가상 머신 풀을 이용한 가상 머신 Provisioning 연구

이지형° 고광원 우영춘 배승조

과학기술연합대학원° 한국전자통신연구원

neomark4@gmail.com° {kwangwon.koh,ycwoo,sbae}@etri.re.kr

A Study of Fast Virtual Machine Provisioning using VMOSPOOL

JIHYOUNG LEE° KWANGWON KOH YOUNGCHOON WOO and SEOUNGJO BAE

Korea University of Science and Technology °

Electronics and Telecommunications Research Institute

요 약

가상화는 요즘 각광받는 기술 중 하나이다. 가상화의 개념이 처음 소개된 것은 20년 전의 일이다. 최근에 가상화가 다시 주목 받는 이유는 인터넷 사용자의 증가로 인해 서버의 수가 급증하였고 그에 반해 서버들의 활용률은 20~30%에 그치기 때문이다. 가상화를 채택하는 분야 중 하나는 바로 인터넷 데이터 센터(Internet Data Center, IDC)이다. IDC에서는 하나의 고성능 서버 위에 여러 개의 가상 머신을 구동함으로써 서버가 차지하는 공간을 줄이고 관리 비용을 절감하는 서버통합(server consolidation)에 주로 사용된다.

가상화를 통해 서비스를 제공하기 위한 첫 번째 단계는 가상 머신을 생성하는 것이다. 일반적으로 가상 머신의 생성은 물리적 노드 (비 가상 머신)에 운영체제를 설치하는 것과 동일하다. 본 논문에서는 서비스 제공을 위해 선행되어야 할 가상 머신을 생성함에 있어 가상 머신 풀(Virtual Machine OS Pool, VMOSPOOL)을 사용하여 빠르게 동적으로 가상 머신을 생성하는 방법에 대해 논의한다. 특히 가상 머신 풀의 사용은 고가의 공용 스토리지가 없는 상황에서 부하 분산 클러스터를 구축하는데 유용함을 보인다.

1. 서 론

가상화란 하나의 물리적 서버 위에 여러 개의 운영체제를 구동할 수 있는 기술을 말한다. 각각의 운영체제는 가상 머신이란 이름으로 불린다. 가상화는 각 가상 머신간 실행 환경의 분리, 서버의 활용률 증가, 편리한 가상 머신의 리소스 관리 그리고 가상 머신의 오류와 상관없는 안정성과 같은 장점을 갖는다[1,2]. 이와 같은 장점 때문에 많은 기업 환경에서 가상화를 채택하고 있다. 특히 저가의 컴퓨터를 사용하여 클러스터를 구축하고 있는 인터넷 데이터센터(Internet Data Center, IDC)나 각종 포털 업체들이 많은 관심을 보이고 있다. 그런 업체들은 낮은 성능을 내는 수많은 컴퓨터들을 고성능 서버 위에서 동작하는 가상 머신으로 바꾸고자 한다. 이와 같은 작업을 서버 통합(consolidation)이라 한다[3,4].

서비스를 제공하기 위한 제일 처음 단계는 가상 머신의 설치이다. 기본적으로 가상 머신의 설치 절차는 물리 노드 (비 가상 머신)의 설치와 유사하다. 가상 머신은 자신의 디스크로 블록 디바이스, 파일, 로지컬 볼륨(LVM) 등을 가질 수 있다. 따라서 가상 머신을 생성할 때 특정 파일을 복사하거나 블록 장치를 덤프(dump)하여 가상 머신 이미지(가상 머신이 사용하게 될 디스크에 포함될 운영체제 및 프로그램)를 만들 수 있다.

가상화 환경에서 서비스를 하는 동안 새로운 가상 머신을 만들어야 할 필요가 있다. 특히 부하 분산용 클러스터에서 새로 생성되는 가상 머신은 기존의 가상 머신들의 역할과 동일하게 된다. 하지만 새로운 가상 머신을 생성하는 것은 최소한 이미지를 복사하는 긴 시간이 걸리는 작업이다. 따라서 일반적으로 여분의 가상 머신을 미리 생성해 놓는 방법이 사용된다. 하지만 여분의 이미지는 시스템 자원을 소모하는 낭비가 되게 된다. 만일 가상 머신 이미지를 동적으로 생성할 수 있다면 자원을 절약하고 시스템의 유연성을 키울 수 있게 된다.

본 고에서는 디스크 공간을 절약하고 가상 머신의 설치 시간을 단축할 수 있는 유연한 가상 머신 프로비저닝 기술에 대해서 논의한다.

다음 장에서는 가상 머신의 동적 프로비저닝이 필요한 이유에 대해 알아본다.

2. 동 기

가상화에서 중요한 것은 실질적인 사용이다. 가상화 기술을 통해 물리적 서버 공간의 감소와 관리의 용이함을 얻을 수 있었지만 이것은 가상 머신의 국한된 특징들이다. 최종적인 목표는 가상화를 통해 유연한 서비스를 제공하는 것이다. 그러기 위해 SLA(Service Level Agreement)를 만족하기 위한 방법을 지속적으로 찾아야 한다.

서비스를 제공하는 동안에 새로운 가상 머신을 생성하거나 없애는 것은 매우 빈번히 발생하는 일이다. 가상 머신의 제거는 간단하며 시간 또한 오래 걸리지

않는다. 워크로드를 분산하기 위해 새로운 가상 머신을 생성하려 할 때, 이것은 시간에 민감한 작업이 될 것이고 최소한 빠른 수행이 시스템 전체적으로 도움이 되게 된다. 이런 상황을 대비해서 여분의 가상 머신을 준비하게 되는데 이런 방식은 디스크 공간의 낭비일 뿐 아니라 미리 생성해 놓은 가상 머신 이미지의 위치에 따라 가상 머신을 생성할 수 있는 서버가 제한되는 문제점이 발생하게 된다. 가상 머신 이미지의 위치에 따른 문제점을 해결할 수 있는 좋은 방법은 SAN이나 NAS같은 공용 스토리지를 사용하는 것이다. 하지만 여전히 SAN은 고가의 장비이다. 서비스를 제공하는 기업 환경을 감안하면 충분히 감내할 수 있는 부분이지만 적은 비용으로 충분한 효과를 낼 수 있다면 그것이 더 좋은 솔루션이 될 수 있다.

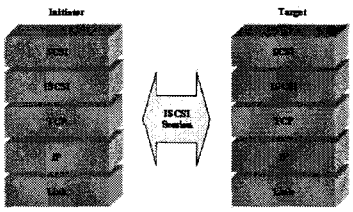
이미지에 국한되지 않고 동적으로 새로운 가상 머신을 생성할 수 있게 된다면 서비스와 가상 머신을 프로비저닝 하는데 있어 좀 더 유연성을 갖게 된다. 다음 장에서는 제안하는 시스템의 전체적인 구조에 대해 알아본다.

3. Virtual Machine OS Pool (VMOSPOOL)

해결해야 할 문제는 두 가지이다. 하나는 적은 비용으로 공용 스토리지를 구축하는 것이고 다른 하나는 동적으로 가상 머신 이미지를 만들어 내는 것이다.

공용 스토리지를 만들기 위해 iSCSI 프로토콜을 사용한다. iSCSI는 네트워크로 연결되는 SCSI 인터페이스이다[5]. iSCSI와 비슷한 역할을 하는 것으로 NBD(Network Block Device), NFS(Network File System), 등이 있으나 iSCSI를 선택한 이유는 iSCSI가 SAN과 비교될 수 있을 만큼 성능이 좋기 때문이다[6].

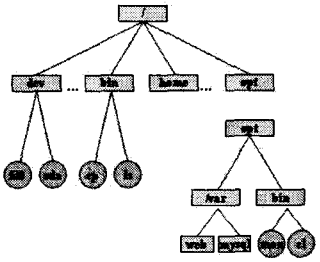
iSCSI 프로토콜은 데이터 전송을 위해 TCP/IP를 사용한다. Fibre 채널을 사용하는 SAN과 달리 Ethernet 과 같이 TCP/IP가 가능한 네트워크 인터페이스를 사용하기 때문에 Fibre 채널 같은 고가의 장비에 소요되는 비용을 줄일 수 있게 된다.



[그림 1] iSCSI 프로토콜 스택. iSCSI는 TCP/IP 위에서 동작한다.

iSCSI는 사용하기 쉽다. 공유하기 원하는 블록 장치를 설정 파일에 명시하고 대몬 프로세스를 시작하면 공유 스토리지를 사용하기 원하는 서버들이 'initiator'라 불리는 클라이언트 프로그램을 통해서 임포트(import)할 수 있다. 일단 임포트된 블록 장치는 로컬 디스크처럼 자유롭게 사용이 가능하다.

iSCSI를 통해 공유 스토리지 문제를 해결한다면 동적 프로비저닝은 리눅스 시스템의 파일 시스템 마운트(mount) 옵션과 XEN 하이퍼바이저(hypervisor)의 읽기 전용 속성을 사용하여 해결한다. 리눅스 시스템에서 특정 디렉토리는 중복되서 마운트될 수 있다. 예를 들면, '/opt' 라는 디렉터리에 '/dev/sdb'라는 블록 장치를 마운트 시키면 '/opt' 디렉터리는 '/dev/sdb' 블록 장치에 있는 내용을 가리키게 된다. 똑같은 디렉터리에 다른 블록 장치를 마운트하면 마지막으로 마운트된 장치의 내용을 가리키게 된다. 이것은 리눅스에서 파일 시스템 마운트의 속성이다[7, 8].

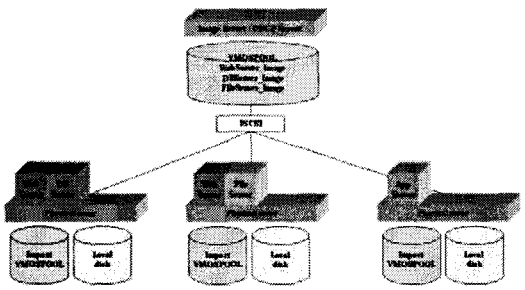


[그림 2] 리눅스 디렉터리 마운트. 리눅스는 중복된 파일 시스템의 마운트를 허용한다.

다음으로, XEN 하이퍼바이저는 가상 머신을 생성할 때 가상 머신용 이미지를 읽기 전용으로 사용할지, 쓰기도 가능한 형태로 사용할지 선택할 수 있다. 가상 머신 이미지를 어떻게 사용할지는 설정 파일에 옵션을 통해서 조정이 가능하다. 물론 읽기 전용으로 가상 머신을 생성한다면 가상 머신을 사용하는 동안 파일 시스템에 쓰는 작업을 할 수 없다. 이것은 리눅스의 마운트 속성으로 해결한다. 즉, 실행 파일은 읽기 전용 모드로 두고 데이터가 저장되는 디렉터리는 쓰기가 가능한 로컬 시스템을 NFS로 마운트해서 가상 머신을 사용하게 된다. 이렇게 되면 쓰기가 가능해지므로 여타의 가상 머신들과 차이가 없어진다. 더불어 읽기 전용으로 가상 머신 이미지를 다루기 때문에 하나의 이미지를 가지고 여러 개의 가상 머신을 생성할 수 있고, 동시 접근으로 인해 이미지 내의 파일 시스템이 깨지는 문제도 막을 수 있다.

[그림 3]은 VMOSPOOL의 전체적인 모습을 보여주고 있다. iSCSI를 통해 생성한 공유 스토리지에 미리 가상 머신용 이미지를 생성해 놓는다. 이 공유 스토리지는 가상 머신 OS풀(Virtual Machine OS Pool, VMOSPOOL)이 된다. 이 VMOSPOOL을 사용하여 가상 머신을 생성하고자 하는 물리 노드들은 VMOSPOOL을 iSCSI initiator로 임포트하고 그 안에 들어있는 가상 머신

이미지를 읽기 전용 모드로하여 새로운 가상 머신을 생성하게 된다. 그 후에 데이터 디렉터리로 사용될 로컬 파일 시스템의 디렉터리를 NFS로 마운트하면 하나의 이미지를 사용하여 여러 개의 가상 머신을 생성해 낼 수 있게 된다.



[그림 3] Virtual Machine OS Pool(VMOSPOOL) 의 구조. iSCSI를 사용하여 공유 스토리지를 임포트하고 리눅스의 중복 마운트 속성을 사용한다.

이와 같은 방식으로 빠르고 동적인 가상 머신 프로비저닝을 적은 비용으로 가능하게 한다. 다음 장에서는 VMOSPOOL을 사용하여 부하 분산 웹 클러스터 서버를 구성해본다.

4. 구현 및 평가

본 장에서는 부하 분산 웹 서버 클러스터를 구축해 봄으로써 VMOSPOOL 모델을 적용한 시스템을 평가해 본다.

4.1 부하 분산 웹 클러스터

부하 분산 웹 서버 클러스터를 구축하는 첫 단계는 웹 서버용 가상 머신 이미지를 만드는 것이다. 이 가상 머신용 이미지에는 'apache' 웹 서버와 부하 분산 클러스터를 구축할 수 있는 LVS(Linux Virtual Server) 관련 패키지들을 포함시킨다. 부하 분산을 담당하는 로드 밸런서(load balancer)용 이미지를 따로 생성하지 않고 동일한 이미지를 사용하기 위해서이다. LVS 패키지에는 하트비트 모니터링 툴이나 IP 기반의 터널링을 할 수 있는 툴들이 포함된다[9, 10]

XEN 가상 머신의 디스크는 블록 디바이스, LVM의 로지컬 볼륨 그리고 파일이 될 수 있다. 가상 머신용 이미지는 XEN 하이퍼바이저 패키지에 포함되어 있는 'xenguest-install.py'라는 도구를 사용하여 생성한다.

[표 1] iSCSI 설정 파일과 서비스 시작 명령

```
'/etc/ietd.conf'
...
Target iqn.node03:node03
    Lun 0 Path=/dev/sdb1, Type=fileio
...

#service iscsi-target start
```

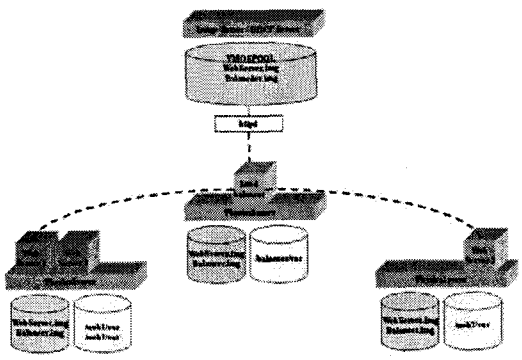
이미지를 생성한 후 iSCSI로 공유 스토리지가 될 블록 디바이스를 익스포트(export) 시킨다. <표 1>은 실제 익스포트할 블록 장치를 명시하고 대몬 프로세스를 띄우는 명령어를 나타낸 것이다.

'/dev/sdb1' 디스크를 파티셔닝(partitioning)한 후 ext3 파일 시스템을 설치한다. 그리고 거기에 'Web-Server.img'라는 가상 머신용 이미지를 생성하였다. 가상 머신은 DHCP로 네트워크 설정을 한다. 가상 머신이 부팅되면 'webmount.sh'라는 스크립트를 자동으로 실행하게 된다.

[표 2] webmount.sh 의 pseudo 코드

```
...
MAC=
if "$MAC" == "mac of WebServer1"
    mount localserver:/web1/var /var
if "$MAC" == "mac of WebServer 2"
    mount localserver:/web2/var /var
...
```

스크립트는 가상 머신의 MAC 주소에 따라 가상 머신을 생성한 물리 노드의 특정 디렉터리를 가상 머신에서 구동되는 웹 서버의 데이터 디렉터리로 마운트하게 된다. 가상 머신을 생성할 때 설정 파일에 임의의 MAC 주소를 선택할 수 있으므로 이와 같은 방식으로 DHCP의 IP주소와 데이터 디렉터리를 구분할 수 있게 된다.



[그림 4] 부하 분산 웹 클러스터. 하나의 로드 밸런서와 두 개의 실제 가상 웹 서버로 구성된다.

'/var/www/html'은 아파치 웹 서버의 기본 데이터

디렉터리이다. 실제 부하 분산용으로 사용하기 위해서는 동일한 데이터를 서비스 해야 하지만 본 논문에서는 동작을 테스트하기 위함이므로 서로 다른 내용으로 데이터를 수정한다. 복잡한 데이터보다는 단순히 'Web-ServerX' 의 메시지와 서로 다른 그림을 출력하도록 하였다.

로드 밸런서의 설정은 다이렉트 라우팅(Direct Routing, DR) 방식을 이용하였다. 다이렉트 라우팅 방식은 가상 IP를 이용하여 처음 서비스 요청은 로드 밸런서에 의해 실제 서비스 서버가 선택되지만 그 이후에는 리얼 서버와 클라이언트가 직접 통신하는 방식이 된다[11]. 그리고 부하 분산의 방법은 라운드 로빈 방식으로 요청이 고르게 분배되도록 하였다.

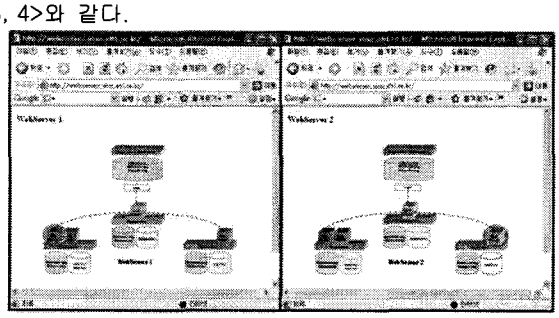
[표 3] 로드 밸런서 서버의 LVS/DR 설정

```
/etc/sysctl.conf
...
net.ipv4.ip_forward=1
...
# sysctl -p
# ifconfig eth0:0 $VIP up
# ipvsadm -A -T $VIP:80 -s rr
# ipvsadm -a -t $VIP:80 -r $REAL_SERVER1 -g
# ipvsadm -a -t $VIP:80 -r $REAL_SERVER2 -g
```

[표 4] 가상 머신의 LVS 설정

```
/etc/sysctl.conf
...
net.ipv4.ip_forward = 1
net.ipv4.conf.lo.arp_ignore = 1
net.ipv4.conf.lo.arp_announce = 2
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.all.arp_annouce = 2
...
#sysctl -p
#ifconfig eth0:0 $VIP up
```

로드 밸런서의 설정과 각각 가상 머신의 설정은 <표 3, 4>와 같다.



[그림 5] 클라이언트의 웹 서비스 요청

<그림 5>는 클라이언트 요청에 따른 부하 분산 웹 서버의 응답을 보여주고 있다. 동일한 주소의 요청에 대해 서로 다른 그림을 보여주는 것으로 보아 부하 분산이 이루어 지고 있음을 알 수 있다.

4.2 평가

VMOSPOOL은 동적인 가상 머신 프로비저닝을 주 목적으로 한다. 경제성, 공간, 시간, 관리 그리고 안정성을 중심으로 본 모델을 평가해 보았다.

[표 5] 기존의 방식과의 비교

	VMOSPOOL	legacy
economy	cheap	expensive
space	tiny	big
time	fast	slow
management	easy	difficult
stability	high	medium

경제성은 공유 스토리지와 관계된다. SAN과 같은 고가의 장비를 구매하게 된다면 시스템 구축 비용은 올라가게 된다. 대조적으로 본 모델은 iSCSI 프로토콜을 이용하여 저렴하게 시스템을 구축할 수 있다.

공간은 가상 머신 이미지의 크기를 말한다. 만일 모든 가상 머신이 각각의 디스크 이미지를 갖는다고 한다면 하나의 이미지 크기를 4GB라고 할 때 클러스터를 위해 최소 12GB이상의 공간이 필요하게 된다. VMOSPOOL에서는 4GB 크기의 이미지 하나와 로컬 디스크에서 데이터용으로 사용할 공간만 있으면 충분하다.

시간은 가상 머신 이미지의 생성 시간이다. 이것은 동적 프로비저닝의 가장 큰 걸림돌이다. 미리 하나의 가상 머신을 만들어 놓는 것 외에 딱히 다른 방법이 없다. 가상 머신 풀에서는 저렴한 공유 스토리지와 가상 머신 이미지 재활용을 통해 이 문제를 해결한다.

관리는 가상 머신 이미지의 업데이트와 패치 같은 작업을 말한다. 클러스터에 사용될 수만큼의 이미지가 따로 존재한다면 개별적으로 모든 가상 머신 이미지를 관리해줘야 한다. 반대로 VMOSPOOL에서는 하나의 이미지만을 관리하면 되므로 편리하다.

안정성은 가상 머신 이미지의 속성을 말한다. 일반적인 시스템에서 중요한 실행 파일이나 설정파일이 손상되면 그 시스템은 사용하기 어려운 상태가 된다. 하지만 VMOSPOOL은 가상 머신 이미지를 읽기 전용으로 사용하므로 파일이 손상될 위험이 적어진다.

위와 같은 장점 이외에 실행 파일과 데이터를 분리함으로써 가상 머신을 마치 어플리케이션처럼 다룰 수 있다는 것도 시스템을 구축하고 유지하는데 유연성을 실어주게 된다.

5. 결론 및 향후 연구 과제

지금까지 동적인 가상 머신을 효율적으로 할 수 있는 VMOSPOOL에 대해 알아보았다. 그리고 가상 머신 풀을 적용한 부하 분산 웹 클러스터를 구현하고 평가해 보았다.

5.1 향후 연구 과제

가상 머신 풀을 연구하는 동안 스토리지와 라이브 마이그레이션이라(Live Migration)는 기능이 유용함을 알 수 있었다.

스토리지의 공유는 가상화 기술에서 무척 까다로운 문제이다. 특히 라이브 마이그레이션과 같은 유용한 기능은 스토리지가 공유되지 않으면 불가능에 가까웠다. 이 기능을 지원하기 위해서는 VMOSPOOL에서는 데이터 디렉터리가 변수가 된다. 데이터 디렉터리의 동기화가 이루어 진다면 쉽게 기능을 이용할 수 있게 된다. 그리고 더 빠른 가상 머신 프로비저닝을 위해 'diskless' 형태의 가상 머신 프로비저닝 기술도 병행하여 진행할 것이다.

5.2 결 론

지금까지 빠르고 쉽고 저렴하게 가상 머신을 프로비저닝할 수 있는 가상 머신 풀에 대해 알아보았다. 이런 접근 방식을 통해 서비스를 좀 더 빠르게 제공할 수 있었다.

VMOSPOOL은 iSCSI 프로토콜을 이용하여 공유 스토리지를 생성한다. 그 결과 고가의 공유 스토리지 장비를 사는데 소요되는 비용을 줄일 수 있다. 그리고 XEN 하이퍼바이저의 가상 머신 이미지 읽기 전용 속성과 리눅스 마운트 명령어의 중복 기능을 사용한다. 이것은 빠르고 시간, 공간, 관리 그리고 안정성 측면에서 많은 이점을 주게 된다. 4.2장을 통해 이러한 특징에 대해 설명하였다. 그리고 라이브 마이그레이션과 diskless 형태의 가상 머신 프로비저닝 기술 연구가 진행될 것이다.

가상화 환경에서 워크로드 관리는 매우 중요한 이슈이다. 전체 시스템의 유연한 동작을 위해 동적인 가상 머신 프로비저닝은 필수 요소가 된다. 이를 위해 가상 머신 풀(VMOSPOOL) 모델을 제안하고 부하 분산 웹 클러스터의 프로토 타입을 구축하여 보았다. 가상 머신 풀 모델은 저렴하면서도 뛰어난 성능을 보이는 프로비저닝을 위한 좋은 모델일 될 수 있을 것이다.

참 고 문 헌

[1] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt and Andrew Warfield, "Xen and the art of virtualization", ACM SIGOPS Operating Systems Review, Volume 37, Issue 5, pp. 164- 177, Dec. 2003.

[2] 김진미, 배승조, 정영우, 심규호, 고광원, 우영춘, "유틸리티 컴퓨팅 시대를 여는 가상화 기술 동향", 주간기술동향, 통권 1208 호, 2005.10.

[3] J. Reumann, A. Mehra, K. G. Shin and D. Kandlur, "Virtual Services: A New Abstraction for Server Consolidation", In USENIX Annual Technical Conference, pp. 117-130, Jun. 2000

[4] 안창원, 김진미, "데이터 센터 통합을 위한 가상화 기술 동향", 주간기술동향, 통권 1287 호, 2007.2

[5] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, E. Zeidner, "iSCSI RFC", http://www.ietf.org/rfc/rfc3720.txt, Apr 2004.

[6] Yingping Lu, Du. D.H.C, "Performance study of iSCSI-based storage subsystems", Communications Magazine, IEEE, volume 41, Issue 8, pp. 76-82, Aug. 2003

[7] Daniel P. Bovet, Marco Cesati, "Understanding the LINUX KERNEL", Third Edition, O'REILLY, Nov. 2005.

[8] Xen User's Manual Xen V3.0, http://www.cl.cam.ac.uk/research/srg/netos/xen/readmes/user/user.html

[9] Wensong Zhang, Wenzhuo Zhang, "Linux Virtual Server Clusters: Build highly-scalable and highly-available network services at low cost", Linux Magazine, Nov. 2003

[10] Karl Kopper, "The Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software", NO STARCH PRESS, May. 2005.

[11] Virtual Server Via Direct Routing http://www.linuxvirtualserver.org/VS-Routing.html