Recipe for getting/building/running Xen/ia64 with pv_ops

This recipe describes how to get xen-ia64 source and build it,
and run domU with pv_ops.

Requirements

- python
- mercurial
  it (aka "hg") is an open-source source code
  management software. See the below.
  http://www.selenic.com/mercurial/wiki/
- git
- bridge-utils

Getting and Building Xen and Dom0

My environment is;

```
Machine  : Tiger4
Domain0 OS  : RHEL5
DomainU OS  : RHEL5
```

1. Download source
   ```
   # hg clone http://xenbits.xensource.com/ext/ia64/xen-unstable.hg
   # cd xen-unstable.hg
   # hg clone http://xenbits.xensource.com/ext/ia64/linux-2.6.18-xen.hg
   ```

2. `# make world`

3. `# make install-tools`

4. copy kernels and xen
   ```
   # cp xen/xen.gz /boot/efi/efi/redhat/
   # cp build-linux-2.6.18-xen_ia64/vmlinux.gz \
     /boot/efi/efi/redhat/vmlinuz-2.6.18.8-xen
   ```

5. make initrd for Dom0/DomU
   ```
   # make -C linux-2.6.18-xen.hg ARCH=ia64 modules_install \
     O=$(/bin/pwd)/build-linux-2.6.18-xen_ia64
   # mkinitrd -f /boot/efi/efi/redhat/initrd-2.6.18.8-xen.img \
     2.6.18.8-xen --builtin mptspi --builtin mptbase \
     --builtin mptscsih --builtin uhci-hcd --builtin ohci-hcd \
     --builtin ehci-hcd
   ```

Making a disk image for guest OS

1. make file
   ```
   # dd if=/dev/zero of=/root/rhel5.img bs=1M seek=4096 count=0
   # mke2fs -F -j /root/rhel5.img
   ```

```
    # mount -o loop /root/rhel5.img /mnt
    # cp -ax /{dev,var,etc,usr,bin,sbin,lib} /mnt
    # mkdir /mnt/{root,proc,sys,home,tmp}

    Note: You may miss some device files. If so, please create them
    with mknod. Or you can use tar instead of cp.

 2. modify DomU's fstab
    # vi /mnt/etc/fstab
       /dev/xvda1  /            ext3    defaults        1 1
       none        /dev/pts     devpts  gid=5,mode=620  0 0
       none        /dev/shm     tmpfs   defaults        0 0
       none        /proc        proc    defaults        0 0
       none        /sys         sysfs   defaults        0 0

 3. modify inittab
    set runlevel to 3 to avoid X trying to start
    # vi /mnt/etc/inittab
       id:3:initdefault:
    Start a getty on the hvc0 console
       X0:2345:respawn:/sbin/mingetty hvc0
    tty1-6 mingetty can be commented out

 4. add hvc0 into /etc/securetty
    # vi /mnt/etc/securetty (add hvc0)

 5. umount
    # umount /mnt

FYI, virt-manager can also make a disk image for guest OS.
It's GUI tools and easy to make it.

==================
Boot Xen & Domain0
==================

 1. replace elilo
    elilo of RHEL5 can boot Xen and Dom0.
    If you use old elilo (e.g RHEL4), please download from the below
    http://elilo.sourceforge.net/cgi-bin/blosxom
    and copy into /boot/efi/efi/redhat/
    # cp elilo-3.6-ia64.efi /boot/efi/efi/redhat/elilo.efi

 2. modify elilo.conf (like the below)
    # vi /boot/efi/efi/redhat/elilo.conf
     prompt
     timeout=20
     default=xen
     relocatable

     image=vmlinuz-2.6.18.8-xen
             label=xen
             vmm=xen.gz
             initrd=initrd-2.6.18.8-xen.img
             read-only
             append=″ -- rhgb root=/dev/sda2″
```

The append options before "--" are for xen hypervisor,
the options after "--" are for dom0.

FYI, your machine may need console options like
"com1=19200,8n1 console=vga,com1". For example,

```
append="com1=19200,8n1 console=vga,com1 -- rhgb console=tty0 \
console=ttyS0 root=/dev/sda2"
```

Getting and Building domU with pv_ops

```
 1. get pv_ops tree
    # git clone
http://people.valinux.co.jp/~yamahata/xen-ia64/linux-2.6-xen-ia64.git/

 2. git branch (if necessary)
    # cd linux-2.6-xen-ia64/
    # git checkout -b your_branch origin/xen-ia64-domu-minimal-2008may19
    (Note: The current branch is xen-ia64-domu-minimal-2008may19.
    But you would find the new branch. You can see with
    "git branch -r" to get the branch lists.

http://people.valinux.co.jp/~yamahata/xen-ia64/for_eagl/linux-2.6-ia64-pv-ops.gi
t/
    is also available. The tree is based on
    git://git.kernel.org/pub/scm/linux/kernel/git/aegl/linux-2.6 test)


 3. copy .config for pv_ops of domU
    # cp arch/ia64/configs/xen_domu_wip_defconfig .config

 4. make kernel with pv_ops
    # make oldconfig
    # make

 5. install the kernel and initrd
    # cp vmlinux.gz /boot/efi/efi/redhat/vmlinuz-2.6-pv_ops-xenU
    # make modules_install
    # mkinitrd -f /boot/efi/efi/redhat/initrd-2.6-pv_ops-xenU.img \
      2.6.26-rc3xen-ia64-08941-g1b12161 --builtin mptspi \
      --builtin mptbase --builtin mptscsih --builtin uhci-hcd \
      --builtin ohci-hcd --builtin ehci-hcd
```

Boot DomainU with pv_ops

```
 1. make config of DomU
   # vi /etc/xen/rhel5
     kernel = "/boot/efi/efi/redhat/vmlinuz-2.6-pv_ops-xenU"
     ramdisk = "/boot/efi/efi/redhat/initrd-2.6-pv_ops-xenU.img"
     vcpus = 1
     memory = 512
     name = "rhel5"
     disk = [ ’file:/root/rhel5.img,xvda1,w’ ]
     root = ″/dev/xvda1 ro″
     extra= ″rhgb console=hvc0″
```

2. After boot xen and dom0, start xend

```
   # /etc/init.d/xend start
   ( In the debugging case, # XEND_DEBUG=1 xend trace_start )
```

3. start domU

```
   # xm create -c rhel5
```

=========
Reference
=========

- Wiki of Xen/IA64 upstream merge
  http://wiki.xensource.com/xenwiki/XenIA64/UpstreamMerge

Written by Akio Takebe <takebe_akio@jp.fujitsu.com> on 28 May 2008