“VMcluster” Stuffo

v0.2

Basic Instructions for setting up and using a “Vmcluster”

“VMcluster”

- Term selected to describe the virtual clusters we use in testing/demos.

- Currently using VMware™ Workstation-3.x

- Hosting (native) OS is Linux and the guest (virtual) machines also run Linux and OSCAR.

Terminology

- *Host* – native machine that has Vmware-Workstation installed upon it.
- *Guest* – virtual machine that is running on the native *host* machine.
- *Host-only* networking – setup for access only on the local host
- *Bridged/NAT* networking – setup for external access to other machines using the host's NIC.

Terminology & Naming

- *Host* – native machine that has Vmware-Workstation installed upon it.
  - *VMhost* – VMcluster host machine (running all the VM's)
  - Actual computer (e.g. 'sc-oscar', 'sc-oscarnode3')
- *Guest* – virtual machine that is running on the native *host* machine.
  - *VMheadnode* – VMcluster headnode (e.g. 'rh72vm')
  - *VMnode* – VMcluster compute node (e.g. 'oscarnode1')
- Networking
  - *Host-only* networking – setup for access only on the local host
  - *Bridged/NAT* networking – setup for external access to other machines using the host's NIC.

Host machine Setup (a)

- Install VMware Workstation RPM

```
rpm --ivh Vmware-workstation-3.1.1-1790.i386.rpm
```

- Run ‘vmware-config.pl’
  - Setup *host-only* networking (internal cluster net)
  - *Optionally setup bridged/NAT networking to enable external access
  - So, it’s easy to see how one could configure a headnode to have *eth0* and *eth1* like on standard headnodes.

Host Machine Setup (b)

- Create supporting directory structure for virtual machines, e.g. ‘/opt/vmware/’
  - *Note* these are full machines when installed therefore they will use a fair amt of disk space so plan accordingly.

- Export RPM directory to VMheadnode
  - Create ‘/etc/exports’ to allow VMheadnode to mount VMhost

Host Machine Setup (c)

- Create supporting directory structure for virtual machines, e.g. '/opt/vmware/'
  - *Note* these are full machines when installed therefore they will use a fair amt of disk space so plan accordingly.
- Created a user 'vmuser'
  - Add all "VMcluster" testers to this group
  - Set '/dev/vmnet*' to 'vmuser' group permissions
  - chown root.vmuser /dev/vmnet*
  - chmod g+rw /dev/vmnet*
- Export RPM directory to VMheadnode
  - Create '/etc/exports' to allow VMheadnode to mount VMhost

Copy All RPMs to VMhost

- Copy all the RPMs from the RedHat CD(s) to for example '/tftpboot/rpm'.
- NOTE, it might be helpful to use a more descriptive dir name, e.g. '/tftpboot/rh72_rpm' and use a symbolic link.

```
mkdir -p /tftpboot/rh72_rpm
ln -s /tftpboot/rh72_rpm  /tftpboot/rpm
mount /mnt/cdrom
cp -a  /mnt/cdrom/RedHat/RPMS/*.rpm \
   /tftpboot/rpm
eject /mnt/cdrom
```

Example '/etc/exports'

Create '/etc/exports' to allow VMheadnode to mount VMhost's directory of RPMs '/tftpboot/rpm'.

- Assuming VM's are in IP range 192.168.152.xxx

```
cat /etc/exports
/tftpboot/rh72_rpm  192.168.152.0/255.255.255.0(rw,no_root_squash)
```

This give all the VM's on this subnet read-write access to the host's (NFS server) directory of RPMs.

Starting VM's

- The 'vmware' service should be running

```
service vmware status
```

Output should display "is running…" lines if all is ok.

- Start vmware manually or add icons to menus

```
vmware &
```

or add menus/icons to explicitly start a vm.

```
/usr/bin/vmware -x /opt/vmware/linux-rh72/linux.cfg
```

DHCP Servers…

- The ‘vmware’ services include DHCP servers for these virtual nets.
  - Often when testing OSCAR you’ll **<u>not</u>** want these VMcluster DHCP’s to be answered by the VMhost’s servers.

Run the following on the VMhost to see these DHCP server

```
ps -A | grep vmnet-dhcpd
```

Run following to kill all the servers on the VMhost

```
killall vmnet-dhcpd
```

SC'02 "VMcluster" Configuration (a)

- Virtual IDE hard disks
  - Headnode with "*undo-able*" mode (~2.5GB)
  - Compute node with "persistent" mode (>1GB)

  NOTE: I've had issues with 'sfdisk' (used by OSCAR/SIS) seeing virtual SCSI disks on the compute nodes, use IDE.

- Memory
  - Headnode ~256MB
  - Compute nodes ~90MB (so you can run multiple vm's)
- Network
  - Ethernet: eth0 (*host-only*)

  NOTE: often I setup an eth0 & eth1 using NAT & host-only.

SC'02 "VMcluster" Configuration (b)

- Floppy drives!
  - No PXE in Phoenix BIOS on VMware
  - Only one floppy…multiple nodes? ☹

- UNIX is good! ☺
  - Create image file of floppy and set nodes to use a file instead of a real device. Typically I use a generic name like '*flpy-vmware.img*' and then symlink to whatever img I want.

```
cd /opt/vmware
dd if=/dev/fd0 of=flpy-oscar-1.4.img
ln -s flpy-oscar-1.4.img flpy-vmware.img
```

- Defaults to not attach to floppy.

SC'02 "VMcluster" Configuration (c)

- The VM's do not attach to the floppy by default.
  - If have boot floppy here, have to detach each reboot.

- Attach Floppy (to network boot node)!!!

  On startup **quickly** click

  Devices →

  floppy0 (not connected) →

  Connect

VMware files

- The VM node is comprised of a handful of files. Depending on the configuration these change but the basics are: (common defaults)

  Directory – I believe it's same as display name
  - linux.cfg – text file containing all node configs
  - linux.log – log file, format similar to syslog
  - linux.nvram – effectively a swap file (I think?)
  - linux.vmdk, linux-02.vmdk – virtual disks, chunked up into multiple files

    (IDE are2GB each I think).

Other Useful Stuffo!

- Tar up a cleanly built VMcluster & nodes after setup so you can always start fresh or even copy to other machines…do OS install once! ;)

- **[CTRL]+[ALT]** – return mouse to *host* machine

- Add the host machine to '/etc/hosts', e.g.
  - *nermal-1* (host-only vmnet for host named "nermal"),
  - *nermal-8* (bridged/nat vmnet for host named "nermal")

- Have the *host* machine NFS export the RPMs in 'tftpboot', then mount this export on the VMheadnode to save space!
  - I often add a sub-directory Updates for updates & misc. files, easy to share between host & guest machine.

Summary

- “VMclusters” are quite useful for testing:
  - Control changes (*undo-able* disks)
  - Run multiple machine/clusters on a laptop!

- A little care and attention needs to be taken when setting the environment up for ease of use (and sometime function…’sfdisk’ & SCSI).

Resources

VMware: http://www.vmware.com/

RedHat: http://www.redhat.com/

OSCAR: http://www.OpenClusterGroup/OSCAR

Thanks to other OSCAR developers for tips and suggestions when using “VMclusters”.