Ubuntu Serverの新規インストール (2010/05/24)

2010年5月24日
10:09

この文書の内容： Ubuntu ServerをVMwareの仮想ハードディスクにインストールした手順のメモです．この作業は１台目の仮想マシンでおこないます．２台目，３台目の仮想マシンを追加するときは，ホストマシンのWindowsで仮想マシンのフォルダをまるごとコピーして，ホスト名を指定するなど調整を行うだけです．（参照：epiの新規ノード追加の作業）

用意するもの： インターネットにつながったWindowsのパソコン．VMware workstation または無料のVMware player. そして Ubuntu Server 10.04 LTS (64-bit)のCD image [http://www.ubuntu.com/getubuntu/download-server から ubuntu-10.04-server-amd64.iso をダウンロードしてください].

作業の概要： 研究室のPCクラスター(名前はepi）の各ノードの元になる仮想マシン（ホスト名はnekos1）をつくります．次のような設定にします．

＊ とりあえずネットワーク設定をしなくてもよいDHCPにする．クラスターを運用するときの準備として固定IPにするためのファイルも準備しておく．

＊ 相互にNFSで/usr/homeをexportする．マウントはautofsでオートマウント．

＊ Windowsのファイル共有(samba)で，各ユーザーのホームを読み書き可能にしてexportする．

＊ NISはつかわない．パスワードファイルは各ホストでローカルに持つ．必要に応じてコピーする．

＊ SSHはホストベース認証にしてクラスタ内は自由にssh (rsh)でログインできる．

＊ Xサーバーはインストールしない．クライアントはすこし入れておく．したがって，このサーバーとは別にUbuntuのデスクトップ版等をどこかにインストールするか，WindowsのX端末アプリケーション(ASTEC-Xなど）を用意することが前提です．

＊ 日本語環境やTeXはとりあえず整備しない．（＝＞デスクトップ版で！） なお，このサーバー版でも日本語はいちおう使えています．

＊ Rの並列実行をRmpi, snowパッケージでできるようにする．MPIはLAMにする．

以下作業の詳細です．VMware workstation 6および7で作業をしました．VMware playerでも同様にできると思います．実際にやった作業を多少アレンジして記録しておきます．相当試行錯誤してるので，このように一直線に進んだわけではないです．また研究室配布用の仮想マシン(epi20100519.zip)を作成したあとに修正した内容も反映させてあります．

＊ VMwareで新規仮想マシンを作成．ゲスト設定をUbuntu 64-bitにする．HD=20GB, CPU=2などしておく．ネットワークはブリッジ（これがデフォルト）．floppyとかUSBはいらないので外した．

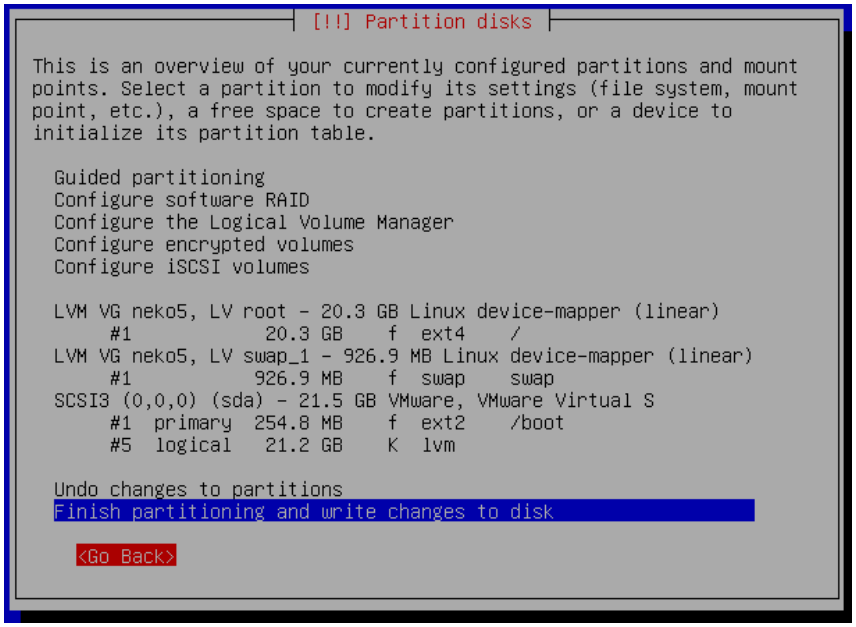
＊ubuntu-10.04-server-amd64.isoを仮想CDにいれる．（CD-Rに焼く必要なくて，すべてHD上のファイル操作だけ）

＊仮想マシンをパワーオンするとUbuntu Serverのインストール開始．

＊言語はEnglishのままにした．

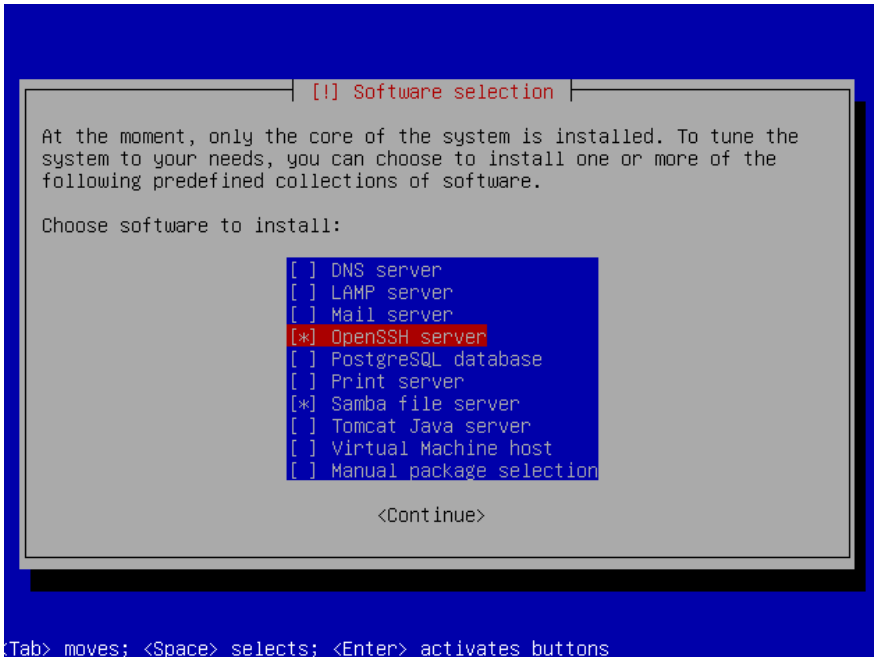
＊キーボード設定 USA, USAにした（英語キーボードつかっているから）．もし日本語キーボードをつかっているなら，そのようにしてください．

＊ひたすらデフォルトのまますすめる．HDのパーティションも全自動ですすめた．



画面の領域の取り込み日時:2010/05/15 21:46

＊Software selectionではOpenSSH, Sambaを選択



画面の領域の取り込み日時:2010/05/15 21:59

＊アカウントはkanriを作成．ここでは管理者用のアカウントだけつくる．個人利用のものはあとでつくる．

＊途中で，ホスト名を指定するときに，nekos1とした．別に何でも良い．

＊これでとりあえずOSのインストール終了．仮想マシンの再起動．

＊kanriアカウントでログイン．

＊ sudo apt-get update; sudo apt-get upgradeする．

＊ここでsudo shutdown -h nowしてパワーオフする．念のためVMwareでスナップショットを作成．ここまでの作業は２５分程度．

＊仮想マシンをパワーオン．kanriでログイン．ifconfig してipを確認．そのままVMwareのコンソールで作業をしてもいいが，他の端末ソフト等からログインして作業する．これでコピペができて作業が容易になる．

＊emacsはemacs22でなくてemacs23にしてみた．sudo apt-get install emacs23 => 63個のパッケージ144MBの展開イメージ

＊gnome-terminalいれてみた．sudo apt-get install gnome-terminal => 99個のパッケージ144MBの展開イメージ

＊並列計算のためLAMいれる．sudo apt-get install lam4-dev => 10個のパッケージ，47.7MBのイメージ

＊一応ドキュメントも．．．　sudo apt-get install lam-mpidoc => 1個，1MB

＊つぎにRをいれる準備．emacsとかで/etc/apt/sources.listを開き，最後の行に次を追加
deb http://cran.r-project.org/bin/linux/ubuntu lucid/

＊そして，kanriアカウントで次を実行
```
gpg --keyserver subkeys.pgp.net --recv-key E2A11821
gpg -a --export E2A11821 | sudo apt-key add -
```

＊sudo apt-get update　してから，つぎのようにRを入れる

sudo apt-get install r-base => 62個のパッケージ，176MBの展開イメージ
sudo apt-get install r-base-core-dbg => 1個，8MB
sudo apt-get install ess => 1個，1MBくらい

＊Rを起動して，次のようにパッケージを追加：
まず sudo R でRを起動する．そしてRの中から
install.packages(c("pvclust","scaleboot","snow"))
メニューが出る：Japan (Tsukuba)とか選ぶと，上記３個はあっさり入る．
install.packages("Rmpi")　を実行してRmpiをいれる．LAM用の設定でRmpiが入るはず．
q() としてＲを抜けたら，下記を実行する必要があるかも．
sudo rmdir /usr/local/lib/R/site-library/00LOCK

＊sambaの設定をここでやっておくとWindowsからファイルが簡単にコピーできて便利かもしれない．まずcd /etc/sambaとしておき，sudo cp smb.conf smb.conf-origを実行して，オリジナルの設定ファイルを保存しておく．そしてemacs等でsmb.confを開いて編集する．各ユーザーのホームを共有するために
```
;[homes]
;   comment = Home Directories
;   browseable = no
```
のセミコロンをとって
```
[homes]
   comment = Home Directories
   browseable = no
```
とする．そして書き込み可能にするために
```
;   read only = yes
```
を編集して
```
read only = no
```
とする．このほかに[printers]と[print$]のブロックはすべて行頭に;をつけてコメントアウトした．またworkgroupは
```
workgroup = SHIMODAIRA
```
とした．これでsmb.confの編集はおわり．仮想マシンをrebootするか，
sudo service smbd restart　とすれば，Windowsから共有フォルダが見れる．

＊ここでsudo shutdown -h nowしてパワーオフする．念のためVMwareでスナップショットを作成．ここまでの作業は１時間１５分程度．

これで必要最低限の状態になっている．

＊仮想マシンをパワーオン．kanriでログインして，ネットワーク関係の設定を行う．IPアドレスはDHCPから自動取得になっているはずだが，これを固定IPへ変える．ホスト名はepi00とする．

＊/etc/hostsをemacs等で開いて編集する．専攻や家庭のルーターなどがDHCPサーバーになっているとき，DHCPがIPを割り当てる範囲と重ならないところに，ホストのIPアドレスを自分で適当に割り当てていく．一部を抜粋すると，次のようにした．２行目のnekos1はコメントアウトした．

```
127.0.0.1       localhost
#127.0.1.1      nekos1
### shimodaira-lab epi cluster
192.168.40.140 epi00    epi00.is.titech.ac.jp
192.168.40.141 epi01    epi01.is.titech.ac.jp
192.168.40.142 epi02    epi02.is.titech.ac.jp
192.168.40.143 epi03    epi03.is.titech.ac.jp
```

＊/etc/hostnameを編集して，nekos1からepi00に変更．

＊DHCPの設定なので，/etc/network/interfacesをみると次のようになっているはず．

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

＊これを次のように変更して固定IP＝192.168.40.140にした．ただしaddress, netmask, gatewayは自分の環境にあわせて変える．自宅でルーターが動いていれば，IPは適当に決めて良くて，netmask=255.255.255.0，gateway=192.168.0.1などだと思う．

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.40.140
netmask 255.255.224.0
gateway 192.168.32.231
```

＊/etc/resolv.confの内容を確認しておく．DHCPで自動取得になっていると思う．念のためsudo cp /etc/resolv.conf /etc/resolv.conf-dhcpのように記録しておくと安心．

＊ここまでできたら，sudo rebootなどして，仮想マシンをリブート．kanriでログインする．ifconfigして，自分で指定した固定IPになっているか確認する．

＊クラスタ内で相互にパスワード無しでログインできるように，/etc/hosts.equivを作成して，信用できるホスト名をすべて並べる．epiクラスターでの設定をみると

```
totoro
kiki
epi00
epi01
epi02
epi03
```

．．．のようにした．totoroとkikiはクラスタの一部ではないが，ここにいれておくと totoro -> epi00へのログインはパスワードなしにできる．逆にtotoroではhosts.equivの設定は行わないので，epi00 -> totoroへのログインはパスワードが必要になる．

＊sshの設定を行う．/etc/sshd_configと/etc/ssh_configをemacs等で編集する．Ubuntuについてきたsshd_configをsshd_config-origにコピー，ssh_configをssh_config-origにコピーしておく．このsshd_config-origと私の設定したssh_configを比較すると次のようになっている．たとえばIgnoreRhosts yesだったのを変更してIgnoreRhosts noにしたということ．

```
kanri@epi00:/etc/ssh$ diff sshd_config-orig sshd_config
34c34
< IgnoreRhosts yes
---
> IgnoreRhosts no
38c38
< HostbasedAuthentication no
---
> HostbasedAuthentication yes
40c40
< #IgnoreUserKnownHosts yes
---
> IgnoreUserKnownHosts yes
47c47
< ChallengeResponseAuthentication no
---
> ChallengeResponseAuthentication yes
```

つぎにもともとあったssh_config-origと私の設定したssh_configを比較すると

```
kanri@epi00:/etc/ssh$ diff ssh_config-orig  ssh_config
52a53,56
>
> # for client (shimo 2010/05/16)
> HostbasedAuthentication yes
> EnableSSHKeysign yes
```

つまり，HostbasedAuthentication yesとEnableSSHKeysign yesを追加した．

＊あとで使うために次の二つのファイル/etc/ssh/make_host_key.shと/etc/ssh/make_known_hosts.shを作成して，sudo chmod +x /etc/ssh/make_host_key.shのようにして両方とも実行許可をだしておく．なおウェブにおいてあるパッチ(patch-20100524.tgz)を展開して，ディレクトリsshのなかをみれば，これらのシェルスクリプトが入れてあるから，コピーするだけでもよい．

```
---------- /etc/ssh/make_host_key.sh ----------------
#!/bin/sh

echo "Regenerate host keys"
echo "Also regenerate machine-id"
echo "WARNING: This command should NOT be run except for the first time"
echo "Press [enter] to proceed (Press Control-C to stop)"
read aaa

rm /etc/ssh/ssh_host_dsa_key*
rm /etc/ssh/ssh_host_rsa_key*
dpkg-reconfigure openssh-server

## make a unique id (used for authentication in gnome-terminal)
dbus-uuidgen > /var/lib/dbus/machine-id

---------- /etc/ssh/make_known_hosts.sh ----------------

#!/bin/sh

echo "Collect public keys from /etc/hosts.equiv"
echo "/etc/ssh/ssh_known_hosts will be updated"
echo "Please merge entries of power-downed-machines in backuped files"

#ssh-keyscan -f /etc/hosts.equiv > /etc/ssh/ssh_known_hosts
ssh-keyscan -f /etc/hosts.equiv > /etc/ssh/tmp_hosts
mv /etc/ssh/tmp_hosts /etc/ssh/ssh_known_hosts --backup=numbered
cp /etc/hosts.equiv /root/.shosts
-------------------------------------------------
```

これでsshの設定おわり．

＊一度shutdown -h nowして，スナップショットをとると安心．再びパワーオンしてkanriでログイン．

＊ nfsとautofs5をインストールする．

```
sudo apt-get install nfs-kernel-server
sudo apt-get install autofs5
```

＊ nfsの設定を行う．/etc/exportsを次のような内容で作成する．

```
/home epi00(rw,no_subtree_check)
/home epi01(rw,no_subtree_check)
/home epi02(rw,no_subtree_check)
/home epi03(rw,no_subtree_check)
```

．．．これで，/homeは，クラスタ内の他のホストへexportする．つまりクラスタ内のすべてのホストからepi00:/homeが読み書きできるようになる．

＊autofsの設定をおこなう．/etc/auto.masterを次の内容にする（#から始まる行はコメントなので無視）．

```
/var/autofs/nfs /etc/auto.nfs
```

次に/etc/auto.nfsは次のような内容にする．

```
epi00 -fstype=nfs epi00:/home
epi01 -fstype=nfs epi01:/home
epi02 -fstype=nfs epi02:/home
epi03 -fstype=nfs epi03:/home
```

．．．これで，epi01:/homeは，epi00からみると/var/autofs/nfs/epi01に自動的にマウントされる（そこを読もうとしたときにマウントされる）．ただし，これだとパスが長くて不便なので/nfsにシンボリックリンクを張っておく．

```
sudo mkdir /nfs
sudo ln -s /var/autofs/nfs/epi01 /nfs/epi01
```

これで/nfs/epi01をアクセスすると，内容はepi01:/homeになる．私は次のシェルスクリプトを作成して，sudoで実行した．

```
kanri@epi00:/etc$ cat make_nfs_link-epi.sh
#!/bin/sh
```

```
mkdir /nfs
for i in epi00 epi01 epi02 epi03 epi04 epi05 epi06 epi07 epi08 epi09 epi10 epi11 epi12 epi13 epi14 epi15 epi16
epi17 epi18 epi19 epi20 ; do
echo $i
ln -s /var/autofs/nfs/$i /nfs/$i
done
```

これでネットワーク関係の設定が終わり

＊やり残したことをやっておく.

＊いくつかおもいつきで追加パッケージ

```
sudo apt-get install ghostscript-x xterm kterm x11-apps
```

＊visudo でsudoersに次を追加すると，パスワードなしでshutdownできるようになる.

```
# anybody shutdown without password
ALL ALL=NOPASSWD: /sbin/shutdown
```

＊ここまででで一度shutdown -h nowして，スナップショット作成しておく.

＊パワーオンしてkanriでログイン. vmware-toolsのインストールするために，VMwareのメニューからvmware-toolsをインストールするを選択する.

＊kanriアカウントにてsudo mount /dev/cdrom /cdromなどとして，cdromをマウント. cd /cdromしてみると，VMwareTools-8.1.4-227600.tar.gzがあるので，tar xfzで展開する. vmware-tooks-distribというディレクトリができるので，sudo ./vmware-install.plなどとしてインストール. デフォルトのまま進めてOK.

＊シャットダウンしたあとで，VMwareの仮想マシンのファイル(vmxファイル）をメモ帳で開き， tools.syncTime = "FALSE" という行をtools.syncTime = "TRUE"になおす. これで仮想マシンの時計が遅れなくなる.

＊一度shutdown -h nowする. これでホストepi00は一応できたことになる.

＊他のホストを作るときの元になる仮想マシンを作成する. VMwareで完全クローンを作成して，できたクローンをパワーオンする. ここではクローンはepiという名前の仮想マシン名にしておく.

＊epiを起動すると，実際はepi00になっている. これをホスト名nekos1，IPはDHCPで自動取得の設定にもどす. つまり，/etc/hostsを編集して，２行目の

```
#127.0.1.1       nekos1
```

を

```
127.0.1.1       nekos1
```

にする. /etc/hostnameをnekos1にする. そしてcd /etc/networkして，sudo cp interfaces interfaces-epiとしておく. これで固定IPの設定（epi00用）が保存される. sudo emacs interfacesとして編集し，内容を元のDHCPのものに戻す. つまり/etc/network/interfacesは

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
```

になる. これで一度再起動して，ちゃんとnekos1，DHCPの設定になっているか確認する.

＊これでnekos1ができた. shutdown -h nowして終了. 配布するのはこの仮想マシン. zip形式で圧縮して固める.

これでおしまい.