Virtualisation

Professeur : Florent Gluck Assistant : Sebastien Chassot

March 5, 2022

QEMU/KVM First Steps

Introduction

Le but de ce travail pratique est de prendre en main QEMU/KVM, de comprendre son fonctionnement et de se familiariser avec les images au format qcow2.

Contrairement à VirtualBox, vous allez ici uniquement utiliser la ligne de commande. Afin de plus facilement gérer vos VMs et leurs configurations, vous organiserez votre travail (les différents exercices) à l'aide d'un `Makefile`, similairement à l'exemple ci-dessous :

```
X86=qemu-system-x86_64
ARM=qemu-system-arm
MEM=-m 1024
CDROM=-cdrom image.iso
OPTS=$(MEM) $(CDROM)

...

ex1:
    $(X86) $(OPTS)

exN:
    $(ARM) $(OPTS)
```

Exercice 1

Le but de cet exercice est d'installer une première VM avec QEMU depuis un live CD. Vous partirez de l'image ISO déjà téléchargée pendant le premier lab sur Virtualbox, soit l'image `xubuntu-20.04.3-desktop-amd64.iso`.

Partie 1A

Commencez par démarrez une VM avec `qemu-system-x86_64` et l'image ISO ci-dessus. N'installez pas Xubuntu dans la VM guest pour le moment. Cette VM doit posséder un lecteur CD-ROM/DVD-ROM contenant l'image ISO ci-dessus, 4GB de RAM, et 2 CPUs. N'utilisez pas d'options avancées de QEMU, seulement les arguments de base, à savoir :

- `-cdrom x` spécifie que le contenu du lecteur CD-ROM/DVD-ROM est le fichier `x`
- `-m x` spécifie `x` MB de RAM
- `-smp cpus=1` spécifie un CPU

Pour plus d'information sur la syntaxe de QEMU, exécutez `man qemu-system`.

Important : n'exécutez **jamais** QEMU en tant que `root` !. En plus d'être inutile, cela ouvre des failles de sécurité potentielles. De manière générale, assurez-vous de ne jamais exécuter de commandes en tant que `root` (ou `sudo`), à moins que cela soit absolument nécessaire.

- Combien de temps s'est écoulé jusqu'à arriver au choix de la langue dans le processus d'installation de Xubuntu ? Vous pouvez déterminer le temps en exécutant la commande `time` et en inspectant la valeur `real`. Par exemple, `time qemu-system-x86_64 ...` puis en pressant CTRL-C dans le terminal une fois le moment attendu arrivé).

Partie 1B

Réalisez la même mesure de temps qu'au point précédent, mais cette fois-ci ajoutez l'option `-enable-kvm` à QEMU.

- Quel est le nouveau temps obtenu ?
- Pourquoi l'exécution est-elle beaucoup plus rapide qu'au point précédent ?
- Plus précisément :
  - dans le cas présent (avec `-enable-kvm`), quel type de technique de virtualisation est utilisé par QEMU ?
  - dans le cas précédent (sans `-enable-kvm`), quel type de technique de virtualisation est utilisé par QEMU ?

Partie 1C

Commencez l'installation de Xubuntu dans votre VM.

- Est-il possible d'installer Xubuntu ?

Partie 1D

A l'aide de l'outil `qemu-img`, créez les deux images disques suivantes :

- une image `disk1.qcow` de 40GB au format qcow2
- une image `disk2.raw` de 500MB au format raw

Inspectez les deux fichiers images que vous venez de créer (l'outil `hexedit` permet d'inspecter et éditer le contenu de fichiers binaires).

- En quoi l'image `qcow2` diffère de l'image `raw` ? Décrivez en les différences majeures.
- Quelle est la taille du fichier de chaque image ? Pourquoi les tailles sont elles si différentes ?
- Quel est le contenu exact de `disk2.raw` ?

Partie 1E

Maintenant que vous avez créé deux disques, configurez votre VM pour qu'elle utilise `disk1.qcow` comme premier disque et `disk2.raw` comme deuxième disque. Un moyen simple pour spécifier le premier disque du contrôleur IDE est avec `-hda`, le deuxième avec `-hdb`, etc. Ceci dit, il est préférable d'utiliser l'argument `-drive` car il permet un contrôle plus fin, notamment en permettant de préciser le type d'image disque (nécessaire pour le format `raw`), le type de media, l'indice du disque, etc. Utilisez donc `-drive` pour spécifier les deux disques ci-dessus, en précisant pour chacun le format de l'image.

Utilisez le manuel de QEMU (`man qemu-system`) pour déterminer et comprendre la syntaxe à utiliser.

- Quel est la différence entre les arguments `-hdx` (où `x` représente le numéro de disque `a`, `b`, etc.) et `-drive` ?
- Comment peut-on écrire **exactement** l'équivalent de `-hdb` avec la syntaxe `-drive` ?

Démarrez une VM avec la configuration décrite ci-dessus pour les deux disques et installez Xubuntu sur le premier disque. Pour cela, au début de l'installation choisissez de faire un partionnement manuel des disques. Créez alors une partition primaire pour tout l'espace du premier disque et créez-y un système de fichiers ext4 dans lequel sera monté la racine du système de fichiers (`/`). Pour le deuxième disque, faites pareil mais choisissez d'y monter `/boot`.

Une fois l'installation terminée, rebootez le système et connectez-vous pour vérifier que tout fonctionne correctement (au cas où vous rencontreriez une erreur liée à l'installation de la langue, ignorez la).

- Sur l'hyperviseur (hôte), comparez les types des disques disk1 et disk2 (fichiers) utilisés dans l'OS guest? (commande `file`)

Inspectez les périphériques de votre VM avec la commande `lspci`.

- Voyez-vous une interface réseau ? A votre avis, de quel type d'interface s'agit-t-il : emulée ou paravirtualisée ?
- Avec VirtualBox la description de la VM se trouvait dans un fichier XML. Où se trouve la description de la VM dans le cas de QEMU ?
- Finalement, parvenez vous à réaliser un copier/coller de texte entre la VM et votre machine hôte ?

Exercice 2

Dans cet exercice, vous allez vous familiariser avec le QEMU Guest Agent.

Re-démarrez la VM précédente mais avec les options nécessaires à l'utilisation du QEMU Guest Agent.

Vérifiez que celui-ci s'exécute dans la VM et que vous arrivez bien à l'utiliser depuis la machine hôte.

Ensuite, écrivez un programme dans le langage de votre choix qui réalise les opérations suivantes :

1. Récupère le premier utilisateur du système
2. Créé un fichier "busted" dans le répertoire de l'utilisateur précédent, contenant un message de votre choix
3. Eteint la machine virtuelle (proprement)

Exercice 3

Le but de cet exercice est de manipuler la conversion d'images et de configurer les drivers de QEMU afin d'en améliorer les performances.

Partie 3A

On désire reprendre la VM `hepiadoom` utilisée dans le travail pratique précédent sur VirtualBox.

- Pouvez-vous exécuter directement la VM `hepiadoom` avec QEMU ?

Partie 3B

Convertissez le disque VirtualBox de la VM `hepiadoom` en une image au format qcow2, en vous assurant de spécifier une taille de cluster de 2MB (au lieu de la taille par défaut de 64KB).

Partie 3C

Exécutez une nouvelle VM en utilisant le disque converti au point précédent et mesurez le temps mis pour parvenir à l'écran de connexion.

- Quel temps avez-vous mesuré ?

Connectez-vous et déterminez quel est le nom du périphérique contenant le système de fichiers monté à la racine (`/`) du système. Pour rappel, sur Linux (et autres UNIXes) les périphériques se trouvent dans le répertoire `/dev`. Aussi, la commande `mount` affiche tous les systèmes de fichiers montés (dont celui monté à la racine). Alternativement, la commande `df -h` liste les systèmes de fichiers montés.

- De quel périphérique s'agit-il ?

Partie 3D

On désire améliorer les performances disque de la VM en utilisant des drivers paravirtualisés pour le contrôleur de disque dur. Exécutez la même VM qu'au point précédent, mais ajoutez l'option `if=virtio` à l'argument `-drive`, ce qui a pour effet d'utiliser le driver `virtio` qui est partavirtualisé.

Comme avant, mesurez temps mis pour parvenir à l'écran de connexion.

- Quel temps avez-vous mesuré ?

Tout comme pour la partie précédente, déterminez quel est le nom du périphérique contenant le système de fichiers monté à la racine (`/`) du système.

- De quel périphérique s'agit-il ?

Afin d'investiguer un peu plus ce que signifie ce nom de périphérique, ouvrez le journal des messages du kernel avec la commande :

```
journalctl -ka
```

Il est possible de faire des recherche dans ce journal avec la touche `/` (tout comme avec l'éditeur de texte `vim`).

- Que trouvez-vous si vous cherchez le mot-clé `virtio` ?
- Que voyez-vous également comme contrôleur de stockage si vous listez les périphériques PCI avec la commande `lspci` ?

Enfin, redémarrez la VM sans spécifier de driver paravirtualisé `virtio` pour le disque, connectez-vous et ouvrez le journal des messages du kernel.

- Que remarquez-vous si vous cherchez le mot-clé `virtio` ?
- Aussi, que remarquez-vous si vous listez les périphériques PCI avec la commande `lspci` ?

Exercice 4

Dans cet exercice, vous allez vous familiariser avec l'affichage, notamment via une connexion distante.

Partie 4A

Re-démarrez la VM de l'exercice précédent avec le driver de disque paravirtualisé et observez ensuite le type de carte graphique présent dans votre VM grâce à la commande `lspci`.

- Quelle est le nom de la carte graphique installée ? Pensez-vous qu'il s'agit d'une carte graphique émulée ou paravirtualisée ?

Partie 4B

Démarrez à nouveau la VM mais cette fois-ci en précisant le contrôleur graphique paravirtualisé `virtio`. L'argument `-vga` permet de spécifier le type de contrôleur graphique que QEMU doit utiliser.

- Quelle est le nom de la carte graphique installée ? Pensez-vous qu'il s'agit d'une carte graphique émulée ou paravirtualisée ?
- En inspectant les modules noyau, en voyez-vous liés à l'affichage paravirtualisé ?

Partie 4C : connexion distante avec Spice

Vous allez maintenant expérimenter avec l'affichage graphique distant de VMs grâce à Spice.

Démarrez la même VM que dans la partie précédente de sorte à ce qu'elle devienne un serveur Spice. Pour activer le serveur Spice dans QEMU, ajoutez les arguments suivants (en plus de ceux utilisés habituellement) :

```
-device virtio-serial-pci
-spice port=5930,disable-ticketing
-device virtserialport,chardev=spicechannel0,name=com.redhat.spice.0
-chardev spicevmc,id=spicechannel0,name=vdagent
```

Aussi, on désire améliorer les performances réseau de notre guest. Pour cela, spécifiez une interface réseau paravirtualisée avec l'argument :

```
-nic user,model=virtio-net-pci
```

Démarrez donc une VM où QEMU exécute un serveur spice et utilise une carte réseau paravirtualisée avec les arguments mentionnés ci-dessous.

Partie 4D

Sur une autre machine, par exemple la machine d'un camarade (qui doit se trouver sur le même réseau), utilisez le client `spicy` (ou `spice-client-gtk`) pour vous connecter à la VM s'exécutant sur votre machine (à noter que la combinaison de touches **shift+F12** permet de sortir du mode plein-écran).

Exécutez alors `chocolate-doom` à distance afin de voir si le jeu est jouable ou pas (il devrait l'être) !

- Que se passe-t-il, du point de vue de la VM, si on ferme la fenêtre du client Spice durant l'utilisation de la VM ?
- Parvenez vous à réaliser un copier/coller de texte entre la VM et votre machine hôte ?
- Le jeu Doom est-il jouable ?

Partie 4E

Dans cette dernière partie, vous allez expérimenter avec le *monitor* de QEMU.

Exécutez à nouveau QEMU en mode serveur Spice, mais en précisant en plus d'utiliser le *monitor* avec l'argument `-monitor stdio`. Le *monitor* est accessible dans la console où a été exécuté QEMU.

Parcourez les différentes commandes disponibles dans le *monitor* grâce à la commande `help`.

- Sur le serveur où la VM est en exécution, comment peut on interagir avec la VM sans utiliser de client Spice ?
- Utilisez le *monitor* pour inspecter diverses informations liées à votre VM, par exemple :

    - que est l'état de la VM ?
    - combien de CPUs comporte la VM ?
    - KVM est-il actif ?
    - afficher les informations liées à Spice
- Comment peut-on 1) éteindre et 2) faire un *reset* (forcé...) de la VM depuis le serveur sur laquelle la VM est en exécution (sans client Spice) ? Testez pratiquement chacune de vos réponse.