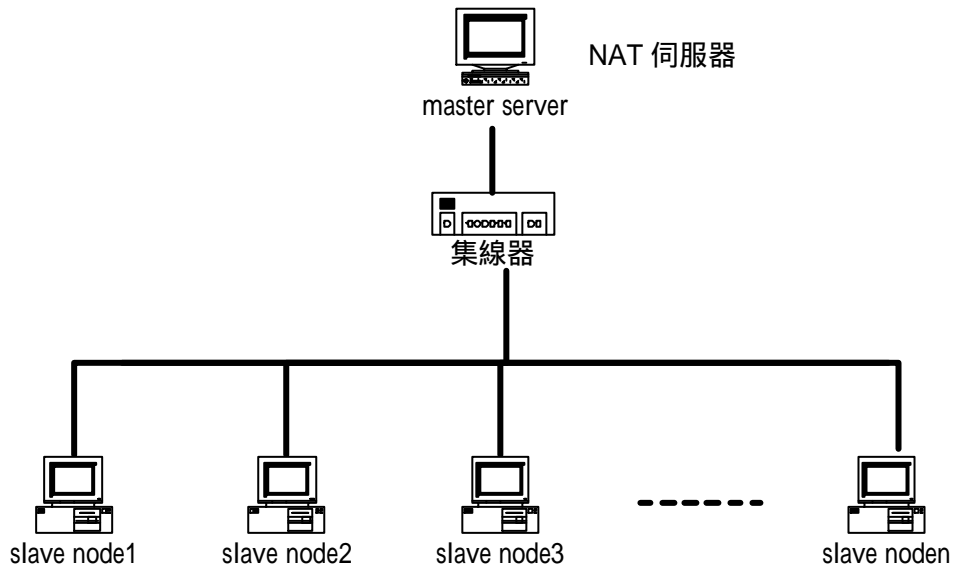




Linux 叢集伺服器

1-1 叢集伺服器概念

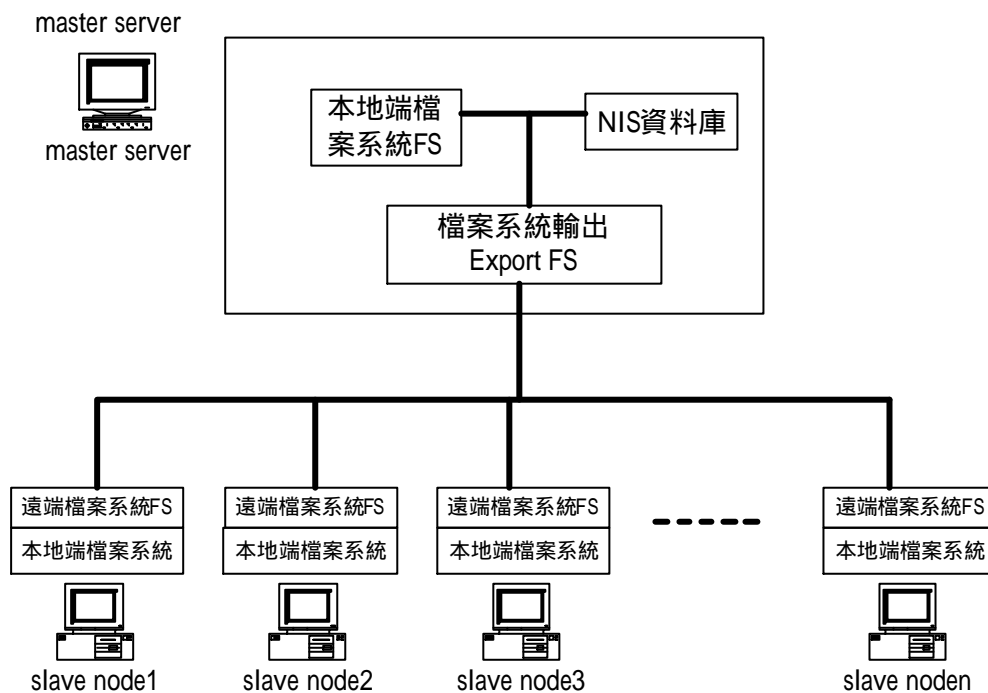
我們可以將多台電腦組成一個大型的 Linux 叢集伺服器，也就是將運算由主電腦 master 分配給次電腦 slave 來運算，然後再將次電腦運算的結果傳回主電腦，這種運算過程也稱為平行運算。一百台電腦就有 100 顆 CPU，也就是一般電腦運算的 100 倍，一千台電腦就有 1000 顆 CPU 其運算速度就會有一般電腦的 1000 倍，在這裏我們假設 Linux 叢集伺服器運算效能不受到網路或其它分散影響。一台超級電腦的價格比我們同等級的 Linux 叢集伺服器還貴，因此如何將這些個人電腦組成一個大型的運算體，將可以節省成本和提高整體效能。Linux 叢集伺服器是由一個 master 伺服器和多個 slave 伺服器所組成，master 伺服器透過網路將運算分散給這些 slave 伺服器執行，當運算完之後再將結果傳回 master 伺服器。管理叢集電腦就是管理 N 台電腦一樣，它的管理成本就像管理 N 台的成本，但是一台大型電腦，它只要一個大型電腦的管理費用，因此叢集 cluster 的電腦管理成本很高。叢集電腦的另外一個缺點是傳輸速度，我們通常使用網路來傳輸 master 和 slave 的傳輸，因此它比大型電腦使用匯流排來作資料 IO 還慢。在圖中，我們的 master 伺服器透過網路和多個 slave 伺服器溝通，這樣就組成一個 Linux cluster 叢集伺服器。



叢集cluster的結構

在叢集系統中有下列各項特色。

1. 其中一台為 master 伺服器其餘為 slave node 節點。
2. 在 Master 上有兩張網路卡，一張為對外 61.218.29.3，一張為對內 192.168.0.5，對外是連接到網路，對內是連結到叢集 cluster。
3. Master 伺服器提供 NIS 伺服器網路資訊系統，作為每個 slave 連接的帳號分享伺服器，而每個 slave 節點則安裝 NIS 使用者端。
4. 一般使用者可以使用 rsh 從 master 伺服器登錄 slave，而不用輸入密碼。
5. Master 伺服器使用 NFS 來輸出檔案系統來分享，每個 slave 節點則使用 mount 來掛載 Master 所分享出來的檔案系統。
6. 每個 slave 節點都有它自己的工作，或只作運算。



叢集cluster的結構

Linux 叢集伺服器安裝順序。

1. 先安裝 master 主機，並設定 NAT 伺服器於 master，並設定一個為對外 61.218.29.3，另外一個為對內的 IP 192.168.0.5。
2. 設定 master 啟動 rlogin、rsh、rexec、rcp 和 rwho 等遠端登錄伺服器。
3. 在 master 主機上設定 NFS 網路檔案系統，並且將 /usr/local 和 /home 目錄分享給 slave。/usr/local 是放置平行運算 MPI 軟體 mpich 的目錄。
4. 在 master 主機上設定 NIS 伺服器，讓它掌管所有帳號和密碼，如果有新增 slave，則執行 ypasswd 來更新 NIS 中的帳號和密碼。
5. 安裝平行運算軟體 mpich 於 master 上。
6. 設定每個 slave 啟動 rlogin、rsh、rexec、rcp 和 rwho 等遠端登錄伺服器。
7. 安裝每一個 slave 節點，並且在 slave 中指定 NIS 伺服器的領域名稱和指定 NIS 伺服器。
8. 在每個 slave 上的 /etc/fstab 上設定掛載 master 的 NFS 檔案系統，這樣在 slave 開機時，就會掛載 master 的 NFS 檔案系統。

1-2 在 Linux 上安裝 master 和 slave 叢集軟體套件

我們在實作 Linux 叢集伺服器中所需要安裝的軟體可分為 Master 和 slave 兩方面。Master 主機安裝需要防火牆 NAT 的設定、NFS 伺服器、RSH 使用者端、編譯器、MPICH 平行函式庫和程式。Slave 主要需要安裝 NIS 使用者端、NFS 使用者端和 RSH 伺服器。

我們現在可以用兩台 PC 來實作 Linux 叢集伺服器。一台當作 Master，一台當作 slave。我們設定 master 對內的 IP 為 192.168.0.5，對外的 IP 為 61.218.29.3，並且設定 NAT。我們設定 slave 為 192.168.0.6 的 IP。

我們在 master 主機上設定名稱查詢系統 /etc/hosts。

```
#vi /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          flash.aasir.com flash localhost.localdomain localhost
192.168.0.5       flash.aasir.com flsah
192.168.0.6       mandrake.aasir.com mandrake
```

1-3 組態 master 伺服器的開道和 NAT

我們使用 `iptables -t nat -L -n` 就可以觀看我們 iptables 網路濾器在 NAT 伺服器上的串列規則。

我們使用 `echo 1 > /proc/sys/net/ipv4/ip_forward` 來啟動 IP 轉向 forwarding。

我們使用 `iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE` 來輸出偽裝的封包。-t nat 是指定使用 NAT 表。-A 是指將規則加入串列中。POSTROUTING 只用在 NAT 表上，是轉向或修改輸出封包的規則。-s 192.168.0.0/24 是指定來源的位址。-o eth0 是指定輸出的網路介面卡 eth0，用來轉向或輸出串列。-j 指定轉向目標的連接埠。MASQUERADE 為偽裝。

```
#iptables -t nat -L -n
```

```
#vi nat.sh
```

```
echo "1" > /proc/sys/net/ipv4/ip_forward  
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE
```

執行 NAT。

```
#!/.nat.sh
```

1-3-1 防火牆：iptables 和 NAT

一個好的網路安全基礎，就是在我們作業系統中建立防火牆，來保護我們作業系統來自未授權的攻擊。我們可以使用防火牆來作封包過濾或代理。封包過濾就是由我們的防火牆決定是否讓外面的網路封包進入到我們作業系統或內部網路。封包過濾檢查封包的來源位址和它要到達的目的地。網路過濾軟體套件實作套件過濾和 NAT(網路位址轉換)的工作。我們可以使用 iptables 指令來實作網路過濾和 NAT 的工作。iptables 軟體已經內建到 2.4Kernel(linux2.4 核心)，其名稱為 iptable_filter.o。iptables 指令是比 ipchains 指令還好用，而且還有延伸性。網路過濾器 iptables 指令的參數有封包過濾、使用者定義串列、ICMP 封包、連接埠、狀態、NAT(網路位址轉換)。

語法：

指令：iptables 參數

參數：

這是命令參數

-A chain 串列：將規則加入串列中。

-D chain 串列：從串列中刪除合適的規則。

-D chain 規則編號：從串列中刪除規則編號(1=first)。

-I 串列 規則編號：將規則編號插入串列中。

-R 串列 規則編號：取代在串列中的規則編號。

-L 串列：列出所有串列的規則。

-E 串列：修改串列的名稱。

-F 串列：刪除所有串列中的規則。

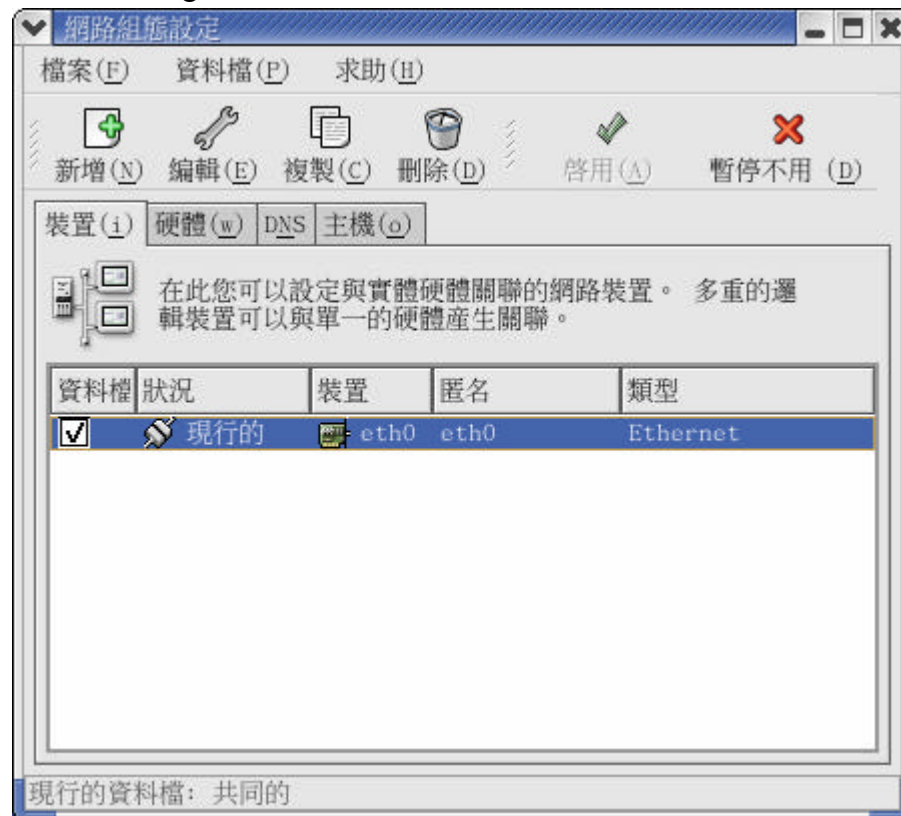
- R 串列：取代規則。
- Z 串列：將所有規則的封包和串列計數歸 0。
- N 串列：建立新的使用者定義串列。
- X 串列：刪除使用者定義串列。
- P 串列：改變目標串列的策略。

1-3-2 設定內部 slave 的網路組態

我們設定 slave 節點 1 的網路裝置。我們設定可以有 253 台 slave 主機，192.168.0.0/24。

執行 RedHat Linux Fadora 1 網路裝置。

```
#redhat-config-network&
```



這是我們內部虛擬網路的電腦，它的 IP 位址為 192.168.0.6。它的網路遮罩是 255.255.255.0。其閘道位置是 192.168.0.5。

乙太網路裝置

一般(G) 路由(R) 硬體裝置(H)

匿名(N): eth0

電腦啟動時啓用裝置(A)

允許所有的使用者啓用與停用這個裝置

在此自動取得 IP 位址設定: dhcp

DHCP 設定

主機名稱(可不輸入的)(H):

從提供者自動取得 DNS 資訊

靜態設定 IP 位址:

手動 IP 位址設定

位址(A): 192.168.0.6

子網路遮罩(S): 255.255.255.0

預設閘道器位址(G): 192.168.0.5

確定(O) 取消(C)

1-3-3 設定 master 的 NAT 伺服器的兩張網卡

設定 master 的 NAT 伺服器的兩張網卡

#nredhat-config-network

我們使用 eth0 網卡來對外，而使用 eth1 網卡來對內。eth0 是對外的網路卡，它的位址是 61.218.29.3、網路遮罩是 255.255.255.248、閘道是 61.218.29.1。
這是 eth0 網路卡的硬體裝置。

乙太網路裝置

一般(G) 路由(R) 硬體裝置(H)

匿名(N): eth0

電腦啟動時啟用裝置(A)

允許所有的使用者啟用與停用這個裝置

在此自動取得 IP 位址設定: dhcp

DHCP 設定

主機名稱(可不輸入的)(H):

從提供者自動取得 DNS 資訊

靜態設定 IP 位址:

手動 IP 位址設定

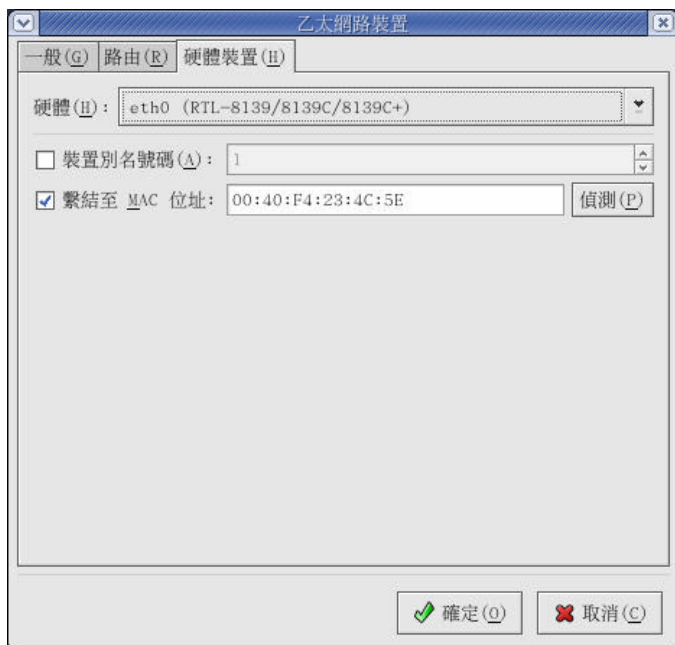
位址(A): 61.218.29.3

子網路遮罩(S): 255.255.255.248

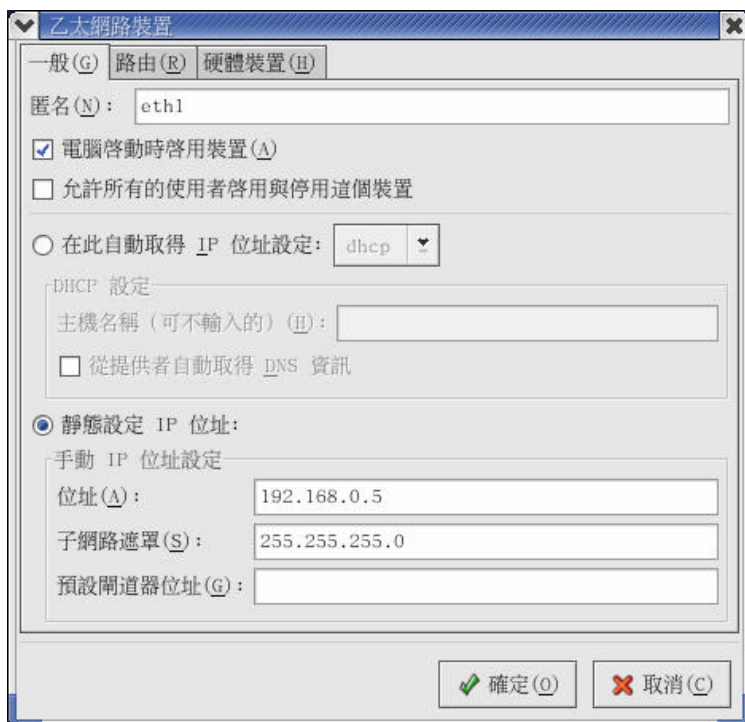
預設閘道器位址(G): 61.218.29.1

確定(O) 取消(C)

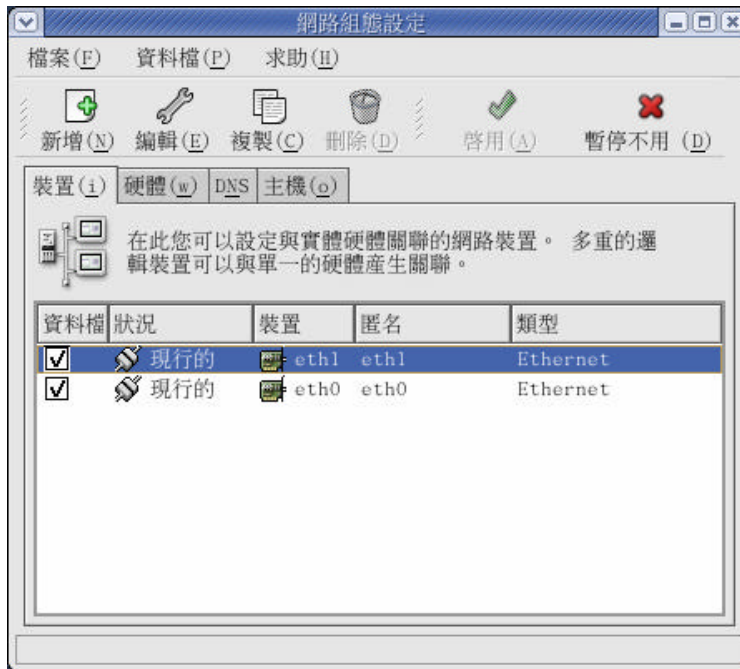
我們使用 eth0 網卡來對外，而使用 eth1 網卡來對內。eth0 是對外的網路卡，它的位址是 61.218.29.3、網路遮罩是 255.255.255.248、閘道是 61.218.29.1。這是 eth0 的硬體裝置，顯示其 MAC 位置。



這是第二張網路卡 eth1 的設定，它的匿名為 eth1，而它的位址為 192.168.0.5，而它的網路遮罩是 255.255.255.0。



這是兩張網路卡 eth0 和 eth1 裝在 NAT 伺服器上。



這是 NAT 伺服器的主機，名稱為 flash.aasir.com，而其 IP 為 61.218.29.3。



這是 eth0 網路卡 DNS 的設定，我們設定其名稱伺服器為 168.95.1.1。



1-4 組態 RSH 伺服器的伺服器端和使用者端

因為我們的 master 伺服器的 mpich 會自動的經由遠端 shell 登錄 slave 並且執行平行運算。所以我們要啟動 master 和 slave 的 rsh 伺服器、rlogin 伺服器和 rexec 伺服器，我們輸入 serviceconf，並且選取伺服器。

```
#serviceconf&
```



我們也可以輸入 chkconfig 指令來啟動 rsh、rlogin 和 rexec 指令。

```
#/sbin/chkconfig --level 35 rsh on
```

```
#/sbin/chkconfig --level 35 rlogin on
```

```
#/sbin/chkconfig --level 35 rexec on
```

我們在 master 和 slave 上的/etc/hosts.equiv 設定 flash +。這是設定主機可以由那一些登錄，且不用密碼的認證，在這裏內容填入主機的名稱，空一格(tab)，在加上”+”符號。”+”符號是表示所有的 user，都可以用 RSH 的方式登錄此主機，而且不用認證，可以使用 man hosts.equiv 來查詢相關的使用資訊。

```
#vi /etc/hosts.equiv
```

```
flash +
```

1-4-1 master 上設定 .rhosts

我們的使用者 test 要能夠不用認證而以 RSH 的方式登入主機，則在 /home/test 的目錄下要設定 .rhosts，我們只要在 .rhosts 來設定允許的主機 flash 就可以了。如果是使用其它的使用者來執行平行運算，則修改其家目錄下的 .rhosts 檔案，並且加入允許的主機就可以了。

```
#vi /home/test/.rhosts
```

```
flash
```

```
#vi /etc/skel/.rhosts
```

```
flash
```

我們可以嘗試使用 test 使用者直接登錄 mandrake，而不用輸入密碼。在 NIS 網路資訊系統安裝過後就可以測試看看。

```
$rlogin mandrake
```

1-5 組態 NFS 伺服器端和使用者端

在 master 主機上設定 NFS 網路檔案系統，並且將 /usr/local 和 /home 目錄分享給 slave。/usr/local 是放置平行運算 MPI 軟體 mpich 的目錄。我們在使用者端使用 mount 來掛載，也可以編輯 /etc/fstab，在開機時 slave 會自動掛載 master 所輸出的檔案系統。/usr/local 是放置 mpich 軟體的地方，也是執行時的地點。/home 目錄是所有使用者的家目錄，大家可以共用這個家目錄的帳號和密碼。

1-5-1 NFS 伺服器端 master

我們設定將 master 的檔案系統 /home 和 /usr/local 給分配出去，只有 192.168.0.0 網域的使用者才能掛載使用這個檔案系統。

```
#vi /etc/exports
```

```
/home 192.168.0.0/24(rw,async,no_root_squash)
```

```
/usr/local 192.168.0.0/24(rw,async,no_root_squash)
```

這是啟動伺服器端的檔案系統。

```
#/sbin/chkconfig --level 35 nfs on
```

1-5-2NFS 使用者端 slave

我們可以讓 slave 主機在啟動時就掛載遠端 flash 的檔案系統。

```
#vi /etc/fstab
```

```
LABEL=/          /          ext3    defaults          1 1
none            /dev/pts   devpts  gid=5,mode=620    0 0
none            /proc      proc    defaults          0 0
none            /dev/shm   tmpfs   defaults          0 0
flash:/home     /home      nfs     auto,hard,bg,intr 0 0
flash:/usr/local /usr/local nfs     auto,hard,bg,intr 0 0
/dev/hda2       swap       swap    defaults          0 0
/dev/cdrom      /mnt/cdrom udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/cdrom1     /mnt/cdrom1 udf,iso9660 noauto,owner,kudzu,ro 0 0
```

/etc/fstab 掛載語法：

來源裝置	掛載點	檔案型態	選項	n	n
------	-----	------	----	---	---

如果來援裝置是遠端的網路檔案系統，則以主機名稱：來源裝置來定義。n 可以設定為 0。我們可以使用 man fstab 來觀看其它欄位的內容。

這是啟動使用者端的 nfs 檔案系統。

```
[root@flash root]# /sbin/chkconfig --level 35 nfs on
[root@flash root]# /sbin/chkconfig --level 35 portmap on
```

1-5-3NFS 概念

NFS 就是網路檔案系統。網路檔案系統可以將遠端的檔案系統掛載 mount 到我們的檔案系統下。我們這樣就可以透過網路 TCP/IP 直接存取遠端的檔案。

NFS 是 Unix 系統所使用的標準檔案分享通訊協定。NFS 允許一個系統將其目錄透過網路分享給其他的主機，允許在其他主機上的使用者或程式可以存取被匯出的檔案，並認為該檔案是在本地端系統上。

NFS 伺服器是將一個或多個目錄匯出的系統。NFS 客戶端則是從遠端伺服器掛載一個或多個目錄的系統。一個主機可以同時是 NFS 伺服器與其他伺服器的客戶端。伺服器經由檢查客戶端的 IP 位置是否在列表中，以決定該客戶端是否可以掛載匯出的目錄。伺服器可以選擇以唯讀的方式或可讀寫的方式匯出給特定的客戶端。不像其他的網路檔案系統，NFS 客戶端再掛載目錄時，不需要登入到伺服器中。NFS 伺服器將信任可以存取客戶端系統的使用者存取匯出的目錄。所以我們應該只將目錄匯出給可信任的系統。讓我們從我們的系統經由 NFS 分享系統上的目錄 (在 Linux 中稱為匯出)，這顯示出分享的目錄與可以存取分享的客戶端。

```
#vi /etc/exports
```

我們使用/home/chaiyen aasir.com(rw)將我們主機的/home/chaiyen 目錄輸出給 aasir.com , 並給於 aasir.com 讀取和寫入(rw)的權限。我們使用/home 61.218.29.0/255.255.255.248(rw,no_root_squash)來輸出/home 目錄給網路位址為 61.218.29.0 到 61.218.29.7 的區域網路其遮罩是 255.255.255.248。我們使用 /home/virtual2 *(secure)來將我們/home/virtual2 目錄輸出給大家都可存取。

```
#
/home/chaiyen aasir.com(rw)
/home 61.218.29.0/255.255.255.248(rw,no_root_squash)
/usr (rw,no_root_squash,insecure)
#/home/virtual (rw,insecure)
/var/www/ (rw)
/pub aasir.com(rw)
/lib aasir.com
/opt aasir
/home/virtual2 *(secure)
/ (rw)
```

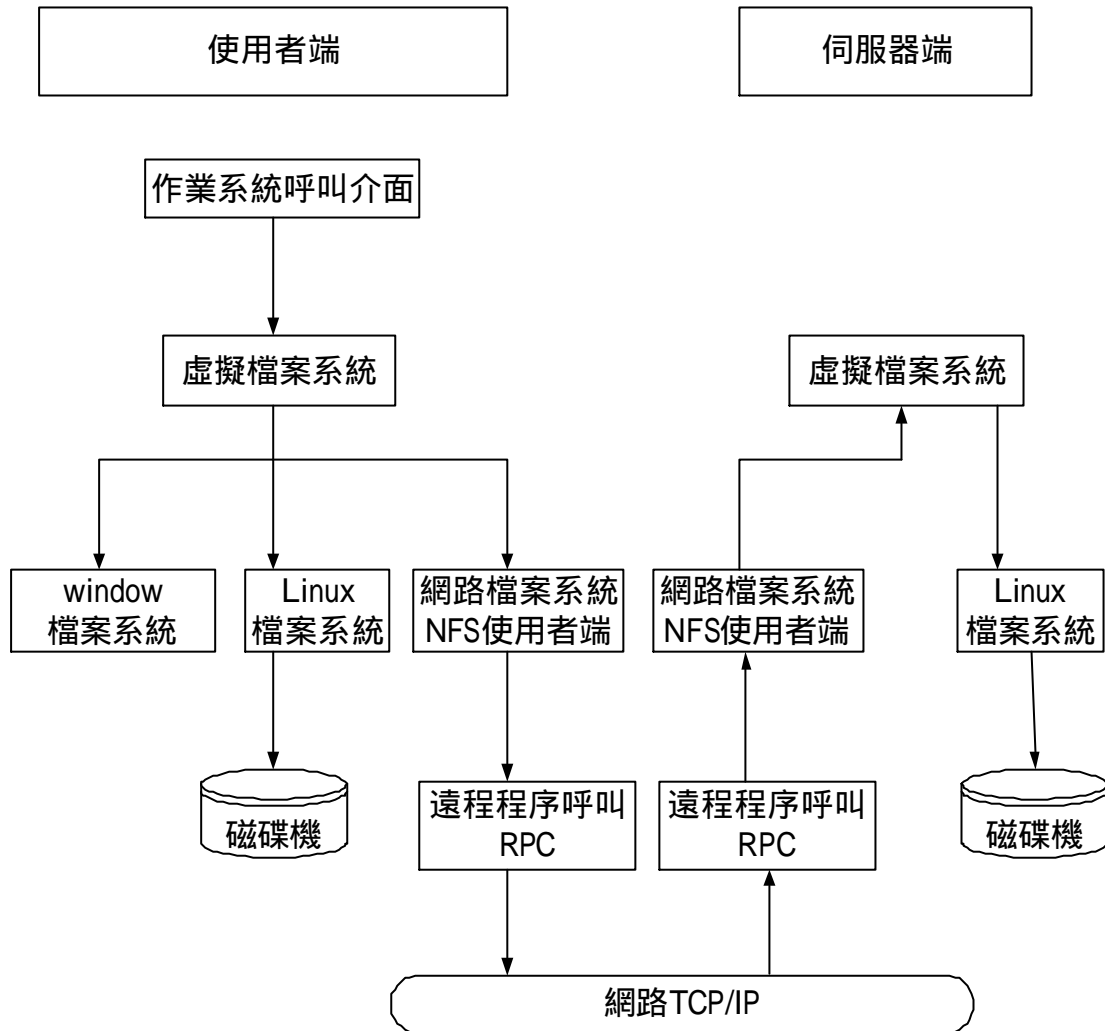
語法：目錄路徑

主機名稱(選項)

一般選項	說明
secure	主機必需經過 RPC(選端存取呼叫)認證才能夠存取檔案系統。為預設值。
insecure	關閉 secure 選項
ro	允許可讀(Read-only)。為預設。
rw	允許可讀可寫(read-write)。
sync	當請求時執行寫入動作。為預設。
async	當伺服器端準備好時，執行寫入動作。
no_wdelay	立刻執行寫入動作。
wdelay	為預設。檢查是否寫入相關，如果是，將它們一起執行。
hide	自動隱藏這其它輸出目錄的子目錄。這子目錄一定要被明確的掛載存取，且不允許掛載它的上一層目錄存取。為預設。
no_hide	不隱藏這其它輸出目錄的子目錄。只有單一主機可以使用。
subtree_check	檢查檔案系統的上一層目錄，來證實輸出的子目錄。為預設。
No_subtree_check	不檢查檔案系統的上一層目錄，來證實輸出的子目錄。
Insecure_locks	不需要鎖住需求的證明。

使用者 ID 對應選項	說明
all_squash	對應所有使用者 uid 和群組 gid 到匿名使用者，對 NFS-exported 給大家的 FTP 目錄非常有用。
no_all_squash	沒有使用 all_squash。為預設值。
root_squash	對映遠端超級使用者的需求給匿名使用者或群組 uid/gid。為預設
no_root_squash	關閉 root_squash。允許超級使用者當作遠端超級使用者存取。
anonuid	設定匿名使用者帳號的使用者 uid 和群組 gid。
anongid	給 all_squash 和 root_squash 的選項。預設是 nobody 和 nogroup。

我們使用者可以透過 NFS 網路檔案系統來使用遠端伺服器的磁碟機檔案。我們使用者端程式使用虛擬檔案系統，再轉成 NFS 網路檔案系統，經過遠程程序呼叫 RPC 再經過網路，來讀取遠端伺服器的網路檔案系統 NFS，這時伺服器端的 NFS 網路檔案系統會轉成虛擬檔案系統，再來存取伺服器端的檔案系統。



網路檔案系統NFS的讀取檔案情況

1-6 組態 NIS 伺服器端和使用者端

NIS 就是維護多台在網路的 Linux 機器，並且如何將資訊分享。這些分散於網路的資訊其組成是由使用者資料庫和授權資訊，像是/etc/passwd 和/etc/group。假如一個使用者不論在哪一台者機登錄，只要在 NIS 伺服器上有他的帳號和密碼，他就能夠通過。NIS 被稱為 Network Information System(網路資訊系統)。我們所使用者 Linux 機器可以分為 Client 使用者端的 NIS,和 SERVER 伺服器端的 NIS。NIS 的 Domain 是一個系統的管理工具，一個方便的方法來組織這些機器，並且只要經過 NIS 伺服器的授權就可以存取這些機器。

NIS 的伺服器端就是 Linux 叢集伺服器的 Master，而 NIS 的使用者端就是 Linux 叢集伺服器的 slave。

1-6-1NIS 伺服器端 Master

我們建立 NIS 伺服器端 Master。

我們先建立 NIS 的領域名稱，並且將它命名為 nistest。

```
[root@mandrake /]# nisdomainname nistest  
[root@mandrake /]# nisdomainname  
nistest
```

永久的儲存 NIS 領域名稱。我們在/etc/sysconfig/network 中新增下面一行。在這裏 nistest 是我們領域的名稱。

```
#vi /etc/sysconfig/network
```

```
NISDOMAIN=nistest
```

我們使用 `ypinit` 初始化 NIS 伺服器。我們在這裏增加我們主機的名稱 `flash.aasir.com`。輸入完後我們按下 `<CTRL+D>`。

```
[root@flash test]# /usr/lib/yp/ypinit -m

At this point, we have to construct a list of the hosts which will run NIS
servers.  flash.aasir.com is in the list of NIS server hosts.  Please continue t
o add
the names for the other hosts, one per line.  When you are done with the
list, type a <control D>.
    next host to add:  flash.aasir.com
    next host to add:  192.168.0.5
    next host to add:
The current list of NIS servers looks like this:

flash.aasir.com
192.168.0.5

Is this correct?  [y/n: y]  y
We need a few minutes to build the databases...
Building /var/yp/nistest/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/nistest'
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
```

我們建立 NIS 設定檔並加入下列兩行。

```
#vi /etc/ypserv.conf
```

```
主機: 領域 :Map :security
```

```
192.168.0.0/255.255.255.0 :* :* :none
```

```
127.0.0/255.255.255.255.0 :* :* :none
```

然後我們啟動 NIS 伺服器。

```
#/etc/rc.d/init.d/ypserv start
```

```
#/etc/rc.d/init.d/yppasswdd start
```

我們也可以使用 `ypcat passwd` 來觀看是否已經啟動 NIS 伺服器了。

```
[root@flash /]# /etc/rc.d/init.d/yppasswdd restart
停止 YP passwd 服務: [ 確定 ]
啓動 YP passwd 服務: [ 確定 ]
[root@flash /]# ypcat passwd
test:$1$IYcKFROQ$0HuDECPhTuP14R1xz503U0:501:501::/home/test:/bin/bash
chaiyen:$1$GOFwhco9$A29w2RLBeQx9/7nzzCF.N1:500:500:wuchaiyen:/home/chaiyen:/bin/bash
```

我們也可以在開機時啟動。

```
#/sbin/chkconfig - - level 35 ypserv on
```

```
#/sbin/chkconfig - - level 35 yppasswdd on
```

我們也可以使用 serviceconf 來設定服務組態設定。我們要啟動 ypbind yppasswdd 和 ypserv，這樣開機時就會啟動這些伺服器了。



1-6-2NIS 使用者端 slave

我們設定 NIS 的使用者端，也就是 slave 節點。

我們定義領域為 nistest。

```
[root@mandrake /]# nisdomainname nistest
[root@mandrake /]# nisdomainname
nistest
```

永久的儲存 NIS 領域名稱。我們在/etc/sysconfig/network 中新增下面一行。在這裏 nistest 是我們領域的名稱。

```
#vi /etc/sysconfig/network
```

```
NISDOMAIN=nistest
```

我們建立 NIS 查詢的主機名稱為 192.168.0.5，也就是 master 的 IP 位置。

```
#vi /etc/yp.conf
```

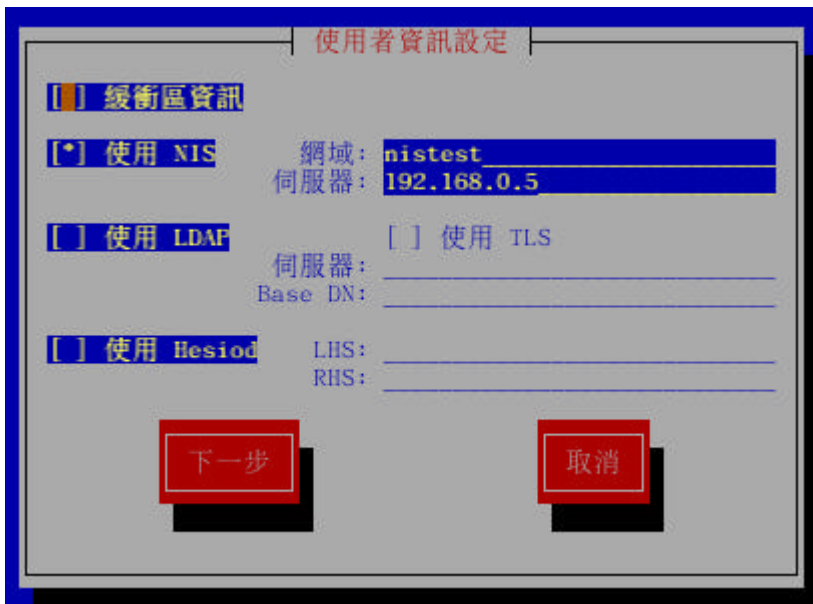
```
# /etc/yp.conf - ypbind configuration file
# Valid entries are
#
#domain NISDOMAIN server HOSTNAME
#    Use server HOSTNAME for the domain NISDOMAIN.
#
#domain NISDOMAIN broadcast
#    Use broadcast on the local net for domain NISDOMAIN
#
#ypserver HOSTNAME
#    Use server HOSTNAME for the local domain. The
#    IP-address of server must be listed in /etc/hosts.
#
domain nistest server 192.168.0.5
```

設定 NIS 使用者端，我們可以直接編輯/etc/yp.conf，或者使用 authconfig 工具。

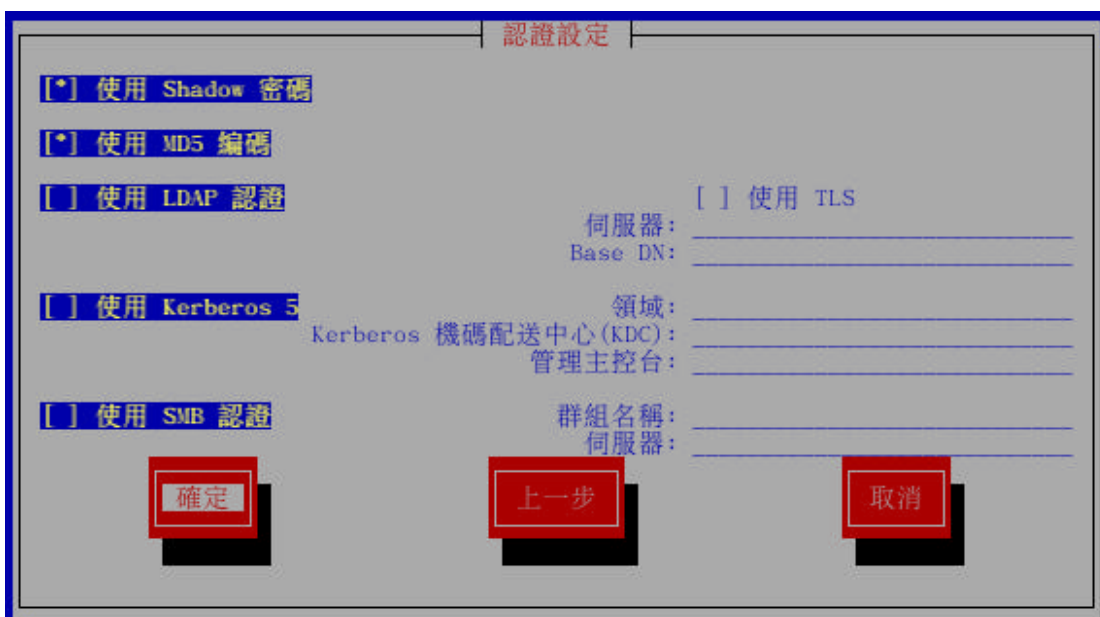
我們使用超級使用者身份 root 來執行 authconfig。

```
#authconfig
```

我們選取緩衝區資料，這樣可以加速快取，然後我們使用 Tab 鍵跳到使用 NIS，並且選取。我們設定網域為 nistest，而 NIS 伺服器為 192.168.0.5，伺服器是輸入 NIS 伺服器的 IP 位址。我們按下一步。



我們選取使用 Shadow 密碼和 MD5 編碼，然後使用 Tab 鍵到確定。



我們可以使用 rpcinfo 指令來觀看 NIS 的服務。

```
[root@mandrake /]# authconfig
停止 portmapper: [ 確定 ]
啓動 portmapper: [ 確定 ]
關閉 NIS 服務: [ 確定 ]
連結到 NIS 網域: [ 確定 ]
傾聽 NIS 網域伺服器的回應。
[root@mandrake /]# /etc/rc.d/init.d/ybind restart
關閉 NIS 服務: [ 確定 ]
連結到 NIS 網域: [ 確定 ]
傾聽 NIS 網域伺服器的回應。
[root@mandrake /]# /usr/sbin/rpcinfo -u localhost ybind
程式 100007 的第 1 版已經就緒並等待服務中
程式 100007 的第 2 版已經就緒並等待服務中
```

我們修改密碼驗證的方法，我們在 /etc/passwd 檔案的最後一行新增

```
+ : : : : : :
```

```
#vi /etc/passwd
```

```
desktop:x:80:80:desktop:/var/lib/menu/kde:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
chaiyen:x:500:500:chaiyen:/home/chaiyen:/bin/bash
justin:x:501:501:/home/justin:/bin/bash
linux:x:504:504:/home/linux:/bin/bash
+ : : : : : :
```

這是確認 NIS 使用者端的認證順序，我們編輯/etcnsswitch.conf。

```
#vi /etc/nsswitch.conf
```

```
passwd:      files nisplus nis
shadow:      files nisplus nis
group:       files nisplus nis
hosts:       files nisplus nis dns
```

我們可以啟動 NIS。

```
#/etc/rc.d/init.d/ybind start
```

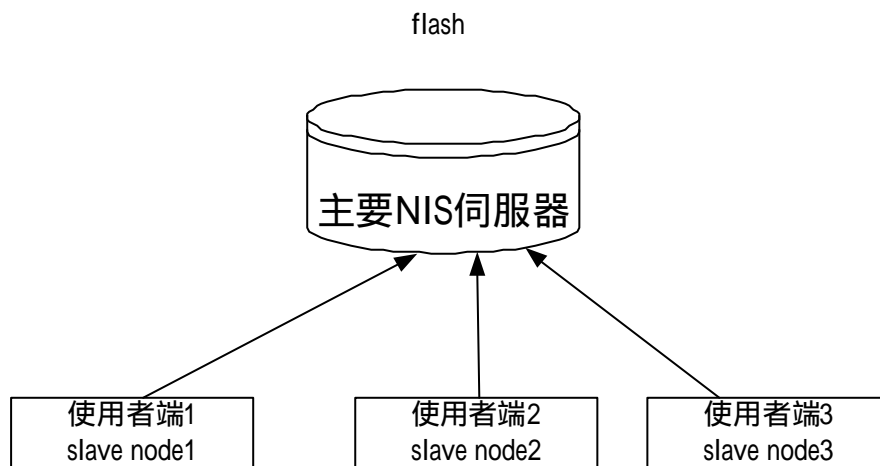
我們可以使用 ypcat passwd 來驗證 NIS 服務是否有提供，如果有出現帳號和密碼，則表示正確。

```
[root@mandrake /]# ypcat passwd
test:$1$1YcKFROQ$0HuDECPhTuP14R1xz503U0:501:501:/home/test:/bin/bash
chaiyen:$1$G0Fwhco9SA29w2RLBeQx9/7nzzCf.N1:500:500:wuchaiyen:/home/chaiyen:/bin/bash
```

1-6-3NIS 網路資訊系統概念

NIS 就是維護多台在網路的 Linux 機器，並且如何將資訊分享。NIS 就是在網路上分享其資訊。一開使 NIS 是由 SUN 開發，也就是一般所說的 Yellow Pages(YP)，所以 NIS 的指令都是以 yp 開頭，像是 ypserv、ypbind 和 yppasswd。這些分散於網路的資訊其組成是由使用者資料庫和授權資訊，像是/etc/passwd 和/etc/group。假如一個使用者不論在哪一台者機登錄，只要在 NIS 伺服器上有他的帳號和密碼，他就能夠通過。NIS 被稱為 Network Information System(網路資訊系統)。我們所使用者 Linux 機器可以分為 Client 使用者端的 NIS，和 SERVER 伺服器端的 NIS。NIS 的組態和 domain 很有關係。NIS 的 Domain 和 DNS 的 Domain 不同。一個 NIS 的 domain 名稱參考到一群系統，它是一個區域網路或者是一個子區域網路，它使用 NIS 的地圖 maps。NIS 的 Domain 是一個系統的管理工具，一個方便的方法來組織這些機器，並且只要經過 NIS 伺服器的授權就可以存取這些機器。

NIS 的資料庫是以 DBM 的格式儲存，一個以 ASCII 文字檔為基礎的二位元檔案。例如/etc/passwd 和/etc/group 能夠使用 makedbm 來被轉成 DBM 格式。這是單一 Domain 和單一 NIS 伺服器，這是最經常使用到的。這單一 NIS 伺服器回應所有從 NIS 使用者端的查詢，而且是這領域的單一資源，我們命名為 nistest。



單一Domain,單一NIS伺服器

1-6-4 組態 NIS 伺服器

NIS 是由單一 NIS 伺服器和多個使用者端所組成，其組態過程包含設定 NIS 領域名稱，組態和啟動伺服器領域 `ypbind`，初始化 NIS 地圖，啟動 NIS 密碼常駐行程，假如我們有使用次要 NIS 伺服器則要啟動 NIS 轉換常駐行程，最後是修改啟動行程來在系統啟動時啟動 NIS 常駐行程。

指令	說明
<code>nisdomainname</code>	設定系統領域名稱
<code>ypserv</code>	處理主要 NIS 伺服器責任區
<code>ypinit</code>	建立和安裝 NIS 資料庫
<code>yppasswdd</code>	處理使用者密碼在 NIS 環境中的改變
<code>ypxfrd</code>	加速轉移地圖從主要 NIS 伺服器到次要 NIS 伺服器
<code>yppush</code>	廣播更新 NIS 地圖到次要伺服器
<code>/var/yp/securenets</code>	列出允許存取 NIS 地圖的主機
<code>/etc/ypserv.conf</code>	儲存執行組態選項和特別主機存取目錄

組態 NIS 使用者端的第一步就是設定 NIS 領域名稱。當第一次組態 NIS 伺服器，就是使用 `nisdomainname` 指令來設定 NIS 的領域名稱。

我們使用 `nisdomainname` 來定義領域名稱。

```
#nisdomainname 領域名稱
```

我們在這裏設定 NIS 的領域名稱為 `nistest`，我們再使用 `nisdomainname` 來觀看其領域名稱。

```
[root@flash chaiyen]# nisdomainname nistest
[root@flash chaiyen]# nisdomainname
nistest
```

當 NIS 領域已經設定，我們可以組態和啟動這主要的 NIS 伺服器常駐行程。這主要的組態檔案是 `/etc/ypserv.conf`。`/etc/ypserv.conf` 是主要 NIS 伺服器常駐行程 `ypserv` 和 NIS 傳輸轉換行程 `ypxfrd` 的檔案。

`ypserv` 和 `ypxfrd` 讀取 `/etc/ypserv.conf`，當它們啟動和送出 `SIGUP` 信號。

在啟動伺服器前，我們要確定 portmap 常駐行程已經啟動。NIS 需要 portmap，因為 NIS 使用遠端程序呼叫(RPC)。我們可以使用/etc/rc.d/init.d/portmap status 來觀看目前 portmap 是否已經啟動，我們也可以使用 rpcinfo 指令來觀看。在這裏可以顯示 portmap 的行程正在啟動，而其行程編號是 1563。

```
[root@flash yp]# /etc/rc.d/init.d/portmap status
portmap (pid 1563) 正在執行中...
```

我們可以使用下列指令來啟動地圖。

```
#/etc/rc.d/init.d/portmap start
```

我們也可以使用 rpcinfo 指令來觀看地圖是否已經啟動。

```
[root@flash yp]# /usr/sbin/rpcinfo -p localhost|grep portmapper
100000    2    tcp    111    portmapper
100000    2    udp    111    portmapper
```

當我們的地圖已經啟動時，我們可以使用下列指令啟動 NIS 伺服器。

```
#/etc/rc.d/init.d/ypserv start
```

```
[root@flash yp]# /etc/rc.d/init.d/ypserv start
啟動 YP 伺服器服務:[ 確定 ]
```

我們使用下列指令來觀看 NIS 伺服器 ypserv 已經啟動了。

```
[root@flash yp]# /usr/sbin/rpcinfo -p localhost|grep ypserv
100004    2    udp    799    ypserv
100004    1    udp    799    ypserv
100004    2    tcp    802    ypserv
100004    1    tcp    802    ypserv
```

當 NIS 伺服器啟動，我們必需要建立 NIS 資料庫來當作 NIS 伺服器。我們可以使用 ypinit 指令。ypinit 建立一個我們系統完整的 NIS 地圖，而且放置它們在 /var/yp 的子目錄。這資料庫的資訊是由/etc/passwd、/etc/group、/etc/hosts、/etc/networks、/etc/services、/etc/protocols、/etc/netgroup 和/etcrpc 所組成。我們可以使用下列指令來建立 NIS 資料庫/usr/lib/yp/ypinit -m。

```
#/usr/lib/yp/ypinit -m
```

這個指令使用-m 的選項來指出建立主要 NIS 伺服器的地圖。假如我們也組態次要 NIS 伺服器，我們需要確定他們可以成功的和主要次服务器溝通。我們可以在次要伺服器上確定 ypwhich -m 指令。在組態完次要伺服器，我們可以執行下列指令/usr/lib/yp/ypinit -s masterhost。這-s 指出 ypinit 從主要 NIS 伺服器 masterhost 的資料庫來建立次要 NIS 伺服器。

我們輸入 `/usr/lib/yp/ypinit -m` 指令。我們輸入 NIS 伺服器的主機名稱，在這裏是 `flash.aasir.com`。當我們建立完成，我們可以輸入 `<control+D>` 來結束。

```
[root@flash yp]# /usr/lib/yp/ypinit -m
```

```
At this point, we have to construct a list of the hosts which will run NIS
o add
```

```
the names for the other hosts, one per line. When you are done with the
list, type a <control D>.
```

```
    next host to add: flash.aasir.com
```

```
    next host to add:
```

```
The current list of NIS servers looks like this:
```

```
flash.aasir.com
```

```
Is this correct? [y/n: y] y
```

```
We need a few minutes to build the databases...
```

```
Building /var/yp/nistest/ypservers...
```

```
Running /var/yp/Makefile...
```

```
gmake[1]: Entering directory `/var/yp/nistest'
```

```
Updating passwd.byname...
```

```
Updating passwd.byuid...
```

```
Updating group.byname...
```

```
Updating group.bygid...
```

```
Updating services.byname...
```

```
Updating services.byservicename...
```

```
Updating netid.byname...
```

```
Updating protocols.bynumber...
```

```
Updating protocols.byname...
```

```
Updating mail.aliases...
```

```
gmake[1]: Leaving directory `/var/yp/nistest'
```

```
flash.aasir.com has been set up as a NIS master server.
```

```
Now you can run ypinit -s flash.aasir.com on all slave server.
```

因為 NIS 通常用來分享授權資訊。當使用者改變他在 NIS 資料庫的資訊時，然後要將更新的資料廣播傳出到每一個 NIS 使用者端和次要 NIS 次要伺服器。當新增加使用者或者當使用者被刪除時，NIS 使用者端和次要 NIS 伺服器都要能夠知道。能夠處理這個的常駐行程是 `yppasswdd`。`yppasswdd` 處理密碼的改變和更新其它 NIS 資訊。`yppasswdd` 只有在 NIS 主伺服器上跑。我們可以使用 `yppasswdd` 來啟動 NIS 密碼常駐行程。

```
#/etc/rc.d/init.d/yppasswdd start
```

我們啟動密碼常駐行程。

```
[root@flash yp]# /etc/rc.d/init.d/yppasswdd start
啓動 YP passwd 服務:[ 確定 ]
```

我們確定是否 `yppasswd` 已經啟動。

```
[root@flash yp]# /usr/sbin/rpcinfo -u localhost yppasswd
程式 100009 的第 1 版已經就緒並等待服務中
```

當我們組態我們的 NIS 伺服器，我們應該讓系統持續的運行。這包括 NIS 領域名稱的持續存在，確保 ypserv 和 yppasswdd 能夠在系統啟動時啟動，在系統關機時停止，假如我們有使用次要 NIS 伺服器 ypxfrd 也是一樣。

第一步就是永久的儲存 NIS 領域名稱。我們在/etc/sysconfig/network 中新增下面一行。在這裏 nistest 是我們領域的名稱。

```
#vi /etc/sysconfig/network
```

```
NISDOMAIN=nistest
```

下一步是執行 serviceconf，這是新的 RedHat Linux Fedora 1 系統服務組態工具，我們可以組態 NIS 常駐行程，讓它開機時能夠啟動。我們選取主選單 系統設定 伺服器設定 服務。我們可以選取 yppasswdd 和 ypserv 兩個選項。然後我們選取檔案 儲存。這樣就可以開機時啟動主要 NIS 伺服器。假如我們有執行次要 NIS 伺服器，則選取 ypxfrd，在開機時就可以啟動。如果都已經設定好，這時 NIS 主伺服器已經啟動而且已經準備好給 NIS 使用者端作查詢。



1-6-5 組態 NIS 使用者端

當我們成功的組態完成至少一個 NIS 伺服器，我們就以經準備好要組態一個以上的 NIS 使用者端。建立 NIS 使用者端的程序包含設定 NIS 領域名稱、組態和啟動 NIS 使用者端常駐行程、測試使用者端常駐行程、組態使用者的啟始檔案使用 NIS 和重新啟動使用者端。

下列是 NIS 使用者端的組態檔和指令。

指令	說明
ypwhich	顯示主要 NIS 伺服器的名稱。
ypcat	顯示 NIS 資料庫的進入點。
yppasswd	改變在 NIS 伺服器上的使用者密碼和資訊。
yppoll	顯示伺服器和 NIS 地讀的版本編號。
ypmatch	顯示在 NIS 地圖上，一個或多個的數值。
/etc/yp.conf	組態使用者端 NIS 的聯結。
/etc/nsswitch.conf	組態系統名稱資料庫的查詢。
/etc/host.conf	組態主機名稱解析。

第一個步驟就是設定使用者端的 NIS 領域名稱。

我們使用 nisdomainname 來設定我們領域的名稱。

```
#nisdomainname 領域名稱
```

我們設定我們的領域名稱為 nistest，而我們使用 nisdomainname 來觀看所設定的領域名稱。

```
[root@mandrake chaiyen]# nisdomainname nistest
[root@mandrake chaiyen]# nisdomainname
nistest
```

NIS 使用者端常駐行程 ypbind 使用名為/etc/yp.conf 的組態檔案，這指定使用者端聯接 binding 到 NIS 伺服器端 NIS 使用者端有三種方法來連接到 NIS 伺服器，我們可以在/etc/yp.conf 中輸入 ypserver 的 nisserverip。這告訴使用者來使用 ip 為 nisserverip 的伺服器。這告訴我們使用者端的 NIS 使用 nisserverip 為 192.168.0.5 的主要 NIS 伺服器。

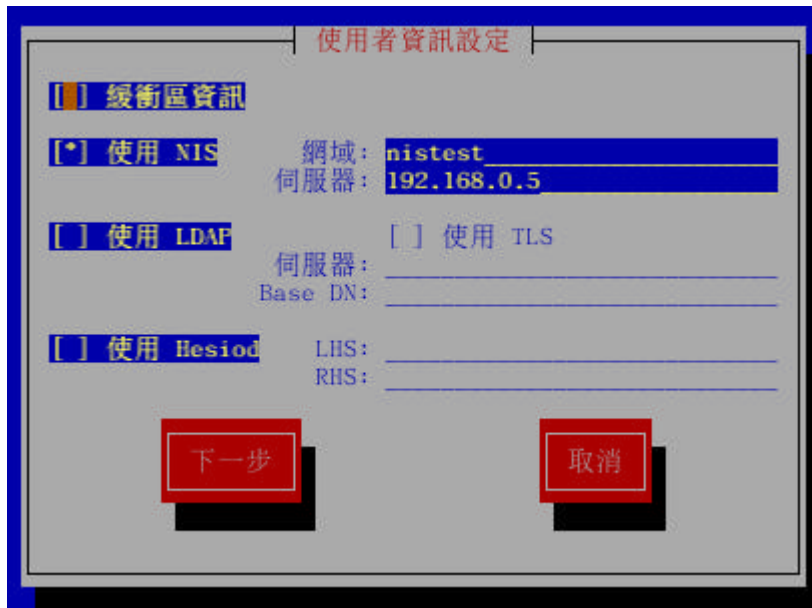
```
#vi /etc/yp.conf
```

```
ypserver 192.168.0.5
```

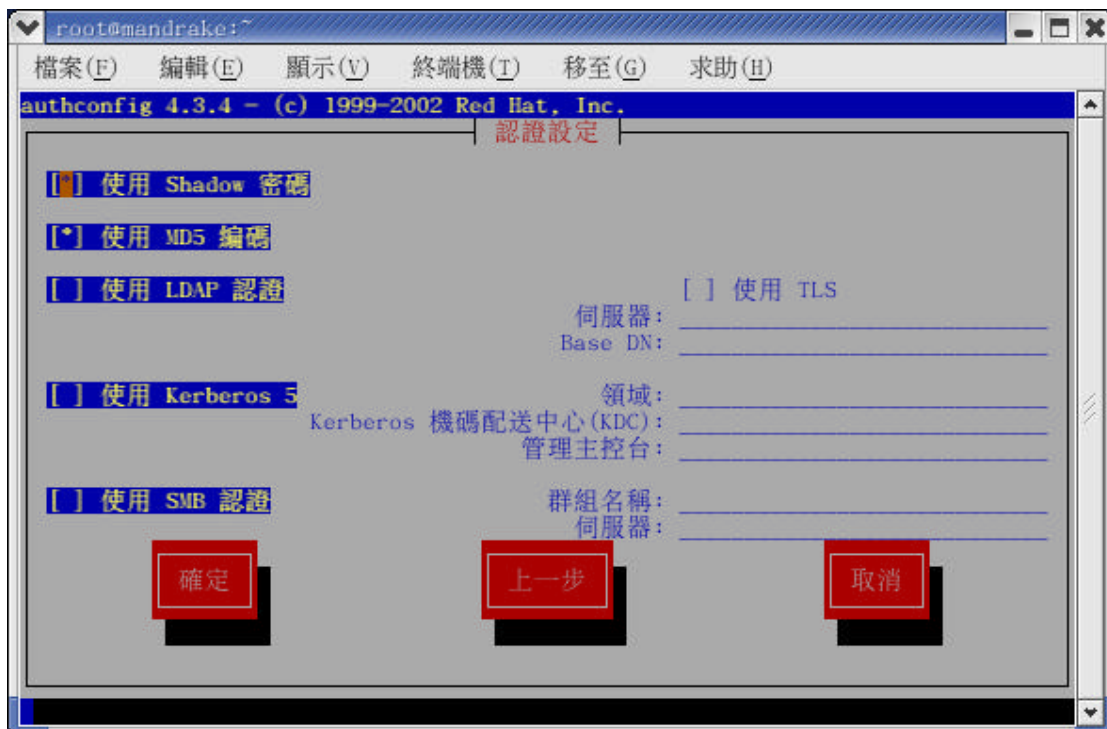
設定 NIS 使用者端，我們可以直接編輯/etc/yp.conf，或者使用 authconfig 工具。我們使用超級使用者身份 root 來執行 authconfig。

```
#authconfig
```

我們選取緩衝區資料，這樣可以加速快取，然後我們使用 Tab 鍵跳到使用 NIS，並且選取。我們設定網域為 nistest，而 NIS 伺服器為 192.168.0.5，伺服器是輸入 NIS 伺服器的 IP 位址。我們按下一步。



我們選取使用 Shadow 密碼和 MD5 編碼，然後使用 Tab 鍵到確定。



NIS 使用者端和 NIS 伺服器端一樣，它們都使用 RPC 遠端程序呼叫，所以在開始使用者端 NIS 的常駐行程 ypbind 前需先確定 portmapper 正在執行。我們啟動使用者端常駐行程，我們可以使用 ypbind 指令。

```
#/etc/rc.d/init.d/ypbind start
```

```
[root@mandrake home]# /etc/rc.d/init.d/ypbind start  
連結到 NIS 網域:[ 確定 ]  
傾聽 NIS 網域伺服器的回應。
```

在啟動完 NIS 使用者端常駐行程後，我們使用 rpcinfo -u localhost ypbind 來確定 ypbind 能夠在地圖登錄它自己的服務。

```
[root@mandrake home]# /usr/sbin/rpcinfo -u localhost ypbind  
程式 100007 的第 1 版已經就緒並等待服務中  
程式 100007 的第 2 版已經就緒並等待服務中
```

為了使用者在使用者端適當的查詢，不要增加在 NIS 使用者端的授權資訊。我們要在我們 NIS 使用者端的/etc/passwd 最後一行上寫入+::::::。

我們編輯/etc/host.conf 來讓它使用 NIS 來查詢主機。預定的 Linux 主機組態檔查詢像下列。order hosts,bind 我們改成 order hosts,nis,bind。

```
#vi /etc/host.conf  
order hosts,nis,bind
```

最後我們編輯/etc/nsswitch.conf 來增加標準 NIS 查詢。

```
#vi /etc/nsswitch.conf  
我們將下列像這樣的內容：
```

```
passwd:      files nisplus  
shadow:     files nisplus  
group:      files nisplus  
hosts:      files nisplus
```

改成像這樣的內容。

```
passwd:      files nis nisplus  
shadow:     files nis nisplus  
group:      files nis nisplus  
hosts:      files nis nisplus
```

我們使用 serviceconf 工具來確定在啟動作業系統時啟動 ypbind。

#serviceconf

我們選取 ypbind 然後選取檔案 儲存變更。當重新開機時，就會自動啟動我們的 ypbind 使用者端 NIS。

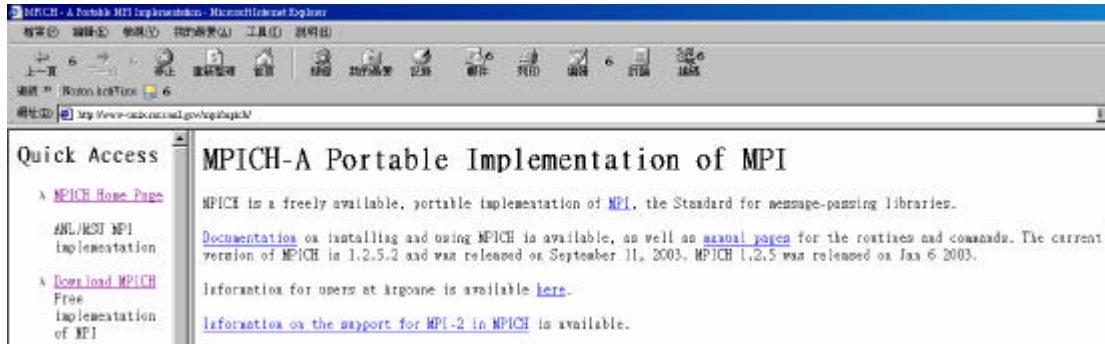


我們可以測試 NIS 組態，我們在 NIS 使用者端使用 ypcat 來顯示 NIS 伺服器端的密碼資料庫。

```
[root@mandrake /]# ypcat passwd
chaiyen2:!!:502:502::/home/chaiyen2:/bin/bash
wuchaiyen:$1$s/buXOTd$5qO2yT/D/HHd.ZcJvvD6Q.:503:503::/home/wuchaiyen:/bin/bash
chaiyen:$1$AbfmBICT$VqjvoPke1h.r8St.S6Qql0:500:500:chaiyen:/home/chaiyen:/bin/ba
sh
chaiyen1:$1$/2y2AxFS$NEta5B5oDL4vtCSexk1Uw.:501:501::/home/chaiyen1:/bin/bash
```


1-7 安裝平行函式庫

MPI 為平行運算的標準，而 mpich 為平行運算的軟體。下載訊息傳遞介面 MPI，mpich 為訊息傳遞函式庫它實作 MPI。下載網址 <http://www-unix.mcs.anl.gov/mpi/mpich/>



我們將 mpich 下載後，將它放到 /usr/local/src 的目錄下並且解壓縮。

```
#tar -zxvf mpich.tar.gz
```

我們進入 mpich 的目錄。

```
#cd mpich-1.2.5.2
```

我們組態 mpich，並且讓它支援叢集運算-device=ch_p4，並且將它的檔案放到 /usr/local/mpich 的目錄下。我們可以使用 configure --help 來觀看組態方法。

```
#!/configure --enable-debug -device=ch_p4 -fc=pgf77 -f90=pgf90  
--prefix=/usr/local/mpich
```

我們使用 make 來編譯 mpich。

```
#make
```

我們使用 make install 來安裝 mpich。

```
#make install
```

我們要編輯 /usr/local/mpich/share/machines.Linux，這裏定義了哪一些主機可以供我們作平行運算，其格式為”主機名稱：n”，n 代表該主機有 n 顆 CPU。我們在這裏只有兩台主機，一台是 flash，一台是 mandrake。主機名稱一定要是 master 可以查詢到的，也可以透過 /etc/hosts 或 DNS 來查到。

```
#cd /usr/local/mpich/share
```

```
#vi machines.LINUX
```

```
mandrake
```

```
flash
```


因為我們指定 test 作平行運算，如果使用者 test 使用 BASH，則我們要修改 /home/test/.bashrc。這樣就可以使用 mpich 軟體的指令直接執行平行運算。我們將 mpich 的路徑加入。

```
#vi /home/test/.bashrc
# .bashrc
# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
PATH=$PATH:/usr/local/mpich/bin
export PATH
MPI_HOME=/usr/local/mpich
MPI_ARCH=$MPI_HOME/bin/tarch
export MPI_ARCH MPI_HOME
```

這是我們使用 BASH，並將 mpich 路徑加入。

```
PATH=$PATH:/usr/local/mpich/bin
```

```
export PATH
```

```
MPI_HOME=/usr/local/mpich
```

```
MPI_ARCH=$MPI_HOME/bin/tarch
```

```
export MPI_ARCH MPI_HOME
```

如果執行的使用者是使用 csh 或 tcsh 指令來執行平行運算，則要將 mpich 的路徑加入下列的地方。

csh 和 tcsh 的環境變數設定 ~/.cshrc 或 ~/.tcshrc。

```
set path=($path/usr/local/mpich/bin)
```

```
setenv MPI_ARCH `usr/local/mpich/bin/tarch`
```

```
setenv MPI_HOME "/usr/local/mpich"
```

我們將 /usr/local/mpich/examples 的範例拿來執行。我們先編譯 c 的原始檔，然後使用 /usr/local/mpich/bin/mpirun -np 2 cpi 來執行平行運算，如果有出現下列的情況，則表示成功了。-np 2 是指定使用 2 顆 cpu 來作運算。Process 0 是 flash.aasir.com，而 Process 1 是 mandrake.aasir.com。

```
[test@flash test]$ cp -r /usr/local/mpich/examples/ .
[test@flash test]$ cd examples/
[test@flash examples]$ ls
cpi.c      cpip.c    Makefile  MPI-2-C++  README
cpilog.c  hello++.cc Makefile.in mpirun     simpleio.c
[test@flash examples]$ make cpi
/usr/local/mpich/bin/mpicc -c cpi.c
/usr/local/mpich/bin/mpicc -o cpi cpi.o -lm
[test@flash examples]$ /usr/local/mpich/bin/mpirun -np 2 cpi
Process 0 on flash.aasir.com
Process 1 on mandrake.aasir.com
pi is approximately 3.1416009869231241, Error is 0.0000083333333309
wall clock time = 0.031579
[test@flash examples]$
```

我們可以在/etc/man.config 來增加 mpich 的說明文件路徑。

```
#vi /etc/man.config
```

```
MANPATH /usr/local/mpich/man
```

我們可以使用 `man mpirun` 來觀看 `mpirun` 指令的使用方法。