Gigabit

Table des matières

1 Introduction

Il s'agit d'utiliser un des deux slot PCI présent sur les cartes compact-PCI.

- 192.168.8.1 Carte controleur
- 192.168.8.2 Pizza Box
- 192.168.8.3 Pc Nec

2 Modules

2.1 Carte controleur

La carte mise à disposition est une carte Intel(R) PRO/1000 Gigabit et son driver est le module *e1000*.

- Carte muette.
  Bien que le module **e1000** se charge et que la carte se configure normalement, la carte n'émet aucune requête ARP et semble ne pas recevoir de trames non plus. Le symptôme est qu'il manque le flag `RUNNING` dans le compte rendu de **ifconfig**. En ajoutant la définition de la variable DBG (`#define DBG`) au début du fichier */usr/src/linux/drivers/net/e1000/e1000_osdep.h* on constate que le lien ne peut être établi (`Unable to establish link!!!`). Pour contourner cette erreur, nous avons flashé l'EEPROM de la carte GIGABIT :

```
# modprobe e1000
# ~/ethtool-6/ethtool -E eth1 magic 0x10088086 offset 31 value 0xf2
```

L'outils ETHOOL utilise en fait le driver de la carte éthernet. Le module utilise le numéro `magic` pour vérifier l'identité de la carte à modifier (cf *e1000/e1000_ethtool.c*). Ce numéro correspond aux numéros identifiants du vendeur et de la carte.

```
# lspci -n
00:02.0 Class 0200: 8086:1008 (rev 02)
```

- Boot.
  Si l'on compile le driver en dur dans le noyau, alors au boot le noyau n'arrive pas à monter la partition racine via NFS. On dirait que le noyau inverse les interfaces **eth0** et **eth1** au cours du processus de bout. Etrangement, que l'on boot sur l'une interface ou l'autre, le boot s'arrête toujours au montage NFS.

  Si l'on compile le driver en tant que module, alors le boot se passe normalement. cf */etc/rc.d/rc.sysinit :152* :

  ```
  echo "modprobe e1000" > /etc/sysconfig/modules/e1000.modules
  chmod +x /etc/sysconfig/modules/e1000.modules
  ```

  Avec le noyau monolitique on s'en sort en modifiant les paramètres de boot depuis PPCMON.

  ```
  PPCMon> set boot_line
   -> ip=::::eth1
  ```

  Cf le mail de **Rupert** :

  ```
  If you have a PMC ethernet controller plugged, the on-board ethernet will
  be enumerated later because its driver is loaded later.
  To boot from on-board ethernet we use the boot_line argument "ip=::::eth1" to
  use the other device for boot-time IP configuration.
  ```

2.2 Pizza Box

La carte utilisée n'est pas référencée sur le site d'Intel.

```
[root@lpnhess211 sysconfig]# lspci | grep Ethernet
1e:00.0 Ethernet controller: Intel Corporation 80003ES2LAN Gigabit Ethernet Controller (Copper) (rev 01)
1e:00.1 Ethernet controller: Intel Corporation 80003ES2LAN Gigabit Ethernet Controller (Copper) (rev 01)

# lspci -n | grep 1e:00.
1e:00.0 0200: 8086:1096 (rev 01)
1e:00.1 0200: 8086:1096 (rev 01)
```

Elle utilise le module **e1000e**.

```
# ethtool -i eth1
driver: e1000e
version: 0.3.3.3-k6
firmware-version: 1.0-11
bus-info: 0000:1e:00.1
```

2.3 PC Nec

Realtek RTL8168 Gigabit Ethernet controller

```
# lspci | grep Gigabit
03:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller (rev 01)

# lspci -n | grep 03:00.0
03:00.0 0200: 10ec:8168 (rev 01)
```

Elle utilise le module **r8169**.

```
# ethtool -i eth0
driver: r8169
version: 2.2LK-NAPI
firmware-version:
```

3 Configuration

Cf le mail de **Rupert** :

```
We have achieved around 600-700 MBit with RIO3 hardware (Which should
be similar to your RIOC), but it requires some fine-tuning to achieve
such performance. I will try to find the old docs, but from my memory,
you can try the following:
- Set prefetch size to maximum
  You can set it in PPC_Mon with
  set bridge_lpci_prefetch 32
  (replace lpci with cpci if you have the card on cpci)
  you can verify the correct setting in linux when the xpc driver
  is loaded with
  cat /proc/bus/xpc/lpci/info
- use jumbo frames
  you can alter the mtu in linux with
  ifconfig eth0 mtu 9000
  note that you need to set the same mtu on all network interfaces on that
  network and you switches need to support it (they usually do).
- cache flushing
  This is already part of the standard CES kernel, and you don't need to
  activate it.
  We do an explicit cache flush before data written by the CPU is read by a
  PCI device. This avoids retries on the XPC bus which are very costly on the
  RIO hardware.
```

Note : Afin de ne pas dégrader les performances, ne pas placer la passerelle par défaut sur l'interface gigabit.

3.1 Carte controleur

Via le shell PPCMon :

```
> set bridge_lpci_prefetch 32
> boot
# modprobe xpc
# cat /proc/bus/xpc/lpci/info
```

Fichier */etc/sysconfig/network-scripts/ifcfg-eth0* sur la carte controleur.

```
DEVICE=eth0
IPADDR=192.168.8.1
NETMASK=255.255.255.0
NETWORK=192.168.8.0
BROADCAST=192.168.8.255
ONBOOT=yes
NAME=gigabit
MTU=9000
```

Nous avons essayé d'utiliser la même interface sur la pizza box et donc une seconde IP sur le même réseau depuis zora :

Fichier */etc/sysconfig/network-scripts/bonnet-bleu* sur la carte controleur.

```
if [ $1 == "down" ]
then

  route del -host 192.168.1.40 dev eth0
  route del -host 192.168.1.41 dev eth0

else

  route del -net 192.168.1.0 netmask 255.255.255.0 dev eth0
  route del -net 169.254.0.0 netmask 255.255.0.0 dev eth0
  route add -host 192.168.1.40 dev eth0
  route add -host 192.168.1.41 dev eth0

fi
```

En fait les route ne sont d'aucune utilité car ici nous n'avons pas configuré le noyau pour agir en tant que routeur (*/proc/sys/net/ipv4/ip_forward*). Nous utilisons finalement un sous réseau. La solution aurait peut-être été amenée par iptables mais bon...

3.2 Pizza Box

Fichier */etc/sysconfig/network-scripts/ifcfg-eth0* sur la pizza box.

```
# Intel Corporation 80003ES2LAN Gigabit Ethernet Controller (Copper)
DEVICE=eth0
BOOTPROTO=none
BROADCAST=192.168.8.255
HWADDR=00:21:85:6a:40:4c
IPADDR=192.168.8.2
NETMASK=255.255.255.0
NETWORK=192.168.8.0
ONBOOT=yes
DNS1=192.168.1.3
DNS2=134.158.152.146
NM_CONTROLLED=no
TYPE=Ethernet
USERCTL=yes
PEERDNS=yes
IPV6INIT=no
DNS3=134.158.69.191
SEARCH=in2p3.fr
MTU=9000
```

4 Connectique

Définition IEEE802.3ab : 1000BASE-T

- Support minimum : câble en paires de cuivre torsadées non blindées de catégorie 5.
- Longueur maximale 100m

Cette définition est très importante. C'est elle qui permet d'utiliser le Gigabit Ethernet dans la majorité des installations actuelles.

Cela dit, les installations existantes auront certainement besoin d'une 'requalification'. Cette technologie utilise les câbles FTP (Foiled twisted pairs) de catégorie 5 au maximum de leur certification. De nouvelles catégories de câbles sont utilisables : 5enhanced à 100MHz, 6 à 200MHz, 6a à 500MHz et 7 à 600MHz. Il est recommandé de limiter au maximum les brassages intermédiaires dans les armoires de câblage.

Avec les cartes gigabit, il ne faut pas obligatoirement un câble Ethernet croisé pour tester la connection de Pc à Pc.

5 Tests des débits

5.1 Programmes utilisés

- Mesure via PING. En utilisant le script ci-dessous, on trouve des débits trés réduits.

```
#!/bin/sh
IP=192.168.2.1
NB_PING=10
SIZE=996   # +headers => paquets de 1024 octets

# le premier ping est toujours plus long (DNS...)
ping -c 1 $IP > /dev/null

# moyenne sur les pings suivant
averageTime=$(ping -s $SIZE -c $NB_PING $IP | tail -n 1 | cut -d'/' -f 5)

# On envoie 1024*8 bits en t milliseconde.
# d (b/s)  = 1024*8 * 1000/t
# d (Mb/s) = 1024*8 * 1000/t / 1024 / 1024
# d (Mb/s) = 8 * 1000/t / 1024

debit=$(bc << EOF
scale = 3;
8*1000/$averageTime/1024
EOF)
echo $debit Mb/s
```

- IPERF A télécharger puis compiler sur chaque plateforme.
  Note : ne compile pas sur les pizza box. Il faut utiliser la version compiler sur le PC.
- Le programme suivant est analogue à IPERF. Il est utilisé en parallèle d'IPERF.

```
lpnhp90$ cvs -d :pserver:roche@lpnp90.in2p3.fr:/home/cvsroot co netSpeed
```

5.2 Permiers tests

Nous utilisons iPerf pour les tests et le programme maison pour affiner les résultats. Nous avons utilisé 3 câbles différents sans observer de différence de débits.

- Catégorie 6 UTP
- Catégorie 5e FTP
- Catégorie 5 UTP

Debits (bits/s) : TCP / UDP émission / UDP réception

depuis/vers	CES	Pc Nec	Pizza Box
CES	320M / 470M / 440M	280M / 455M / 172M	280M / 460M / 173M
Pc Nec	272M / 4.5G / 260M	9G / 10.5G / 15.2G	380M / 4.6G / 390M
Pizza Box	380M / 3.7G / 320M	941M / 3.4G / 670M	8.5G / 12.5G / 7.8G

Conclusions provisoires :

- On ne se rapproche de gigabit que dans la configuration Pizza Box vers Pc Nec.
- Serait-ce possible que la puissance de la machine émettrice ait de l'influence ?
- Serait-ce possible que la puissance de la machine réceptrice soit aussi un facteur limitant ?
- La carte CES qui devra jouer le rôle de l'émetteur envoit au maximum **280** Mbits/s.

5.3 Tests après amélioration

L'amélioration effectuée est celle décrite dans le second mail de Ruppert donné ci-dessus :

- set prefetch size to maximum (`set bridge_lpci_prefetch 32`)
- use jumbo frames (`ifconfig eth0 mtu 9000`)

Source wikipedia : In computer networking, jumbo frames are Ethernet frames with more than 1,500 bytes of payload (MTU). Conventionally, jumbo frames can carry up to 9,000 bytes of payload, but variations exist and some care must be taken when using the term. Many, but not all, Gigabit Ethernet switches and Gigabit Ethernet network interface cards support jumbo frames, but all Fast Ethernet switches and Fast Ethernet network interface cards support only standard-sized frames.

Cette amélioration permet d'atteindre **520** Mbits/s en émission (et 688 Mbits/s en réception).

5.4 Optimisation de la taille des trames envoyées

A priori si l'on programme une socket TCP et que l'on envoie un buffer de la même taille que celui envoyé par iPerf on trouve le même débit.

TODO : Expliquer les faibles débits à la récéption avec le protocole UDP.

6 Test de la fibre optique

- Enfichement de la fibre optique.
  Parfois, il faut faire pivoter les connecteur de 90 degrés afin de pouvoir les déconnecter.

6.1 Carte Processeur / Pc Nec

Les résultats ne sont pas dégradés.

6.2 Pizza Box vers Pc Nec

Le débit de l'ordre du gigabit par seconde n'est pas dégradés.
On se sert de cette connexion pour tester les nouvelles fibres.

7 Test de la configuration finale

- La carte processeur est relié au switch data via un cable ETHERNET.
- Le switch est configuré avec un MTU à 9000 et emprunte la liaison fibre optique
- Le convertisseur optique vers ETHERNET est relié à la pizza box par un cable ETHERNET

Cette configuration permet d'atteindre :

- **450** Mbits/s en émission TCP avec des envois de 10ko (soit 200 micro s)
- **550** Mbits/s en émission TCP avec des envois de 100ko
- **595** Mbits/s en émission TCP avec des envois de 500ko
- **599** Mbits/s en émission TCP avec des envois de 1Mo
- **595** Mbits/s en émission UDP avec des envois de 10ko (soit 160 micro s)