상 SAN 구성, 205
- 게스트 도메인, 23
 - 마이그레이션, 297
 - 마이그레이션 및 이름 바꾸기, 298
 - 만들기, 43
 - 모두 제거, 374
 - 시작, 43
- 게스트 콘솔
 - 네트워크를 통해 연결, 59
- 결정
 - 도메인 구성, 361
- 결함이 있는 하드웨어 리소스
 - 구성 해제, 345
 - 도메인 복구, 347
 - 차단 목록, 345
- 결합
 - 단일 그룹으로 콘솔, 60
- 경로 지정
 - 가상 스위치 및 서비스 도메인 구성, 240
- 계획
 - DIO(직접 I/O), 139, 139
 - PCIe SR-IOV 가상 기능, 84
 - 이더넷 SR-IOV, 86
- 관리
 - 가상 SCSI HBA, 201
 - 가상 디스크, 164
 - 도메인 구성, 337
 - 리소스 그룹, 322
 - `비primary` 루트 도메인에서 SR-IOV 가상 기능, 157
 - `비primary` 루트 도메인에서 직접 I/O 장치, 156
 - 컨트롤 도메인에서 물리적 리소스, 314
- 관리 권한 위임
 - 권한 프로파일, 27
- 구문 분석이 가능한 도메인 목록
 - 보기, 358
 - 표시, 358
- 구성
 - CPU 전체 코어로 기존 도메인, 308
 - CPU 전체 코어로 도메인, 306
 - CPU 전체 코어로 컨트롤 도메인, 308
 - NAT 및 경로 지정을 위한 가상 스위치 및 서비스 도메인, 240
 - Oracle Solaris 10 시스템에서 NAT, 242
 - Oracle Solaris 11 도메인에 대한 외부 연결을 제공 할 가상 스위치, 241

- Oracle Solaris 11 시스템에서 NAT, 240
- Oracle VM Server for SPARC 환경에서 IPMP, 243
- 가상 SCSI HBA 다중 경로, 208
- 가상 네트워크 장치를 IPMP 그룹으로, 243, 245
- 가상 디스크 다중 경로, 178
- 가상 스위치를 기본 인터페이스로, 38
- 도메인 종속성, 356
- 도메인에 대한 외부 연결을 제공하기 위해 Oracle Solaris 10 서비스 도메인에 가상 스위치, 243
- 마이그레이션용 SSL 인증서, 282
 - Oracle Solaris 11, 282
- 물리적 링크 상태 업데이트, 249
- 부트하도록 선택, 25
- 서비스 도메인에서 IPMP, 246
- 서비스 도메인의 ZFS 풀, 186
- 성능 레지스터 액세스, 332
- 정보 프레임, 264
- 컨트롤 도메인, 35
- 하드 분할을 사용하여 시스템, 305
- 구성 해제
 - 결함이 있는 하드웨어 리소스, 345
- 권한 부여
 - `ldm` 하위 명령, 30
- 권한 프로파일
 - 지정, 27, 28

- L**
- 나열
 - InfiniBand 가상 기능, 112
 - PVLAN 정보, 259
 - 도메인 리소스, 326
 - 리소스 제약 조건, 331
 - 시스템에서 읽을 수 있는 출력으로 리소스, 327
- 내보내기
 - CD 또는 DVD 이미지 여러 번, 182
 - CD 이미지, 181
 - DVD 이미지, 181
 - ISO 이미지, 181
 - ZFS 볼륨을 단일 슬라이스 디스크로, 174
 - ZFS 볼륨을 전체 디스크로, 173
 - 가상 디스크 백엔드, 165
 - 디스크 슬라이스
 - 간접, 176
 - 직접, 176
 - 물리적 디스크 슬라이스를 가상 디스크로, 171
 - 물리적 디스크를 가상 디스크로, 170
 - 백엔드
 - 비교, 175
 - 백엔드, 요약, 175
 - 볼륨, 172
 - 역호환성, 175
 - 서비스 도메인에서 게스트 도메인으로 CD 또는 DVD 이미지, 182
 - 서비스 도메인에서 게스트 도메인으로 ISO 이미지, 183
 - 슬라이스 2, 172
 - 파일, 172
 - 파일 또는 볼륨을 단일 슬라이스 디스크로, 174
 - 파일 또는 볼륨을 전체 디스크로, 172
 - 파일 및 볼륨을 가상 디스크로
 - `lofi`, 176
 - 지침, 176
 - 파일을 전체 디스크로, 172
 - 네트워크 구성
 - 이더넷 SR-IOV, 97
 - 네트워크 부트
 - 이더넷 SR-IOV 가상 기능을 사용하여 I/O 도메인, 97
 - 네트워크 인터페이스 이름, 234
 - 찾기, 236
 - 네트워크 장치
 - 네트워크 대역폭 제한, 설정, 231
 - 사용, 239
 - 네트워크 장치 구성
 - 보기, 228
 - 네트워크 장치 통계
 - 보기, 228
 - 논리적 도메인 채널(LDC)
 - Vnet 간, 225
 - 누락된 하드웨어 리소스
 - 도메인 복구, 347

- ㄷ**
- 다중 경로 살펴볼 내용 가상 SCSI HBA 다중 경로 살펴볼 내용 가상 디스크 다중 경로
 - 데몬
 - `ldmd`, 23

- vntsd, 24
 - 도메인
 - 마이그레이션, 279
 - 복제본을 사용하여 프로비전, 188
 - 부하가 높은 중지, 353
 - 사용 안함으로 설정, 374
 - 서비스, 24
 - 역할, 22, 22
 - 유형, 22, 22, 23, 23
 - 저하됨으로 표시, 350
 - 정의, 20
 - 종속성, 356
 - 종속성 구성, 356
 - 종속성 주기, 358
 - 종속성에 대한 실패 정책 구성, 357
 - 도메인 구성
 - ldm add-domain을 사용하여 XML 파일에서 복원, 343
 - ldm init-system을 사용하여 XML 파일에서 복원, 342
 - 결정, 361
 - 관리, 337
 - 모두 제거, 375
 - 복원, 338, 342
 - 자동 복구 정책, 339, 339
 - 자동 저장을 사용하여 복원, 338
 - 저장, 338, 340
 - 저하됨, 349
 - 지속적, 25
 - 확인, 306
 - 도메인 구성에 대한 복구 모드, 345
 - 제어, 350
 - 도메인 구성에 대한 자동 복구 정책, 339, 339
 - 도메인 리소스
 - 나열, 326
 - 도메인 마이그레이션, 287
 - CPU에 대한 요구사항, 289
 - dry run 수행, 287
 - OpenBoot PROM 또는 커널 디버거에서, 294
 - PCIe 끝점 장치에 대한 요구사항, 292, 295
 - SR-IOV 가상 기능에 대한 요구사항, 293, 295
 - 가상 I/O 장치에 대한 요구사항, 291, 295
 - 다른 도메인에 대한 작업, 294
 - 메모리에 대한 요구사항, 290
 - 모니터링 진행, 295
 - 물리적 I/O 장치에 대한 요구사항, 291
 - 바인드되거나 비활성 도메인, 294
 - 보안, 281
 - 비대화식, 297
 - 비대화식 수행, 288
 - 상태 확인, 298
 - 소프트웨어 호환성, 281
 - 실패 메시지, 298
 - 실패 복구, 297
 - 암호화 장치에 대한 요구사항, 293
 - 작업, 280
 - 진행 중 취소, 296
 - 활성, 288
 - 활성 도메인에 대한 지연된 재구성, 293
 - 활성 도메인에 전원 관리 탄력적 정책이 적용될 때, 293
 - 도메인 마이그레이션 제한 사항, 285
 - 도메인 목록
 - 구문 분석 가능, 358
 - 도메인 콘솔
 - 액세스 제어, 53
 - 동적 경로 선택, 180
 - 동적 리소스 관리, 304
 - CPU, 311
 - 사용, 324
 - 동적 재구성(DR), 300, 317
 - CPU, 304, 309
 - 메모리, 315
 - 일부 메모리 요청, 316
 - 등록 정보
 - 광 섬유 채널 가상 기능 장치 관련 등록 정보, 116
 - 이더넷 SR-IOV 장치 관련, 86
 - 디스크 슬라이스 살피볼 내용 물리적 디스크 슬라이스 디스크 이미지
 - ZFS 볼륨을 사용하여 저장, 187
 - ZFS 파일을 사용하여 저장, 187
 - ZFS를 사용하여 저장, 186
 - 구성되지 않은 시스템의 스냅샷 만들기, 189
 - 스냅샷 만들기, 187, 188
- 르**
- 루트 도메인, 23, 65
 - PCIe 버스를 지정하여 만들기, 65
 - 만들기, 68

- 재부트, 134, 140
- 리소스, 21
 - 살펴볼 다른 내용 가상 장치 정의, 21
 - 출력의 플래그 정의, 327
 - 할당, 301
- 리소스 관리
 - 동적, 304
- 리소스 구성, 25
- 리소스 그룹
 - 관리, 322
 - 요구사항, 323
 - 제한 사항, 323
- 리소스 제약 조건
 - 나열, 331
- 리소스 할당, 301
- 링크 기반 IPMP
 - 사용, 248
- 링크 통합
 - 가상 스위치에서 사용, 262

-
- 마스터 도메인
 - 지정, 357
- 마이그레이션
 - SSL 인증서 사용, 297
 - 게스트 도메인, 297
 - 도메인, 279
 - 비대화식, 297
- 마이그레이션 및 이름 바꾸기
 - 게스트 도메인, 298
- 마이그레이션 제한 사항
 - I/O 도메인, 63
 - PVLAN, 257
- 마이그레이션용 FIPS 140-2 모드, 283
- 마이그레이션용 SSL 인증서
 - 구성, 282
 - Oracle Solaris 11, 282
- 만들기
 - CPU 전체 코어로 도메인, 307
 - I/O 도메인 이더넷 가상 기능, 100
 - InfiniBand 가상 기능, 103
 - PVLAN, 258
 - 게스트 도메인, 43
- 광 섬유 채널 가상 기능, 118
- 구성되지 않은 시스템의 디스크 이미지 스냅샷, 189
- 디스크 이미지 스냅샷, 187, 188
- 역할, 29
- 이더넷 가상 기능, 86, 86
- 이더넷 가상 기능에 vNIC, 99
- 전체 PCIe 버스에서 루트 도메인, 68
- 컨트롤 도메인에서 기본 서비스, 34
- 메모리
 - 도메인에 대한 크기 설정, 321
 - 도메인에 추가, 316
 - 도메인에서 제거, 316
 - 매핑, 360
 - 바인드된 도메인에 대한 정렬, 318
 - 비활성 도메인에 대한 정렬, 318
 - 정렬, 318
 - 활성 도메인, 318
 - 정렬되지 않은 추가, 319
 - 컨트롤 도메인에서 줄이기, 317
- 메모리 동적 재구성
 - 바인드된 도메인에서 작업, 320
 - 일부 요청, 316
 - 활성 도메인에서 작업, 320
- 메모리 동적 재구성(DR), 315
- 메모리 재구성
 - 컨트롤 도메인, 317
- 메모리 크기 요구사항, 46
- 메모리 DR 살펴볼 내용 메모리 동적 재구성(DR) 명령줄 인터페이스, 23
- 문제 해결
 - CPU 및 메모리 주소 매핑, 360
 - Oracle VM Server for SPARC, 25
- 물리적 네트워크 대역폭
 - 가상 네트워크 장치에서 사용 제어, 231
 - 설정 제한, 232
 - 제한, 232
- 물리적 디스크, 170
 - 가상 디스크로 내보내기, 170
- 물리적 디스크 슬라이스, 171
 - 가상 디스크로 내보내기, 171
- 물리적 디스크 LUN, 170
- 물리적 리소스
 - 관리 제한 사항, 315
 - 도메인에 지정, 311

- 컨트롤 도메인에서 관리, 314
- 물리적 링크 상태 업데이트
 - 구성, 249
- 물리적 장치, 22, 24
- 물리적 컴퓨터, 22
- 물리적 CPU 번호
 - 해당하는 가상 CPU 확인, 361
- 물리적으로 바인드되지 않은 리소스
 - 제거, 314

ㅂ

- 백엔드, 175
 - 살펴볼 다른 내용 가상 디스크 백엔드 내보내기
- 버스 지정
 - 동적, 67
 - 정적, 66
- 변경
 - PCIe 하드웨어로 변경, 141
- 보기
 - 구문 분석이 가능한 도메인 목록, 358
 - 네트워크 장치 구성, 228
 - 네트워크 장치 통계, 228
 - 전원 소비 데이터, 379
 - 프로세서 전원 소비 데이터, 381
- 복구
 - 결함이 있는 하드웨어 리소스가 있는 도메인, 347
 - 누락된 하드웨어 리소스가 있는 도메인, 347
 - 실패한 도메인 마이그레이션에서, 297
- 복원
 - Logical Domains 제약 조건 데이터베이스, 374
 - 도메인 구성, 338, 342
 - ldm add-domain을 사용하여 XML 파일에서, 343
 - ldm init-system을 사용하여 XML 파일에서, 342
 - 서비스 프로세서에서 출하 시 기본 구성, 376
 - 자동 저장 구성 디렉토리, 373
 - 출하 시 기본 구성, 375
- 복제
 - 부트 디스크 이미지, 188
- 볼륨 관리자
 - 가상 디스크에서 사용, 190
- 부트
 - SCSI DVD 장치에서, 213
- 부트 디스크 이미지
 - 복제, 188

- 비대화식 도메인 마이그레이션, 297
- 비워진 I/O 리소스, 350
- 비primary 루트 도메인, 151
 - PCIe SR-IOV 가상 기능 지정, 151
 - PCIe 끝점 장치 지정, 151
 - SR-IOV 가상 기능 관리, 157
- 개요, 151
- 제한 사항, 152, 153
- 직접 I/O 장치 관리, 156

ㅅ

- 사용
 - VLAN, 253
 - 가상 네트워크 터미널 서버 도메인(vntsd), 39
 - 링크 기반 IPMP, 248
 - 확인된 부트, 31
- 사용 안함으로 설정
 - Logical Domains Manager, 375
 - 도메인, 374
- 사용으로 설정
 - I/O 가상화, 83
 - ILOM 상호 연결 서비스, 40
 - Logical Domains Manager 데몬, 372
 - PCIe 버스에 대한 I/O 가상화, 154
 - 전원 관리 관찰성 모듈, 380
- 삭제, 86
 - 살펴볼 다른 내용 제거
 - InfiniBand 가상 기능, 106
 - 광 섬유 채널 가상 기능, 122
 - 이더넷 가상 기능, 86, 91
- 서버
 - 전원 꺾다 켜기 수행, 354
- 서비스 도메인, 22, 24
 - ZFS 풀 구성, 186
- 서비스 프로세서(SP)
 - Oracle VM Server for SPARC 사용, 355
 - 물리적 컴퓨터 모니터링 및 실행, 22
 - 출하 시 기본 구성 복원, 376
- 설정
 - 도메인에 대한 메모리 크기, 321
 - 물리적 네트워크 대역폭 제한, 232
 - 전원 제한, 378
- 설치
 - DVD에서 Oracle Solaris OS, 47

- ISO 파일에서 Oracle Solaris OS, 48
- JumpStart(Oracle Solaris 10) 사용, 50
- 게스트 도메인에 Oracle Solaris OS, 46
- 설치 서버가 VLAN에 있을 때 게스트 도메인, 255
- 성능
 - 가상 네트워크 최대화를 위한 요구사항, 222
 - 가상 네트워크에 대해 최대화, 222, 223
- 성능 레지스터 액세스
 - 설정, 332
- 수정
 - 가상 디스크 시간 초과 옵션, 166
 - 가상 디스크 옵션, 166
 - 광 섬유 채널 가상 기능 등록 정보, 124
 - 도메인 구성 자동 복구 정책, 339
 - 이더넷 SR-IOV 가상 기능 등록 정보, 93
- 슬라이스 2
 - 내보내기, 172
- 시간 초과 옵션
 - 가상 디스크, 166
- 시스템 컨트롤러 살펴볼 내용 서비스 프로세서(SP)
 - 시스템에서 읽을 수 있는 출력
 - 리소스 나열, 327
- 시작
 - 게스트 도메인, 43
- 식별
 - InfiniBand 기능, 113
- 신뢰할 수 있는 가상 네트워크, 271
- 실패 정책
 - 도메인 종속성에 대해 구성, 357
-
- 액세스
 - 게스트 도메인에서 광 섬유 채널 가상 기능, 127
- 업데이트
 - /etc/system 파일, 352
 - PVLAN, 258
- 역할
 - 도메인, 22
 - 만들기, 29
 - 사용자에게 지정, 29
 - 지정, 27
- 역호환성
 - 볼륨 내보내기, 175
- 연결
 - 네트워크를 통해 게스트 콘솔에, 59
- 오류
 - CPU 및 메모리 주소 매핑을 통해 문제 해결, 360
- 요구사항
 - InfiniBand SR-IOV, 103
 - PVLAN, 257
 - SR-IOV, 76
 - 가상 네트워크 성능 최대화, 222
 - 광 섬유 채널 가상 기능, 115, 116
 - 동적 SR-IOV, 81, 81
 - 리소스 그룹, 323
 - 이더넷 SR-IOV, 85
 - 정적 SR-IOV, 80
 - 직접 I/O, 137
- 이더넷 SR-IOV
 - 계획, 86
 - 네트워크 구성, 97
 - 요구사항, 85
 - 장치 관련 등록 정보, 86, 98
 - 제한 사항, 85
- 이더넷 SR-IOV 가상 기능을 사용하여 I/O 도메인 부트, 97
- 인터럽트 제한
 - 조정, 369
- 일관성 링크 조정, 378
- ㅈ
- 자동 저장 구성 디렉토리
 - 복원, 373
 - 저장, 373
- 장치 관련 등록 정보
 - 이더넷 SR-IOV, 98
- 재바인드
 - CPU 전체 코어로 시스템, 311
- 재부트
 - CPU 전체 코어로 시스템, 311
 - 루트 도메인, 134, 140
 - 컨트롤 도메인, 37
- 저장
 - Logical Domains 제약 조건 데이터베이스, 374
 - ZFS 볼륨을 사용하여 디스크 이미지, 187
 - ZFS 파일을 사용하여 디스크 이미지, 187
 - ZFS를 사용하여 디스크 이미지, 186
 - 도메인 구성, 338, 340

- 자동 저장 구성 디렉토리, 373
 - 적용
 - 전체 코어 제약 조건, 302
 - 최대 코어 제약 조건, 303
 - 전원 관리(PM)
 - CPU, 311
 - 관찰성 모듈
 - 사용으로 설정, 380
 - 기능, 378
 - 사용, 323, 377
 - 전원 껐다 켜기
 - 서버에서 수행, 354
 - 전원 소비 데이터
 - 보기, 379
 - 전원 제한, 378
 - 전체 코어 제약 조건
 - 적용, 302
 - 정보 프레임
 - Oracle Solaris 10 vnet 및 vsw 드라이버의 정보 미
인식 버전과의 호환성, 267
 - 구성, 264
 - 제거, 96
 - 살펴볼 다른 내용 삭제
 - I/O 도메인에서 InfiniBand 가상 기능, 109
 - I/O 도메인에서 광 섬유 채널 가상 기능, 126
 - I/O 도메인에서 이더넷 가상 기능, 96
 - physical-bindings 제약 조건, 313
 - PVLAN, 258
 - 가상 디스크, 166
 - 도메인에서 메모리, 316
 - 루트 도메인에서 InfiniBand 가상 기능, 111
 - 모든 게스트 도메인, 374
 - 모든 도메인 구성, 375
 - 물리적으로 바인드되지 않은 리소스, 314
 - 제어
 - 복구 모드, 350
 - 제한
 - 물리적 네트워크 대역폭, 232
 - 제한 사항
 - PVLAN, 257
 - SR-IOV, 79
 - 광 섬유 채널 가상 기능, 116
 - 리소스 그룹, 323
 - 비_{primary} 루트 도메인, 153
 - 이더넷 SR-IOV, 85
 - 직접 I/O, 138
 - 조정
 - 인터럽트 제한, 369
 - 종속성 주기, 358
 - 줄이기
 - 컨트롤 도메인 CPU 및 메모리 리소스, 36
 - 컨트롤 도메인의 메모리, 317
 - 중단
 - Oracle Solaris OS, 354
 - 중지
 - 부하가 높은 도메인, 353
 - 지연된 재구성, 300, 317
 - 지정
 - I/O 도메인에 끝점 장치, 135
 - MAC 주소, 237
 - 수동으로, 237
 - 자동으로, 237
 - Oracle Solaris 10 게스트 도메인에서 VLAN, 255
 - Oracle Solaris 10 서비스 도메인에서 VLAN, 254
 - Oracle Solaris 11 게스트 도메인에서 VLAN, 254
 - Oracle Solaris 11 서비스 도메인에서 VLAN, 253
 - PCIe 끝점 장치, 68
 - VLAN, 253
 - 권한 프로파일, 27, 28
 - 도메인에 물리적 리소스, 311
 - 루트 도메인에 PCIe 버스, 65
 - 마스터 도메인, 357
 - 사용자에게 역할, 29
 - 역할, 27
 - 지침
 - I/O 도메인 만들기, 64
 - 파일 및 볼륨을 가상 디스크로 내보내기, 176
 - 직접 I/O(DIO)
 - 문제, 148
 - 비_{primary} 루트 도메인에서 장치 관리, 156
 - 요구사항, 137
 - 제한 사항, 138
- 大**
- 차단 목록
 - FMA(Fault Management Architecture), 345
 - 결함이 있는 하드웨어 리소스, 345
 - 최대 코어 제약 조건
 - 적용, 303

- 최대화
가상 네트워크 성능, 222, 223
- 추가
I/O 도메인에 InfiniBand 가상 기능, 107
I/O 도메인에 광 섬유 채널 가상 기능, 125
I/O 도메인에 이더넷 가상 기능, 95
가상 디스크, 165
도메인에 메모리, 316
루트 도메인에 InfiniBand 가상 기능, 110
정렬되지 않은 메모리, 319
- 출하 시 기본 구성
복원, 375
서비스 프로세서에서 복원, 376
- ㅋ**
컨트롤 도메인, 22
구성, 35
메모리 재구성, 317
메모리 줄이기, 317
재부트, 37, 355
컨트롤 도메인 CPU 및 메모리 리소스
줄이기, 36
컨트롤 도메인에서 기본 서비스
만들기, 34
- 콘솔
단일 그룹으로 결합, 60
로깅, 59
- 콘솔 그룹
사용, 60
- ㅍ**
포트 VLAN ID(PID), 252
- 프로비전
복제본을 사용하여 도메인, 188
- 프로세서 전원 소비 데이터
보기, 381
- ㅎ**
하드 분할
시스템 구성, 305
- 하드웨어 오류
문제 해결, 345
- 하이퍼바이저
Logical Domains Manager 및, 20
정의, 20
- 할당
CPU 리소스, 302
광 섬유 채널 가상 기능에 대한 WWN(world-wide name), 117
리소스, 301
- 확인
ILOM 상호 연결 구성, 40
가상 SCSI HBA의 존재 여부, 203
네트워크 장치의 GLDv3 준수(Oracle Solaris 10), 239
도메인 구성, 306
도메인 마이그레이션 상태, 298
- 확인된 부트
사용, 31
- 활용률 통계, 328

