

Distribution EOLE - Tâche #13864

Scénario # 13674 (Terminé (Sprint)): Valider le client Scribe pour Windows 10

Installation Windows 10 RTM (10240) sur QEMU/KVM, OpenNebula, LibVirt

11/03/2015 03:29 PM - Gilles Grandgérard

Status:	Fermé	Start date:	09/16/2015
Priority:	Normal	Due date:	
Assigned To:	Gilles Grandgérard	% Done:	100%
Target version:	Sprint 2015 45-47 - Équipe MENESR	Estimated time:	4.00 hours
		Spent time:	15.00 hours
Description			
Le but de cette tache est d'installation d'une Image Windows10 RTM dans OpenNebula			
1°) Test en local sur un PC			
Context: OS : Ubuntu 15.10 CPU : Intel(R) Core(TM) i3-3220T CPU @ 2.80GHz QEMU : QEMU emulator version 2.3.0 (Debian 1:2.3+dfsg-5ubuntu9), Copyright (c) 2003-2008 Fabrice Bellard			
Iso : Windows 10 RTM (Build 10240) French Virtio : virtio-win-0.1.110			
1.1) Configuration en ERREUR (Ecran BSOD !) La configuration avec cpu 'qemu64','kvm64' ne fonctionne pas !			
L'installation de Windows génère : <ul style="list-style-type: none">- une écran BSOD (Erreur : system thread exception not handled),- ou se fige sur le logo Windows (l'animation des points tournants ne s'affiche pas, puis la machine reboot)			
Cette ligne confirme http://www.linux-kvm.org/page/Guest_Support_Status#Windows_Family			
<pre>export WINVER=Windows10 export ISO=windows-10_finale_fr_431147_32.iso export ARCHI=i386 [! -f \$WINVER-\$ARCHI.img] && qemu-img create -f raw \$WINVER-\$ARCHI.img 25G qemu-system-x86_64 \ --enable-kvm \ -cpu qemu64,hv_relaxed,hv_spinlocks=0x1fff,hv_vapic,hv_time \ -smp cores=4 \ -soundhw hda \ -vga cirrus \ -usbdevice tablet \ -m 1536 \ -nodefaults \ -no-hpet \ -no-reboot \ -drive file=\$WINVER-\$ARCHI.img,if=virtio,format=raw,media=disk,cache=none \ -cdrom ../downloaded/\$ISO \ -drive file=../downloaded/virtio-win-0.1.110.iso,media=cdrom \ -net nic,model=virtio -net user \ -rtc base=localtime,clock=host \ -boot d \ -k fr</pre>			

1.2) Configuration OK :

Le test se fait avec le mode cpu = 'host'. Dans ce cas, l'installation se fait sans problème.

```
export WINVER=Windows10
export ISO=windows-10_finale_fr_431147_32.iso
export ARCHI=i386
[ ! -f $WINVER-$ARCHI.img ] && qemu-img create -f raw $WINVER-$ARCHI.img 25G
qemu-system-x86_64 \
  --enable-kvm \
  -cpu host,hv_relaxed,hv_spinlocks=0x1fff,hv_vapic,hv_time \
  -smp cores=4 \
  -soundhw hda \
  -vga cirrus \
  -usbdevice tablet \
  -m 1536 \
  -nodefaults \
  -no-hpet \
  -no-reboot \
  -drive file=$WINVER-$ARCHI.img,if=virtio,format=raw,media=disk,cache=none \
  -cdrom ../downloaded/$ISO \
  -drive file=../downloaded/virtio-win-0.1.110.iso,media=cdrom \
  -net nic,model=virtio -net user \
  -rtc base=localtime,clock=host \
  -boot d \
  -k fr
```

2°) Test sur un serveur OpenNebula/Hapy avec LibVirt

Context:

OS : Ubuntu 12.04.5 LTS (+ update)

CPU : Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz

QEMU : QEMU emulator version 2.0.0 (Debian 2.0.0+dfsg-2ubuntu1.6~cloud0), Copyright (c) 2003-2008 Fabrice Bellard

Iso : Windows 10 RTM (Build 10240) French (/var/lib/one//datastores/100/300/disk.1)

2.1) Configuration en Erreur :

La création d'un template dans OpenNebula sans ajout ne permet pas de lancer l'installation.

Propriétés du template OpenNebula (qui fonctionne avec une image Windows 8.1 !)

```
CPU="4"
DISK=[ IMAGE="DSK-40-G", IMAGE_UNAME="nebula", TARGET="hda" ]
DISK=[ DRIVER="raw", IMAGE="windows-10_finale_fr_431147_32.iso", IMAGE_UNAME="jenkins", TARGET="hdb" ]
FEATURES=[ ACPI="yes", LOCALTIME="yes", PAE="no" ]
GRAPHICS=[ KEYMAP="fr", LISTEN="0.0.0.0", TYPE="vnc" ]
INPUT=[ BUS="usb", TYPE="tablet" ]
LOGO="images/logos/windows8.png"
MEMORY="1536"
OS=[ ARCH="x86_64", BOOT="hd" ]
RAW=[ DATA="
<host><cpu><model fallback='allow'>core2duo</model></cpu></host>
<features><hyperv><relaxed state='on' /><vapic state='on' /><spinlocks state='on' retries='8191' /></hyperv></features>
<clock><timer name='hypervclock' present='yes' /></clock>
<video><model type='cirrus' heads='1'></model></video>
```

```
", TYPE="kvm"]
VCPU="2"
```

Le 'domain.xml' créer par la LibVirt est :

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <name>one-300</name>
  <vcpu>4</vcpu>
  <cputune>
    <shares>4096</shares>
  </cputune>
  <memory>2097152</memory>
  <os>
    <type arch='x86_64'>hvm</type>
    <boot dev='cdrom' />
    <boot dev='hd' />
  </os>
  <devices>
    <emulator>/usr/bin/kvm</emulator>
    <disk type='file' device='disk'>
      <source file='/var/lib/one//datastores/100/300/disk.0' />
      <target dev='vda' />
      <driver name='qemu' type='raw' cache='writeback' />
    </disk>
    <disk type='file' device='cdrom'>
      <source file='/var/lib/one//datastores/100/300/disk.1' />
      <target dev='hdb' />
      <readonly />
      <driver name='qemu' type='raw' cache='writeback' />
    </disk>
    <graphics type='vnc' listen='0.0.0.0' port='6200' keymap='fr' />
    <input type='tablet' bus='usb' />
  </devices>
  <clock offset='localtime' />

<host><cpu><model fallback='allow'>core2duo</model></cpu></host>
<features><hyperv><relaxed state='on' /><vapic state='on' /><spinlocks state='on' retries='8191' /></hyperv></features>
<clock><timer name='hypervclock' present='yes' /></clock>
<video><model type='cirrus' heads='1' /></model></video>

</domain>
```

export du domain.xml avec la commande **virsh domxml-to-native qemu-argv /var/lib/one//datastores/100/300/deployment.0**

```
QEMU_AUDIO_DRV=none qemu-system-x86_64 \
  --enable-kvm \
  -cpu qemu64,hv_relaxed,hv_spinlocks=0x1fff,hv_vapic,hv_time \
  -smp cores=4 \
  -vga cirrus \
  -usbdevice tablet \
  -m 1536 \
  -nodefaults \
  -no-hpet \
  -drive file=/var/lib/one//datastores/100/300/disk.0,if=virtio,format=raw,media=disk,cache=none \
  -drive file=/var/lib/one//datastores/100/300/disk.1,media=cdrom \
  -net nic,model=virtio -net user \
  -rtc base=localtime,clock=host \
```

```
-boot d \  
-k fr
```

Nous constatons que la commande ne gère pas le cpu 'core2duo' : La ligne de commande qemu est toujours avec 'cpu qemu64'. C'est un bug dans la LibVirt : il faut trouver un contournement car cela ne pourra jamais fonctionner.

2.2) Configuration OK :

Propriétés du template OpenNebula

```
CPU="4"  
DISK=[IMAGE="DSK-40-G",IMAGE_UNAME="nebula",TARGET="hda"]  
DISK=[DRIVER="raw",IMAGE="windows-10_finale_fr_431147_32.iso",IMAGE_UNAME="jenkins",TARGET="hdb"]  
FEATURES=[ACPI="yes",LOCALTIME="yes",PAE="no"]  
GRAPHICS=[KEYMAP="fr",LISTEN="0.0.0.0",TYPE="vnc"]  
INPUT=[BUS="usb",TYPE="tablet"]  
LOGO="images/logos/windows8.png"  
MEMORY="1536"  
OS=[ARCH="x86_64",BOOT="hd"]  
RAW=[DATA="  
<qemu:commandline>  
  <qemu:arg value='--cpu' />  
  <qemu:arg value='qemu64,+ssse3,+sse4.1,+sse4.2,+x2apic,+fpu,+vme,+de,+pse,+tsc,+msr,+pae,+mce,+  
cx8,+apic,+sep,+mtrr,+pge,+mca,+cmov,+pat,+pse36,+clflush,+mmx,+fxsr,+sse,+sse2,+ss,+syscall,+nx,+  
lm,+pni,+pclmulqdq,+vmx,+ssse3,+cx16,+pcid,+sse4_1,+sse4_2,+popcnt,+xsave,+avx,+lahf_lm,+smep,+erm  
s,+fsgsbase,+rdtscp,+f16c,check' />  
  <qemu:env name='QEMU_AUDIO_DRV' value='non' />  
</qemu:commandline>",>,TYPE="kvm"]  
VCPUs="2"  
  
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>  
  <name>one-300</name>  
  <vcpu>4</vcpu>  
  <cputune>  
    <shares>4096</shares>  
  </cputune>  
  <memory>2097152</memory>  
  <os>  
    <type arch='x86_64'>hvm</type>  
    <boot dev='cdrom' />  
    <boot dev='hd' />  
  </os>  
  <devices>  
    <emulator>/usr/bin/kvm</emulator>  
    <disk type='file' device='disk'>  
      <source file='/var/lib/one//datastores/100/300/disk.0' />  
      <target dev='vda' />  
      <driver name='qemu' type='raw' cache='writeback' />  
    </disk>  
    <disk type='file' device='cdrom'>  
      <source file='/var/lib/one//datastores/100/300/disk.1' />  
      <target dev='hdb' />  
      <readonly />  
      <driver name='qemu' type='raw' cache='writeback' />  
    </disk>  
    <graphics type='vnc' listen='0.0.0.0' port='6200' keymap='fr' />  
    <input type='tablet' bus='usb' />  
  </devices>  
  <clock offset='localtime' />  
  
  <qemu:commandline>  
    <qemu:arg value='--cpu' />
```

```

        <qemu:arg value='qemu64,+ssse3,+sse4.1,+sse4.2,+x2apic,+fpu,+vme,+de,+pse,+tsc,+msr,+
pae,+mce,+cx8,+apic,+sep,+mtrr,+pge,+mca,+cmov,+pat,+pse36,+clflush,+mmx,+fxsr,+sse,+sse2,+ss,+sys
call,+nx,+lm,+pni,+pclmulqdq,+vmx,+ssse3,+cx16,+pcid,+sse4_1,+sse4_2,+popcnt,+xsave,+avx,+lahf_lm,
+smep,+erms,+fsgsbase,+rdtscp,+f16c,check' />
        <qemu:env name='QEMU_AUDIO_DRV' value='non' />
    </qemu:commandline>
</domain>

```

export du domain.xml avec la commande **virsh domxml-to-native qemu-argv /var/lib/one//datastores/100/300/deployment.0**

```

QEMU_AUDIO_DRV=none qemu-system-x86_64 \
  --enable-kvm \
  -cpu qemu64,+ssse3,+sse4.1,+sse4.2,+x2apic,+fpu,+vme,+de,+pse,+tsc,+msr,+pae,+mce,+cx8,+apic,+s
ep,+mtrr,+pge,+mca,+cmov,+pat,+pse36,+clflush,+mmx,+fxsr,+sse,+sse2,+ss,+syscall,+nx,+lm,+pni,+pcl
mulqdq,+vmx,+ssse3,+cx16,+pcid,+sse4_1,+sse4_2,+popcnt,+xsave,+avx,+lahf_lm,+smep,+erms,+fsgsbase,
+rdtscp,+f16c,check \
  -smp cores=4 \
  -vga cirrus \
  -usbdevice tablet \
  -m 1536 \
  -nodefaults \
  -no-hpet \
  -drive file=/var/lib/one//datastores/100/300/disk.0,if=virtio,format=raw,media=disk,cache=none
\
  -drive file=/var/lib/one//datastores/100/300/disk.1,media=cdrom \
  -net nic,model=virtio -net user \
  -rtc base=localtime,clock=host \
  -boot d \
  -k fr

```

L'installation se fait normalement. (il faut ajouter les drivers Virtlo ...)

3°) Rappel des 'features' QEMU par défaut pour les CPU :

ATTENTION : les compilations de QEMU/distrib n'offre pas toutes les combinaisons ! A Vérifier avec la commande 'qem-system-x86_64 -cpu ?'

```

486 = +fpu,+vme,+pse
pentium = 486,+de,+tsc,+msr,+mce,+cx8,+mmx
pentium2 = pentium,+pae,+sep,+mtrr,+pge,+mca,+cmov,+pat,+pse36,+fxsr
pentium3 = pentium2,+sse
pentiumpro = ,+fpu,+de,+pse,+tsc,+msr,+pae,+mce,+cx8,+apic,+sep,+pge,+cmov,+pat,+mmx,+fxsr,+sse,+sse2
coreduo = pentiumpro,+vme,+mtrr,+mca,+clflush,+pni,+monitor,+nx
n270 = coreduo,+ssse3
core2duo = n270,+pse36,+syscall,+lm
qemu32 = pentiumpro,+pni
kvm32 = qemu32,+mtrr,+mca,+pse36,+clflush
cpu64-rhel5 = kvm32,+syscall,+nx,+lm
cpu64-rhel6 = cpu64-rhel5,+cx16,+lahf_lm
kvm64 = cpu64-rhel5,+cx16
qemu64 = kvm64,+svm <!-- ,+popcnt,+lahf_lm,+sse4a,+abm : IF These are supported only by TCG. KVM supports them only if the
host does. So we leave them out: -->
Conroe = pentiumpro,+mtrr,+mca,+pse36,+clflush,+pni,+ssse3,+syscall,+nx,+lm,+lahf_lm
Penryn = Conroe,+cx16,+sse4.1
Nehalem = Penryn,+sse4.2,+popcnt
Westmere = Nehalem,+aes
SandyBridge = Westmere,+pclmuldq,+x2apic,+tsc-deadline,+xsave,+avx,+rdtscp
Haswell = SandyBridge,+fma,+pcid,+movbe,+fsgsbase,+bmi1,+hle,+avx2,+smep,+bmi2,+erms,+invpcid,+rtm

```

History

#1 - 11/03/2015 03:29 PM - Gilles Grandg rard

- Status changed from *Nouveau* to *En cours*

#2 - 11/10/2015 09:50 AM - Scrum Master

- Status changed from *En cours* to *R solu*

#3 - 11/10/2015 04:44 PM - Jo l Cuissinat

- Remaining (hours) changed from 4.0 to 0.25

#4 - 11/12/2015 12:35 PM - Gilles Grandg rard

- Description updated

- Status changed from *R solu* to *En cours*

- Remaining (hours) changed from 0.25 to 0.0

#5 - 11/12/2015 02:34 PM - Gilles Grandg rard

- Subject changed from *Cr er une image Windows 10 dans Nebula* to *Installation Windows 10 RTM (10240) sur QEMU/KVM, OpenNebula, LibVirt*

#6 - 11/13/2015 10:51 AM - Gilles Grandg rard

- Status changed from *En cours* to *Ferm *

#7 - 11/19/2015 05:25 PM - Daniel Dehennin

- % Done changed from 0 to 100