

Virtualisation

Professeur : Florent Gluck

March 1, 2023

Introduction à QEMU

Introduction

Le but de ce travail pratique est de prendre en main QEMU, de comprendre son fonctionnement, de l'utiliser dans diverses situations 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, je vous conseille fortement de réaliser des scripts pour chaque partie ou encore un **Makefile** (avec des cibles pour chaque partie).

1. Prise en main de QEMU

Le but ici est d'installer une première VM avec QEMU depuis un live CD. Partez depuis l'image ISO Xubuntu 22.04 déjà téléchargée pendant le premier lab sur Virtualbox.

Partie 1

Exécutez 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 listés ci-dessous :

- `-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 1 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, c'est dangereux car si le processus est compromis, alors il aura un accès complet à toute la machine. 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 2

Réalisez la même mesure de temps qu'au point précédent, mais cette fois-ci ajoutez les arguments `-enable-kvm` et `-cpu host` 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 ?

Commencez l'installation de Xubuntu dans votre VM.

- Est-il possible d'installer Xubuntu ?

Partie 3

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

- une image `disk1.qcow` de 500GB 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.
- Inspectez la taille apparente (via `ls -l`) de chaque image ainsi que la taille réelle. Que pouvez-vous en conclure ?
- Quel est le contenu de `disk2.raw` ?

Partie 4

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.

Lisez les slides de cours ou 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 partitionnement 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).

Dans l'OS guest, inspectez les périphériques de votre VM avec la commande `lspci`.

- Localisez l'interface réseau. Selon vous, s'agit-il d'une interface réseau émulée ou paravirtualisée ?

- Avec VirtualBox la configuration de la VM se trouvait dans un fichier XML. Où se trouve la configuration 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 ?

2. QEMU Guest Agent

Le but de cette partie est de vous familiariser avec le QEMU Guest Agent (QGA).

Re-démarrez la VM précédente mais avec les options nécessaires à l'utilisation du QEMU Guest Agent et vérifiez que celui-ci s'exécute dans la VM.

Comme le QEMU Guest Agent renvoie du JSON, il est recommandé d'installer et utiliser un formateur de JSON comme **aeson-pretty**. Celui-ci lit du JSON sur l'entrée standard (stdin) et le formate joliment sur la sortie standard (stdout).

En guise de premier essai, exécutez la commande **QGA guest-info** afin de déterminer quelles sont les commandes supportées. Ecrivez ensuite un programme dans le script/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 gracieusement la VM

3. Conversion d'images disque

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

Partie 1

On désire reprendre la VM **hepiadom** utilisée dans le travail pratique précédent sur VirtualBox.

- Pouvez-vous exécuter directement la VM **hepiadom** avec QEMU ?

Convertissez le disque VirtualBox de la VM **hepiadom** 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).

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 2

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 un contrôleur de disque **virtio** paravirtualisé.

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

- Quel gain de temps avez-vous obtenu par rapport au contrôleur de disque émulé ?

Vous remarquerez qu'avec ce nouveau contrôleur, le nom du périphérique a changé. Il s'appelle `/dev/vda` où le `v` signifie probablement *virtual*.

- Que voyez-vous également comme contrôleur de stockage si vous listez les périphériques PCI avec la commande `lspci` ?