Kvm 虚拟机实验搭建过程(kvm不同于xen 在cpu和内存虚拟化方面使用cpu硬件虚拟化技术，需要cpu的支持，如果不支持只能采用完全虚拟化了)。

一、虚拟化组件安装

yum groupinstall Virtual\*

加载kvm模块

# modprobe -l |grep kvm

kernel/arch/x86/kvm/kvm.ko

kernel/arch/x86/kvm/kvm-intel.ko

kernel/arch/x86/kvm/kvm-amd.ko

Inter加载模块

Modprobe kvm

Modprobe kvm-intel

创建虚拟机并以光盘启动：

virt-install -n "server" -r 1024 --disk path=/home/servier.img,size=20,format=raw,cache=writeback,sparse --force -w bridge=virbr0,model=virtio --graphics vnc,port=5900 --boot cdrom,hd --cdrom=/home/CentOS-6.5-x86\_64-bin-DVD1.iso

描述：基于cdrom安装

镜像文件此处使用raw

开启稀疏文件格式

网卡采用半虚拟化virtio

如果需要使用libvirt进行管理的话，无法使用虚拟化交换机

导出已生成的server.xml作为以后虚拟机的基版

virsh dumpxml server >/etc/libvirt/qemu/server2.xml

修改的地方有：

Server name

Uuid 此处为基于

img名字

Vnc端口号（）

mac地址

virsh define /etc/libvirt/qemu/server2.xml #使用虚拟描述文档建立虚拟机

Virsh start server2

#################################################################

因为涉及到迁移（如果需要的话），需要用到san存储。

Qemu-kv 启动脚本

#!/bin/bash

qemu-kvm -m 512 -name "gust1" -drive file=./gust1.img,if=virtio,media=disk,cache=writeback,format=raw -smp sockets=1,cores=2,threads=2 -vnc :0 -daemonize -net nic,macaddr=52:54:00:00:00:02,model=virtio -net tap,ifname=dom2,script=/etc/qemu-ifup,downscript=/etc/qemu-ifdown

Qemu-ifup 启动时自动加入

启动脚本：

#!/bin/bash

bridge=br0

if [ -n "$1" ];then

ip link set $1 up

sleep 1

#brctl addif $bridge $1

ovs-vsctl add-port $bridge $1

[ $? -eq 0 ] && exit 0 ||exit 1

else

echo "Error :no interface"

fi

关机移除脚本：

#!/bin/bash

bridge=br0

if [ -n "$1" ];then

ovs-vsctl del-port $bridge $1

ip link set $1 down

exit 0

else

echo "Error: no interface specified."

exit 1

fi