

知 在linux环境下设置双网卡绑定

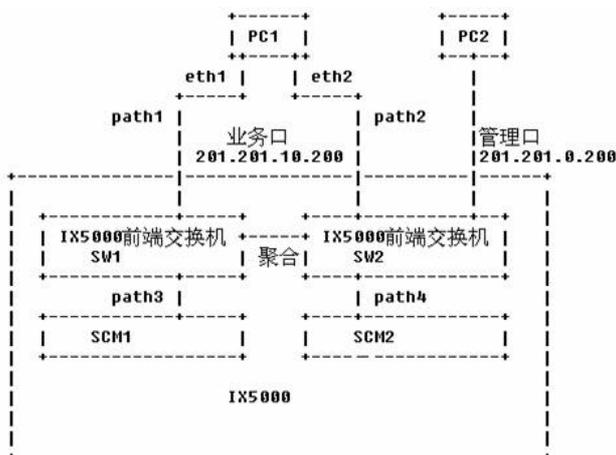
其它功能 王延峰 2006-07-03 发表

Linux as3下使用bonding实现多路径

实验目的:

将多块网卡虚拟成一块网卡,使其具有相同的IP地址,来实现提升主机的网络吞吐量或者是提高可用性,这种技术在Linux中的实现被称作bonding, Linux AS3的最小安装自带了bonding,结合Linux iSCSI Initiator的版本是3.6.2就可以为linux主机业务端口实现多路径冗余。

组网图:



组网图如上图所示, PC1表示安装Linux AS3的测试用机,安装了两块网卡(假设其接口分别为eth1和eth2),分别连接到IX5000的2个前端交换机(两个交换机使用链路聚合连接),而IX5000的两个SCM分别连接到这两个交换机。PC2是管理终端,连接到IX5000的管理口。path1、path2分别代表PC1和SW1、SW2之间的链路,path3、path4分别代表两台SW和两台SCM之间的链路。

实验过程:

1. 分别禁用eth1和eth2,使用“ping 201.201.10.200”验证网络是否可达;
2. 通过检查系统是否包含“ifenslave”命令,确定系统中是否支持bonding,若没有,安装补丁;
3. 配置系统,使得系统重启之后Bonding依然生效;

首先需要在/etc/modules.conf(针对Linux AS3)中添加如下内容:

```
vi /etc/modules.conf
```

```
alias bond0 bonding
```

```
options bond0 mode=1 miimon=100 downdelay=200 updelay=200
```

然后在/etc/sysconfig/network-scripts目录中添加ifcfg-bond0文件(假设bond0的IP是192.168.1.1):

```
vi /etc/sysconfig/network-scripts/ifcfg-bond0
```

```
DEVICE=bond0
```

```
IPADDR=192.168.1.1
```

```
NETMASK=255.255.255.0
```

```
BROADCAST=192.168.1.255
```

```
OnBOOT=yes
```

```
BOOTPROTO=none
```

```
USERCTL=no
```

注:编辑此文件时需要注意,环境变量名字一定要大写

所有属于bond0的接口必须被定义为SLAVE或MASTER。例如,假设希望定义eth0和eth1为属于接口bond0的一部分,那么它们的配置文件(ifcfg-eth0, ifcfg-eth1)应修改为:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth1
```

```
USERCTL=no
```

```
OnBOOT=yes
```

```
MASTER=bond0
```

```
SLAVE=yes
```

```
BOOTPROTO=none
```

注:编辑此文件时需要注意,环境变量名字一定要大写

```
vi /etc/sysconfig/network-scripts/ifcfg-bond0
```

```
DEVICE=eth1
```

```
USERCTL=no
```

```
OnBOOT=yes
```

```
MASTER=bond0
```

SLAVE=yes

BOOTPROTO=none

注：编辑此文件时需要注意，环境变量名字一定要大写

4. 使用“service network restart”以及“ifconfig bond0”命令检查接口bond0是否正常启用；
5. 使用“ping 201.201.10.200”验证网络是否可达；
6. IX5000为PC1分配卷vol1；
7. PC1运行iSCSI Initiator连接到IX5000，在vol1上创建文件系统，并mount到系统中（比如/mnt/test）
此处操作请参看韩立君编辑的linux环境下软件initiator的安装配置使用！！
8. copy.sh是一个测试脚本，假设位于/home/test/目录下，并且该目录下有一个文件file1。
copy.sh的原理就是在一个while循环中拷贝/删除文件，一直保持对卷的I/O操作。

路径切换试验：

1. 在PC1上运行“copy.sh file1 /home/test /mnt/test”；
2. 断开path1，copy.sh脚本的运行中断很短的时间，继续运行
3. 恢复path1，等待2分钟，断开path2，copy.sh脚本的运行中断很短的时间，继续运行
4. 恢复path2，等待2分钟，断开path1，copy.sh脚本的运行中断很短的时间，继续运行
5. 恢复path1，等待2分钟，断开path2，copy.sh脚本的运行中断很短的时间，继续运行