Déployer l'ASA virtuel à l'aide de KVM

Vous pouvez déployer l'ASA virtuel sur n'importe quel périphérique CPU de *classe de serveur* x86 capable d'exécuter la machine virtuelle basée sur le noyau (KVM).

Important La mémoire minimale requise pour l'ASA virtuel est de 2 Go. Si votre ASA virtuel actuel fonctionne avec moins de 2 Go de mémoire, vous ne pouvez pas effectuer de mise à niveau vers la version 9.13(1)+ à partir d'une version antérieure sans augmenter la mémoire de votre machine ASA virtuel. Vous pouvez également redéployer une nouvelle machine ASA virtuel avec la dernière version.

Lignes directrices et limites relatives à la licence

Le matériel spécifique utilisé pour les déploiements d'ASA virtuel peut varier en fonction du nombre d'instances déployées et des exigences d'utilisation. Chaque appliance virtuelle que vous créez nécessite une allocation minimale de ressources (mémoire, nombre de CPU et espace disque) sur la machine hôte.

Important L'ASA virtuel est déployé avec une taille de stockage sur disque de 8 Go. Il est impossible de modifier l'allocation des ressources de l'espace disque.

Remarque À partir de la version 9.16.x d'ASA virtuel, lorsque vous rétrogradez d'ASAv100, dont la configuration de l'appareil est de 16 vCPU et de 32 Go de RAM, à ASAv10, vous devez configurer l'appareil avec 1 vCPU et 4 Go de RAM.

Passez en revue les lignes directrices et les limites suivantes avant de déployer l'ASA virtuel.

Configuration système requise pour ASA virtuel sur KVM

Assurez-vous de vous conformer aux caractéristiques ci-dessous pour assurer des performances optimales. Les conditions requises par l'ASA virtuel sont les suivantes :

- Le CPU de l'hôte doit être un processeur Intel ou AMD de *classe de serveur* x86 avec extension de virtualisation.

  Par exemple, les laboratoires de tests de performance d'ASA virtuel utilisent au minimum les éléments suivants : Cisco Unified Computing System (Cisco UCS®), serveur série C M4 avec les processeurs Intel Xeon® E5-2690v4 fonctionnant à 2,6 GHz.

vNIC recommandées

Les vNIC suivantes sont recommandées pour des performances optimales.

- i40e en mode Transmission directe PCI : dédié à la carte réseau physique du serveur à la machine virtuelle et transfère les données de paquets entre la carte réseau et la machine virtuelle par DMA (accès direct à la mémoire). Aucun cycle de CPU n'est nécessaire pour le déplacement des paquets.
- i40evf/ixgbe-vf : en fait, la même chose que ci-dessus (paquets DMA entre la carte réseau et la VM), mais permet de partager la carte réseau sur plusieurs VM. SR-IOV est généralement préféré, car il offre une plus grande souplesse de déploiement. Voir
- virtio : il s'agit d'un pilote de réseau para-virtualisé qui prend en charge le fonctionnement à 10 Gbit/s, mais nécessite également des cycles du CPU.

Remarque L'instance d'ASA virtuel s'exécutant sur le système KVM peut rencontrer des problèmes de connectivité des données avec l'interface SR-IOV en utilisant la version 2.17.4 du pilote vNIC i40e. Nous vous recommandons de mettre à niveau cette version de vNIC vers d'autres versions comme solution de contournement pour résoudre ce problème.

Optimisation des performances

Pour obtenir les meilleures performances avec ASA virtuel, vous pouvez apporter des ajustements à la machine virtuelle et à l'hôte. Consultez Réglage de la performance, à la page 9 pour de plus amples renseignements.

- **NUMA** : vous pouvez améliorer les performances d'ASA virtuel en isolant les ressources du CPU de la machine virtuelle invitée en un seul nœud d'accès à la mémoire non uniforme (NUMA). Consultez Lignes directrices NUMA, à la page 11 pour de plus amples renseignements.
- **Receive Side Scaling** (Dimensionnement côté réception) : l'ASA virtuel prend en charge Receive Côté Scaling (RSS), qui est une technologie utilisée par les adaptateurs réseau pour distribuer le trafic de réception réseau entre plusieurs cœurs de processeur. Consultez Files d'attente RX multiples pour le dimensionnement de la réception (RSS)., à la page 13 pour de plus amples renseignements.
- **VPN Optimization** (Optimisation du VPN) : consultez Optimisation du réseau privé virtuel (VPN), à la page 16 pour obtenir des considérations supplémentaires afin d'optimiser les performances du VPN avec l'ASA virtuel.

Mise en grappes

À partir de la version 9.17, la mise en grappe est prise en charge sur les instances d'ASA virtuel déployées sur KVM. Consultez la section Grappe ASA pour l'ASAv pour de plus amples renseignements.

Épinglage de CPU

L'épinglage de CPU est nécessaire pour que l'ASA virtuel fonctionne dans un environnement KVM; voir Activer l'épinglage du processeur (CPU), à la page 10.

Lignes directrices relatives au basculement pour la haute disponibilité

Pour les déploiements de basculement, assurez-vous que l'unité de secours a les mêmes droits de licence; par exemple, les deux unités doivent avoir le droit de 2 Gbit/s.

Important Lors de la création d'une paire à haute accessibilité à l'aide d'ASA virtuel, il est nécessaire d'ajouter les interfaces de données à chaque ASA virtuel dans le même ordre. Si exactement les mêmes interfaces sont ajoutées à chaque ASA virtuel, mais dans un ordre différent, des erreurs peuvent s'afficher à la console ASA virtuel. La fonctionnalité de basculement peut également être affectée.

ASA virtuel sur Proxmox VE

Proxmox Virtual Environment (VE) est une plateforme de virtualisation de serveurs à code source libre qui peut gérer les machines virtuelles KVM. Proxmox VE fournit également une interface de gestion web.

Lorsque vous déployez l'ASA virtuel sur Proxmox VE, vous devez configurer la VM pour qu'elle ait un port série émulé. Sans le port série, l'ASA virtuel entrera en boucle pendant le processus de démarrage. Toutes les tâches de gestion peuvent être effectuées à l'aide de l'interface de gestion web Proxmox VE.

Remarque Pour les utilisateurs avancés qui sont habitués au confort de l'interface Shell Unix ou de Windows Powershell, Proxmox VE fournit une interface de ligne de commande pour gérer tous les composants de votre environnement virtuel. Cette interface de ligne de commande dispose d'une fonction intelligente de complétion par tabulation et d'une documentation complète sous forme de pages de manuel UNIX.

Pour que l'ASA virtuel démarre correctement, la VM doit avoir un périphérique série configuré :

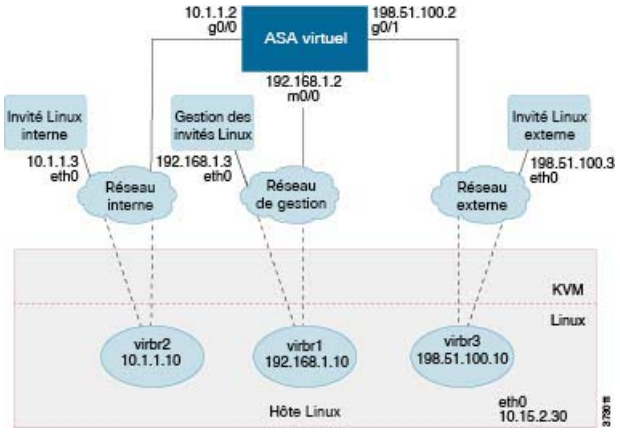
1. Dans le centre de gestion principal, sélectionnez la machine ASA virtuel dans l'arborescence de navigation de gauche.
2. Mettez la machine virtuelle hors tension.
3. Choisissez **Hardware (Matériel)** > **Add (Ajouter)** > **Network Device (Appareil réseau)** et ajoutez un port série.
4. Mettez l'ordinateur virtuel sous tension.
5. Accédez à la machine ASA virtuel à l'aide de Xterm.js.

Consultez la page Terminal série de Proxmox pour en savoir plus sur la configuration et l'activation du terminal sur l'invité ou le serveur.

Aperçu

La figure suivante montre un exemple de topologie de réseau avec ASA virtuel et KVM. Les procédures décrites dans ce chapitre reposent sur l'exemple de topologie. L'ASA virtuel agit comme pare-feu entre les réseaux interne et externe. Un réseau de gestion distinct est également configuré.

Illustration 1 : Exemple de déploiement d'ASA virtuel à l'aide de KVM



Prérequis

- Téléchargez le fichier qcow2 ASA virtuel à partir de Cisco.com et placez-le sur votre hôte Linux :

  http://www.cisco.com/go/asa-software

Remarque Une connexion à Cisco.com et un contrat de service Cisco sont requis.

- Aux fins de l'exemple de déploiement présenté dans ce document, nous supposons que vous utilisez Ubuntu 18.04 LTS. Installez les paquets suivants sur l'hôte Ubuntu 18.04 LTS :
  - qemu-kvm
  - libvirt-bin
  - bridge-utils
  - virt-manager
  - virtinst
  - virsh tools
  - genisoimage
- Les performances sont affectées par l'hôte et sa configuration. Vous pouvez maximiser le débit de l'ASA virtuel sur KVM en réglant votre hôte. Pour les concepts génériques de réglage d'hôte, consultez NFV fournit des performances de traitement des paquets avec Intel.

- Voici des optimisations utiles pour Ubuntu 18.04 :
  - macvtap : pont Linux à haute performance; vous pouvez utiliser macvtap au lieu d'un pont Linux. Notez que vous devez configurer des paramètres précis pour utiliser macvtap au lieu du pont Linux.
  - Transparent Huge Pages : augmente la taille des pages de mémoire et est activé par défaut dans Ubuntu 18.04.

    Hyperthread désactivé : réduit deux vCPU en un seul cœur.
  - txqueuelength : augmente la longueur de la file d'attente par défaut à 4 000 paquets et réduit le taux d'abandon.
  - épinglage : applique des processus qemu et vhost à des cœurs de CPU spécifiques; dans certaines conditions, l'épinglage augmente considérablement les performances.
- Pour en savoir plus sur l'optimisation d'une distribution basée sur RHEL, consultez le Guide de réglage et d'optimisation de la virtualisation Red Hat Enterprise Linux 7.
- Pour la compatibilité du logiciel ASA et des hyperviseurs ASA virtuel, consultez la section Compatibilité Cisco Cisco Secure Firewall ASA.

Préparer le fichier de configuration Day0 (Jour0)

Vous pouvez préparer un fichier de configuration Day0 (Jour0) avant de lancer l'ASA virtuel. Ce fichier est un fichier texte qui contient la configuration ASA virtuel appliquée lors du lancement de l'ASA virtuel. Cette configuration initiale est placée dans un fichier texte nommé « day0-config » dans un répertoire de travail que vous avez choisi, puis manipulée dans un fichier day0.iso qui est monté et lu lors du premier démarrage. Au minimum, le fichier de configuration Day0 (Jour0) doit contenir des commandes pour activer l'interface de gestion et configurer le serveur SSH pour l'authentification par clé publique, mais il peut également contenir une configuration ASA complète.

Le fichier day0.iso (votre fichier day0.iso personnalisé ou le fichier day0.iso par défaut) doit être disponible lors du premier démarrage :

- Pour autoriser automatiquement l'ASA virtuel lors du déploiement initial, placez le jeton d'identité de licence Smart (ID) que vous avez téléchargé à partir de Cisco Smart Software Manager dans un fichier texte nommé « idtoken » dans le même répertoire que le fichier de configuration Day0 (Jour0).
- Si vous souhaitez accéder à l'ASA virtuel à partir du **port série** de l'hyperviseur au lieu de la console virtuelle VGA, vous devez inclure le paramètre série de la console dans le fichier de configuration Day0 (Jour0) pour utiliser le port série lors du premier démarrage.
- Si vous souhaitez déployer l'ASA virtuel en mode transparent, vous devez utiliser un fichier de configuration ASA connu en cours d'exécution en mode transparent comme fichier de configuration Day0 (Jour0). Cela ne s'applique pas à un fichier de configuration Day0 (Jour0) pour un pare-feu routé.

Remarque Nous utilisons Linux dans cet exemple, mais il existe des utilitaires similaires pour Windows.

Procédure

Étape 1 Saisissez la configuration de l'interface de ligne de commande pour l'ASA virtuel dans un fichier texte appelé « day0-config ». Ajoutez des configurations d'interface pour les trois interfaces et toute autre configuration de votre choix.

La première ligne doit commencer par la version de l'ASA. La configuration day0-config doit être une configuration ASA valide. La meilleure façon de générer la configuration day0 (jour0) est de copier les parties pertinentes d'une configuration en cours d'exécution à partir d'un ASA ou d'un ASA virtuel existant. L'ordre des lignes dans la configuration day0 (jour0) est important et doit correspondre à l'ordre observé dans une sortie de commande **show running-config** existante.

Exemple :

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

Étape 2 (Facultatif) Pour l'octroi de licences automatisé lors du déploiement initial d'ASA virtuel, assurez-vous que les renseignements suivants se trouvent dans le fichier day0-config :

- Adresse IP de l'interface de gestion
- (Facultatif) Mandataire HTTP à utiliser pour les licences Smart
- Une commande de **route** qui active la connectivité au mandataire HTTP (si précisé) ou à tools.cisco.com
- Un serveur DNS qui corrige tools.cisco.com en adresse IP
- La configuration de licence Smart précise la licence ASA virtuel que vous demandez
- (Facultatif) Un nom de domaine unique pour que l'ASA virtuel soit plus facile à trouver dans CSSM

Étape 3 (Facultatif) Téléchargez le fichier de jeton d'identité de la licence Smart émis par Cisco Smart Software Manager sur votre ordinateur, copiez le jeton d'ID à partir du fichier téléchargé et placez-le dans un fichier texte nommé « idtoken » qui contient uniquement le jeton d'ID.

Étape 4 Générez le CD-ROM virtuel en convertissant le fichier texte en fichier ISO :

Exemple :

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

Le jeton d'identité enregistre automatiquement l'ASA virtuel sur le serveur de licences Smart.

Étape 5 Répétez les étapes 1 à 5 pour créer des fichiers de configuration par défaut distincts avec les adresses IP appropriées pour chaque ASA virtuel que vous souhaitez déployer.

Préparer les fichiers XML de pont virtuel

Vous devez configurer des réseaux virtuels qui connectent les invités ASA virtuel à l'hôte KVM et qui connectent les invités entre eux.

Remarque Cette procédure n'établit pas la connectivité au monde externe en dehors de l'hôte KVM.

Préparez les fichiers XML de pont virtuel sur l'hôte KVM. Pour l'exemple de topologie de réseau virtuel décrit dans Préparer le fichier de configuration Day0 (Jour0), à la page 5, vous avez besoin des trois fichiers de pont virtuel suivants : virbr1.xml, virbr2.xml et virbr3.xml (vous devez utiliser ces trois noms de fichiers; par exemple, virbr0 n'est pas autorisé, car il existe déjà). Chaque fichier contient les renseignements nécessaires pour configurer les ponts virtuels. Vous devez attribuer au pont virtuel un nom et une adresse MAC uniques. L'attribution d'une adresse IP est facultative.

Procédure

Étape 1 Créez trois fichiers XML de pont de réseau virtuel. Par exemple, virbr1.xml, virbr2.xml et virbr3.xml :

Exemple :

```
<network>
<name>virbr1</name>
<bridge name='virbr1' stp='on' delay='0' />
<mac address='52:54:00:05:6e:00' />
<ip address='192.168.1.10' netmask='255.255.255.0' />
</network>
```

Exemple :

```
<network>
<name>virbr2</name>
```

```
<bridge name='virbr2' stp='on' delay='0' />
<mac address='52:54:00:05:6e:01' />
<ip address='10.1.1.10' netmask='255.255.255.0' />
</network>
```

Exemple :

```
<network>
<name>virbr3</name>
<bridge name='virbr3' stp='on' delay='0' />
<mac address='52:54:00:05:6e:02' />
<ip address='198.51.100.10' netmask='255.255.255.0' />
</network>
```

Étape 2 Créez un script qui contient les éléments suivants (dans notre exemple, nous nommons le script virt_network_setup.sh) :

```
virsh net-create virbr1.xml
virsh net-create virbr2.xml
virsh net-create virbr3.xml
```

Étape 3 Exécutez ce script pour configurer le réseau virtuel. Le script affiche les réseaux virtuels. Les réseaux restent opérationnels tant que l'hôte KVM est en cours d'exécution.

```
stack@user-ubuntu:-/KvmAsa$ virt_network_setup.sh
```

Remarque
Si vous rechargez l'hôte Linux, vous devez réexécuter le script virt_network_setup.sh. Il ne persiste pas pendant les redémarrages.

Étape 4 Vérifiez que les réseaux virtuels ont été créés :

```
stack@user-ubuntu:-/KvmAsa$ brctl show
bridge name bridge id STP enabled Interfaces
virbr0 8000.0000000000000 yes
virbr1 8000.5254000056eed yes virb1-nic
virbr2 8000.5254000056eee yes virb2-nic
virbr3 8000.5254000056eec yes virb3-nic
stack@user-ubuntu:-/KvmAsa$
```

Étape 5 Affichez l'adresse IP attribuée au pont virbr1. Voici l'adresse IP que vous avez attribuée dans le fichier XML.

```
stack@user-ubuntu:-/KvmAsa$ ip address show virbr1
S: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
link/ether 52:54:00:05:6e:00 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.10/24 brd 192.168.1.255 scope global virbr1
valid_lft forever preferred_lft forever
```

Déployer l'ASA virtuel

Utilisez un script de déploiement basé sur virt-install pour lancer l'ASA virtuel.

Procédure

Étape 1 Créez un script virt-install appelé « virt_install_asav.sh ».

Le nom de la machine ASA virtuel doit être unique pour toutes les autres VM sur cet hôte KVM.

L'ASA virtuel prend en charge jusqu'à 10 réseaux. Cet exemple utilise trois réseaux. L'ordre des clauses de pont de réseau est important. La première répertoriée est toujours l'interface de gestion de l'ASA virtuel (Management 0/0), la deuxième répertoriée est GigabitEthernet 0/0 de l'ASA virtuel et la troisième répertoriée est GigabitEthernet 0/1 de l'ASA virtuel et ainsi de suite jusqu'à GigabitEthernet 0/8. La carte réseau virtuelle doit être Virtio.

Exemple :

```
virt-install \
--connect=qemu:///system \
--network network=default,model=virtio \
--network network=default,model=virtio \
--network network=default,model=virtio \
--name=asav \
--cpu host \
--arch=x86_64 \
--machine=pc-1.0 \
--vcpus=1 \
--ram=2048 \
--os-type=linux \
--virt-type=kvm \
--import \
--disk path=/home/kvmperf/Images/desmo.qcow2,format=qcow2,device=disk,bus=virtio,cache=none \
--disk path=/home/kvmperf/asav_day0.iso,format=iso,device=cdrom \
--console pty,target_type=virtio \
--serial tcp,host=127.0.0.1:4554,mode=bind,protocol=telnet
```

Étape 2 Exécutez le script virt_install :

Exemple :

```
stack@user-ubuntu:-/KvmAsa$ ./virt_install_asav.sh

Starting install...
Creating domain...
```

Une fenêtre apparaît, affichant la console de la VM. Vous pouvez voir que la VM démarre. Le démarrage de la VM prend quelques minutes. Une fois que la VM arrête de démarrer, vous pouvez exécuter des commandes de l'interface de ligne de commande à partir de l'écran de la console.

Réglage de la performance

Amélioration des performances pour les configurations KVM

Vous pouvez augmenter les performances d'un ASA virtuel dans l'environnement KVM en modifiant les paramètres sur l'hôte KVM. Ces paramètres sont indépendants des paramètres de configuration du serveur hôte. Cette option est disponible dans Red Hat Enterprise Linux 7.0 KVM.

Vous pouvez améliorer les performances des configurations KVM en activant l'épinglage CPU.

Activer l'épinglage du processeur (CPU)

ASA virtuel exige que vous utilisiez l'option d'affinité CPU KVM pour améliorer la performance de l'ASA virtuel dans les environnements KVM. L'affinité du processeur, ou l'épinglage CPU, permet de lier et de dissocier un processus ou un fil à une unité de traitement centrale (CPU) ou à une plage de CPU, de sorte que le processus ou le fil ne s'exécutera que sur l'unité ou les unités de traitement centrale (CPU) désignées plutôt que sur n'importe quelle CPU.

Configurez les associations d'hôtes pour déployer les instances qui utilisent l'épinglage CPU sur différents hôtes des instances qui ne le font pas, afin d'éviter que les instances non épinglées utilisent les exigences en matière de ressource des instances épinglées.

Attention Ne déployez pas des instances avec la topologie NIMA sur les mêmes hôtes que les instances qui n'ont pas la topologie NIMA.

Pour utiliser cette option, configurez l'épinglage CPU sur l'hôte KVM.

Procédure

Étape 1 Dans l'environnement d'hôte KVM, vérifiez la topologie de l'hôte pour trouver le nombre de vCPU disponibles pour l'épinglage :

Exemple :

```
virsh nodeinfo
```

Étape 2 Vérifiez les numéros de vCPU disponibles :

Exemple :

```
virsh capabilities
```

Étape 3 Épinglez les vCPU aux ensembles de cœurs de processeur :

Exemple :

```
virsh vcpupin <vm-name> <vcpu-number> <host-core-number>
```

La commande **virsh vcupinin** doit être exécutée pour chaque vCPU sur votre ASA virtuel. L'exemple suivant montre les commandes KVM nécessaires si vous avez une configuration ASA virtuel avec quatre vCPU et que l'hôte a huit cœurs :

```
virsh vcpupin asav 0 2
virsh vcpupin asav 1 3
virsh vcpupin asav 2 4
virsh vcpupin asav 3 5
```

Le nombre de cœurs de l'hôte peut être n'importe quel nombre entre 0 et 7. Pour plus d'informations, consultez la documentation KVM.

Remarque
Lors de la configuration de l'épinglage CPU, tenez compte de la topologie CPU du serveur hôte. Si vous utilisez un serveur configuré avec plusieurs cœurs, ne configurez pas l'épinglage CPU sur plusieurs connecteurs.

L'inconvénient de l'amélioration des performances de la configuration KVM est qu'elle nécessite des ressources système dédiées.

Lignes directrices NUMA

L'accès mémoire non uniforme (NUMA, Non-Uniform Memory Access) est une architecture de mémoire partagée qui décrit le placement des modules de mémoire principaux en ce qui concerne les processeurs dans un système multiprocesseur. Lorsqu'un processeur accède à la mémoire qui ne se trouve pas dans son propre nœud (mémoire distante), les données doivent être transférées sur la connexion NUMA à un débit plus lent qu'il ne le serait lors de l'accès à la mémoire locale.

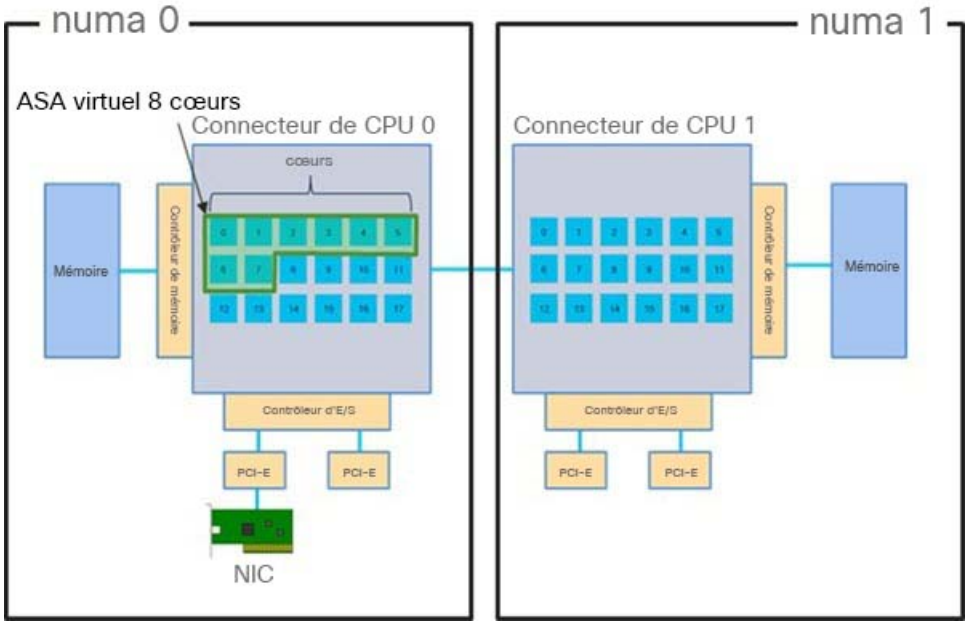
L'architecture du serveur x86 est composée de plusieurs connecteurs et de plusieurs cœurs dans un connecteur. Chaque connecteur CPU, ainsi que sa mémoire et son entrée-sortie, est appelé nœud NUMA. Pour lire efficacement les paquets à partir de la mémoire, les applications invitées et les périphériques associés (comme la carte d'interface réseau) doivent résider sur le même nœud.

Pour une performance ASA virtuel optimale :

- La machine ASA virtuel doit fonctionner sur un seul nœud numa. Si un seul ASA virtuel est déployé de sorte qu'il fonctionne sur deux connecteurs, la performance sera considérablement réduite.
- Un ASA virtuel à 8 cœurs (Illustration 2 : Exemple d'architecture NUMA ASA virtuel à 8 cœurs , à la page 12) exige que chaque connecteur du CPU hôte ait au moins 8 cœurs par connecteur. Il faut tenir compte des autres machines virtuelles en cours d'exécution sur le serveur.
- Un ASA virtuel à 16 cœurs (Illustration 3 : Exemple d'architecture NUMA ASA virtuel à 16 cœurs, à la page 12) exige que chaque connecteur du CPU hôte ait un minimum de 16 cœurs par connecteur. Il faut tenir compte des autres machines virtuelles en cours d'exécution sur le serveur.
- La carte réseau (NIC) doit se trouver sur le même nœud NUMA que la machine ASA virtuel.

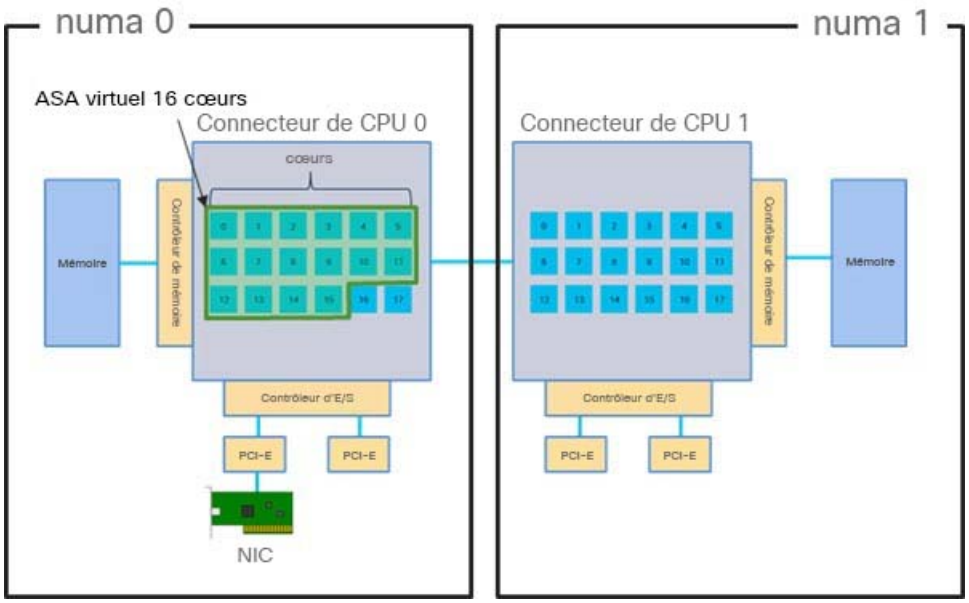
La figure suivante montre un serveur avec deux connecteurs pour CPU, chaque CPU ayant 18 cœurs. L'ASA virtuel à 8 cœurs exige que chaque connecteur du CPU hôte ait au moins 8 cœurs.

Illustration 2 : Exemple d'architecture NUMA ASA virtuel à 8 cœurs



La figure suivante montre un serveur avec deux connecteurs pour CPU, chaque CPU ayant 18 cœurs. L'ASA virtuel à 16 cœurs exige que chaque connecteur du CPU hôte ait au moins 16 cœurs.

Illustration 3 : Exemple d'architecture NUMA ASA virtuel à 16 cœurs



Optimisation NUMA

Idéalement, la machine ASA virtuel devrait fonctionner sur le même nœud numa que celui des cartes réseau (NIC). Voici la marche à suivre :

1. Déterminez sur quel nœud se trouvent les cartes réseau en utilisant « lstopo » pour afficher un diagramme des nœuds. Localisez les cartes réseau et notez le nœud auquel elles sont connectées.
2. Sur l'hôte KVM, utilisez `virsh list` pour trouver l'ASA virtuel.
3. Modifier la machine virtuelle en utilisant : `virsh edit <Numéro de machine virtuelle>`.
4. Alignez ASA virtuel sur le nœud choisi. Les exemples suivants supposent des nœuds à 18 cœurs.

   Aligner sur le nœud 0 :

   ```
   <vcpu placement='static' cpuset='0-17'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='0'/>
   </numatune>
   ```

   Aligner sur le nœud 1 :

   ```
   <vcpu placement='static' cpuset='18-35'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='1'/>
   </numatune>
   ```

5. Enregistrez la modification de .xml et redémarrez la machine ASA virtuel.
6. Pour vous assurer que votre machine virtuelle fonctionne sur le nœud souhaité, effectuez un `ps aux | grep <nom de votre machine virtuelle ASAv>` pour obtenir l'ID de processus.
7. Exécutez `sudo numastat -c <ID de processus de la machine virtuelle ASAv>` pour voir si la machine ASA virtuel est correctement alignée.

Vous trouverez plus d'informations sur l'utilisation de la mise au point de NUMA avec KVM dans le document RedHat 9.3. libvirt NUMA Tuning.

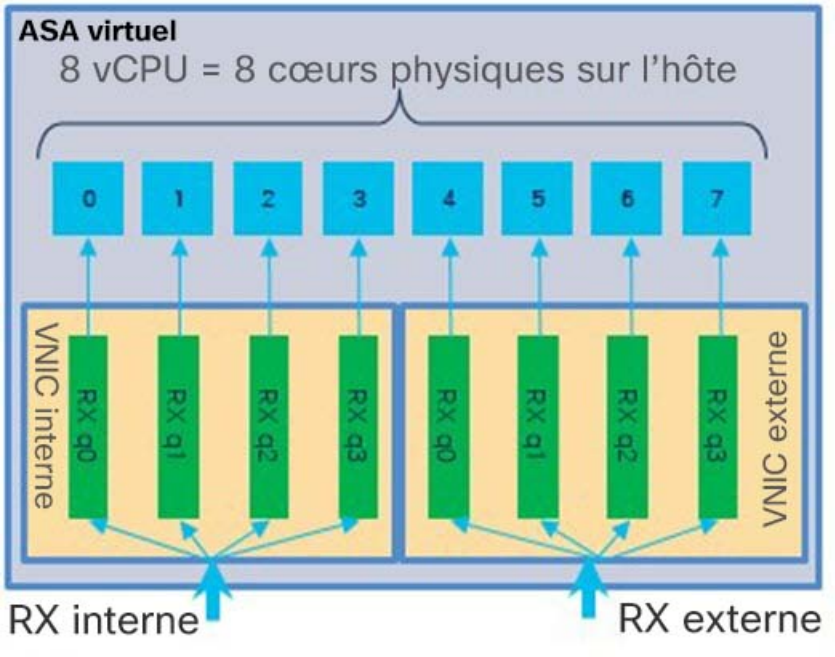
Files d'attente RX multiples pour le dimensionnement de la réception (RSS).

L'ASA virtuel prend en charge le dimensionnement de la réception (RSS), qui est une technologie utilisée par les adaptateurs réseau pour distribuer le trafic de réception réseau entre plusieurs cœurs de processeur. Pour un débit maximal, chaque vCPU (cœur) doit avoir sa propre file d'attente RX NIC. Notez qu'un déploiement typique de réseau privé virtuel (VPN) d'accès à distance peut utiliser une seule paire d'interfaces interne/externe.

Important La version 9.13(1) ou ultérieure d'ASA virtuel est nécessaire pour utiliser plusieurs files d'attente RX. Pour KVM, la version *libvirt* doit être au moins 1.0.6.

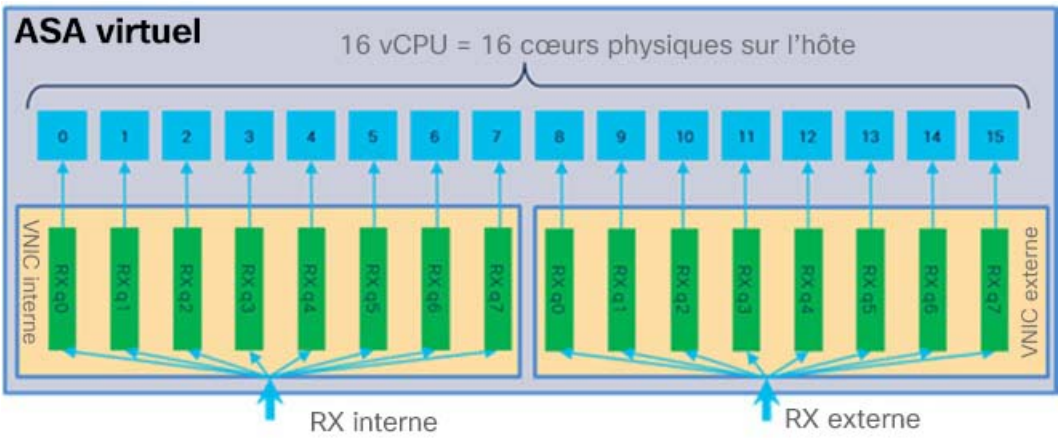
Pour une machine virtuelle à 8 cœurs avec une paire d'interfaces interne/externe, chaque interface aura 4 files d'attente RX, comme indiqué dans Illustration 4 : Files d'attente RX RSS à 8 cœurs ASA virtuel , à la page 14.

Illustration 4 : Files d'attente RX RSS à 8 cœurs ASA virtuel



Pour une machine virtuelle à 16 cœurs avec une paire d'interfaces interne/externe, chaque interface aura 8 files d'attente RX, comme indiqué dans Illustration 5 : Files d'attente RX RSS à 16 cœurs ASA virtuel, à la page 14.

Illustration 5 : Files d'attente RX RSS à 16 cœurs ASA virtuel



Le tableau suivant présente les vNIC de l'ASA virtuelpour KVM et le nombre de files d'attente RX prises en charge. Consultez vNIC recommandées, à la page 2 pour obtenir des descriptions des vNIC prises en charge.

Tableau 1 : NIC/vNIC recommandées par KVM

Carte NIC	Pilote vNIC	Technologie de pilote	Nombre de files d'attente RX	Rendement
x710	i40e	PCI Passthrough (transmission directe de PCI)	8 maximum	Les modes PCI Passthrough et SR-IOV pour le x710 offrent les meilleures performances. SR-IOV est généralement préféré pour les déploiements virtuels, car la carte d'interface réseau (NIC) peut être partagée sur plusieurs machines virtuelles.
	i40evf	SR-IOV	8	
x520	ixgbe	PCI Passthrough (transmission directe de PCI)	6	La carte réseau x520 affiche des performances de 10 à 30 % inférieures à celles de la x710. Les modes PCI Passthrough et SR-IOV pour le x520 offrent des performances similaires. SR-IOV est généralement préféré pour les déploiements virtuels, car la carte d'interface réseau (NIC) peut être partagée sur plusieurs machines virtuelles.
	ixgbe-vf	SR-IOV	2	
S. O.	virtio	Paravirtualisation	8 maximum	Non recommandé pour ASAv100. Pour d'autres déploiements, consultez Activer la prise en charge multifile d'attente pour Virtio sur KVM, à la page 15.

Activer la prise en charge multifile d'attente pour Virtio sur KVM

L'exemple suivant montre comment configurer le nombre de files d'attente RX de la carte réseau Virtio à 4 à l'aide de virsh pour modifier le xml de libvirt :

```
<interface type='bridge'>
 <mac address='52:54:00:43:6e:3f'/>
 <source bridge='clients'/>
 <model type='virtio'/>
 <driver name='vhost' queues='4'/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</interface>
```

Important La version *libvirt* doit être au moins 1.0.6 pour prendre en charge plusieurs files d'attente RX.

Optimisation du réseau privé virtuel (VPN)

Voici quelques considérations supplémentaires pour optimiser les performances du VPN avec l'ASA virtuel.

- IPSec a un débit plus élevé que DTLS.
- Chiffrement : GCM a environ deux fois le débit de CBC.

Provisionnement d'interface SR-IOV

SR-IOV permet à plusieurs machines virtuelles de partager un seul adaptateur réseau PCIe à l'intérieur d'un hôte. SR-IOV définit ces fonctions :

- Fonction physique (PF) : les PF sont des fonctions PCIe complètes qui comprennent les capacités SR-IOV. Celles-ci s'affichent comme des cartes réseau (NIC) statiques normales sur le serveur hôte.
- Fonction virtuelle (VF) : les VF sont des fonctions PCIe allégées qui aident au transfert de données. Une VF est dérivée d'une PF et gérée par l'intermédiaire d'une PF.

Les VF peuvent fournir une connectivité allant jusqu'à 10 Gbit/s à la machine ASA virtuel dans une structure de système d'exploitation virtualisé. Cette section explique comment configurer les VF dans un environnement KVM. La prise en charge de SR-IOV sur l'ASA virtuel est expliquée dans ASA virtuel et provisionnement de l'interface SR-IOV.

Sur ASAv5 et ASAv10, le pilote VMXNET3 est fortement recommandé pour une performance optimale. De plus, l'interface SR-IOV, lorsqu'elle est utilisée en combinaison (mélange d'interfaces), améliore la performance du réseau avec ASA virtuel, en particulier avec l'allocation de plus de cœurs et de ressources pour le processeur (CPU).

Exigences pour le provisionnement de l'interface SR-IOV

Si vous avez une carte réseau physique qui prend en charge SR-IOV, vous pouvez associer des VF compatibles avec SR-IOV ou des cartes réseau virtuelles (vNIC) à l'instance ASA virtuel. SR-IOV nécessite également une prise en charge dans le BIOS ainsi que dans l'instance de système d'exploitation ou l'hyperviseur qui s'exécute sur le matériel. Vous trouverez ci-dessous une liste de lignes directrices générales pour le provisionnement de l'interface SR-IOV pour l'exécution d'ASA virtuel dans un environnement KVM :

- Vous avez besoin d'une carte d'interface réseau physique compatible avec SR-IOV dans le serveur hôte; voir Lignes directrices et limites des interfaces SR-IOV.
- Vous devez activer la virtualisation dans le BIOS sur votre serveur d'hôte. Consultez la documentation de votre prestataire pour en savoir plus.
- Vous avez besoin d'activer la prise en charge globale d'IAMMU pour SR-IOV dans le BIOS sur votre serveur d'hôte. Consultez la documentation de votre prestataire de matériel pour en savoir plus.
- L'ASA virtuel sur KVM à l'aide de l'interface SR-IOV prend en charge le mélange de types d'interfaces. Vous pouvez utiliser SR-IOV ou VMXNET3 pour l'interface de gestion et SR-IOV pour l'interface de données.

Modifier le BIOS et le système d'exploitation de l'hôte KVM

Cette section montre diverses étapes d'installation et de configuration pour le provisionnement des interfaces SR-IOV sur un système KVM. Les renseignements figurant dans cette section ont été créés à partir de

périphériques dans un environnement de laboratoire spécifique, en utilisant Ubuntu 14.04 sur un serveur Cisco série UCS C avec un adaptateur de serveur Ethernet Intel X520 - DA2.

Avant de commencer

- Assurez-vous d'avoir une carte d'interface de réseau (NIC) compatible avec SR-IOV.
- Assurez-vous que les fonctionnalités des technologies de virtualisation Intel (VT-x) et VT-d sont activées.

Remarque Certains fabricants de systèmes désactivent ces extensions par défaut. Nous vous recommandons de vérifier le processus avec la documentation du fournisseur, car différents systèmes ont des méthodes différentes pour accéder aux paramètres BIOS et les modifier.

- Assurez-vous que tous les modules, toutes les bibliothèques, tous les outils utilisateur et toutes les utilitaires Linux ont été installés lors de l'installation du système d'exploitation; voir Prérequis, à la page 4.
- Assurez-vous que l'interface physique est à l'état UP. Vérifiez avec ifconfig <*ethname*>.

Procédure

Étape 1 Connectez-vous à votre système à l'aide du compte utilisateur et du mot de passe « root » (racine).

Étape 2 Vérifiez que Intel VT-d est activé.

Exemple :

```
kvmuser@kvm-host:/$ dmesg | grep -e DMAR -e IOMMU
[    0.000000] ACPI: DMAR 0x000000006F9A4C68 000140 (v01 Cisco0 CiscoUCS 00000001 INTL 20091013)
[    0.000000] DMAR: IOMMU enabled
```

La dernière ligne indique que VT-d est activé.

Étape 3 Activez Intel VT-d dans le noyau en ajoutant le paramètre *intel_iommu=on* à l'entrée GRUB_CMDLINE_LINUX dans le fichier de configuration */etc/default/grub*.

Exemple :

```
# vi /etc/default/grub
...
GRUB_CMDLINE_LINUX="nofb splash=quiet console=tty0 ... intel_iommu=on"
...
```

Remarque
Si vous utilisez un processeur AMD, ajoutez plutôt *amd_iommu=on* aux paramètres de démarrage.

Étape 4 Redémarrez le serveur pour que la modification iommu prenne effet.

Exemple :

```
> shutdown -r now
```

Étape 5 Créez des VF en écrivant une valeur appropriée au paramètre *sriov_numvfs* via l'interface *sysfs* en utilisant le format suivant :

```
#echo n > /sys/class/net/device name/device/sriov_numvfs
```

Pour vous assurer que le nombre souhaité de VF est créé à chaque cycle d'alimentation du serveur, vous devez ajouter la commande ci-dessus au fichier *rc.local*, qui se trouve dans le répertoire */etc/rc.d/*. Le système d'exploitation Linux exécute le script *rc.local* à la fin du processus de démarrage.

Par exemple, les éléments suivants montrent la création d'une VF par port. Les interfaces pour votre configuration particulière varient.

Exemple :

```
echo '1' > /sys/class/net/eth4/device/sriov_numvfs
echo '1' > /sys/class/net/eth5/device/sriov_numvfs
echo '1' > /sys/class/net/eth6/device/sriov_numvfs
echo '1' > /sys/class/net/eth7/device/sriov_numvfs
```

Étape 6 Redémarrez le serveur.

Exemple :

```
> shutdown -r now
```

Étape 7 Vérifiez que les VF ont été créées à l'aide de *lspci*.

Exemple :

```
 > lspci | grep -i "Virtual Function"
kvmuser@kvm-racetrack:~$ lspci | grep -i "Virtual Function"
0a:10.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.1 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.2 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.3 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
```

Remarque
Vous verrez des interfaces supplémentaires à l'aide de la commande **ifconfig**.

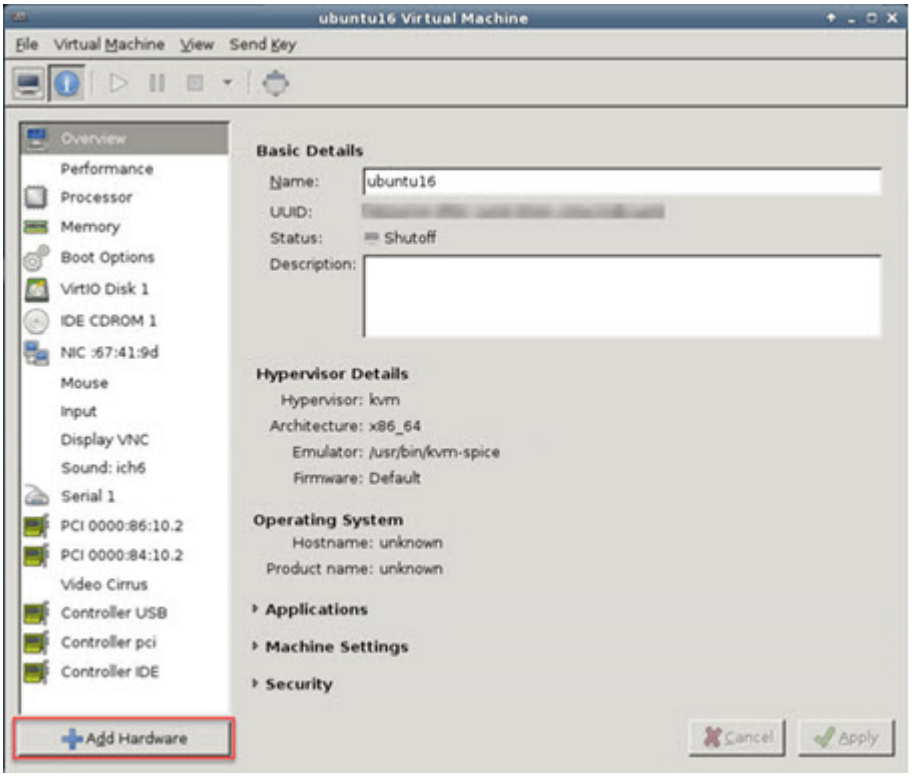
Affecter des périphériques PCI à l'ASA virtuel

Une fois que vous avez créé des VF, vous pouvez les ajouter à l'ASA virtuel comme vous ajouteriez n'importe quel périphérique PCI. L'exemple suivant explique comment ajouter un contrôleur VF Ethernet à un ASA virtuel à l'aide de l'outil graphique **virt-manager**.

Procédure

Étape 1 Ouvrez l'ASA virtuel et cliquez sur le bouton **Add Hardware** (Ajouter du matériel) pour ajouter un nouveau périphérique à la machine virtuelle.

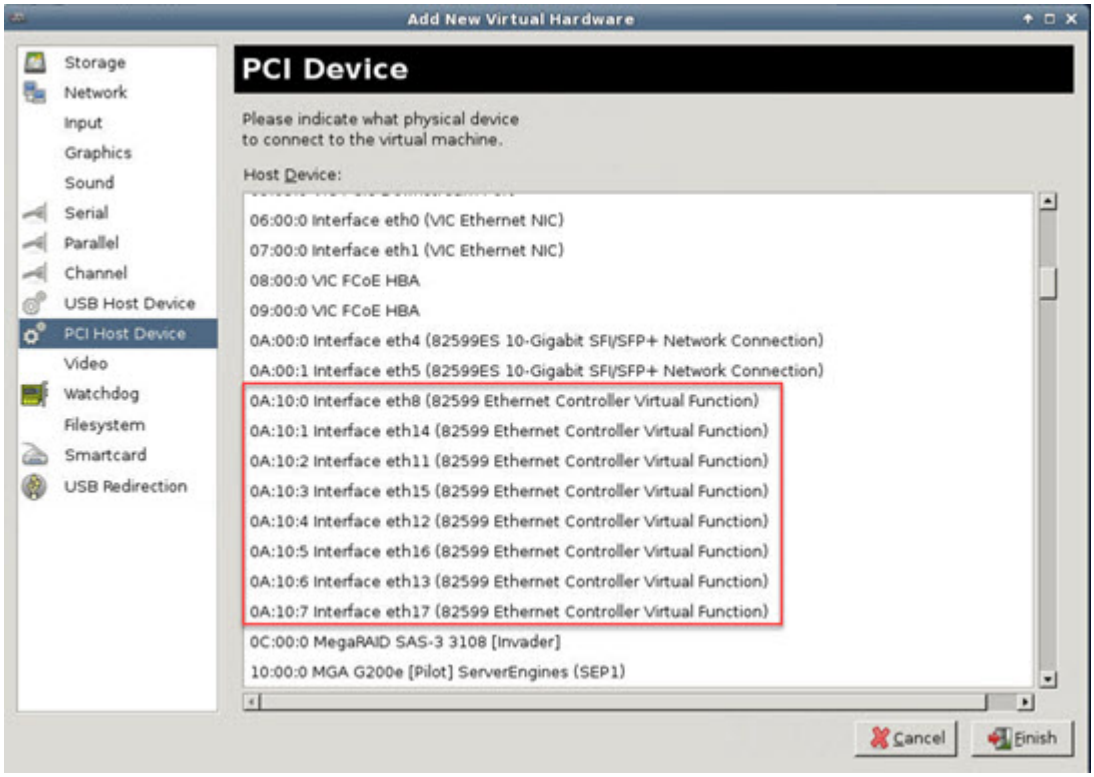
Illustration 6 : Ajouter du matériel



Étape 2 Cliquez sur **PCI Host Device** (Périphérique hôte PCI) dans la liste **Hardware** (Matériel) du volet gauche.

La liste des périphériques PCI, y compris les VF, apparaît dans le volet central.

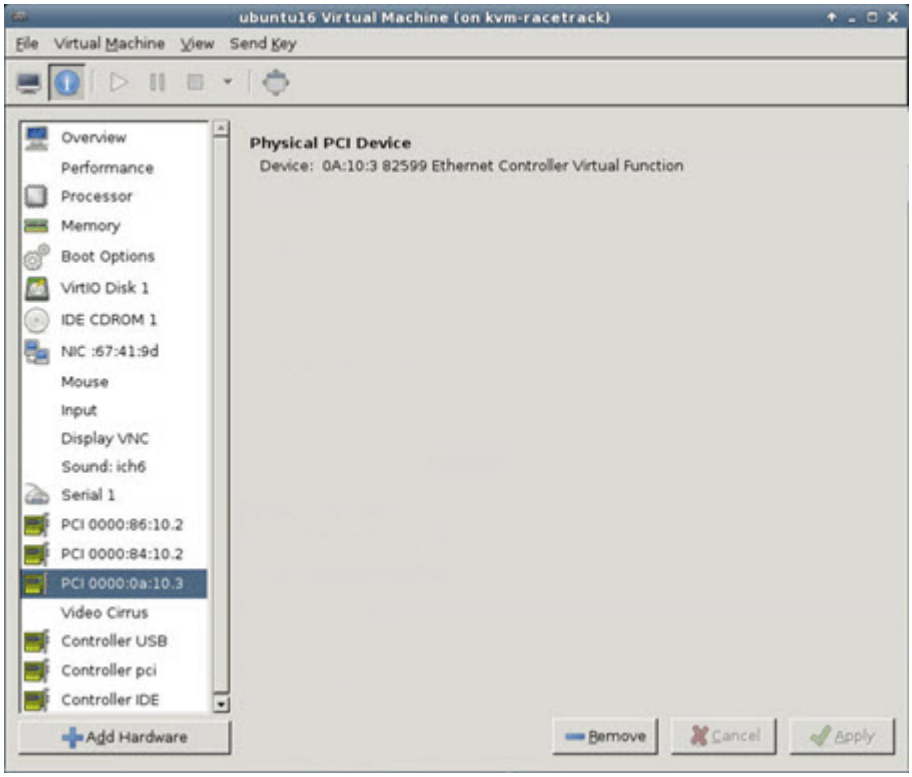
Illustration 7 : Liste des fonctions virtuelles



Étape 3 Sélectionnez l'une des fonctions virtuelles disponibles et cliquez sur **Finish** (Terminer).

Le périphérique PCI apparaît dans la liste du matériel; notez la description du périphérique comme fonction virtuelle de contrôleur Ethernet.

Illustration 8 : Fonction virtuelle ajoutée



Prochaine étape

- Utilisez la commande **show interface** (afficher l'interface) de la ligne de commande ASA virtuel pour vérifier les interfaces nouvellement configurées.
- Pour en savoir plus, utilisez le mode de configuration d'interface sur l'ASA virtuel pour configurer et activer l'interface pour la transmission et la réception du trafic; consultez le chapitre *Configuration de base de l'interface* du Guide de configuration de l'interface de ligne de commande pour les opérations générales de série Cisco Cisco Secure Firewall ASA.

Utilisation et rapport d'utilisation du processeur (CPU)

Le rapport d'utilisation du processeur résume le pourcentage du processeur utilisé pendant la période précisée. En règle générale, le cœur fonctionne sur environ 30 à 40 % de la capacité totale du processeur en dehors des heures de pointe et sur environ 60 à 70 % de capacité pendant les heures de pointe.

Important À partir de la version 9.13(1), n'importe quelle licence d'ASA virtuel peut être utilisée sur n'importe quelle configuration de vCPU/de mémoire d'ASA virtuel. Cela permet aux clients d'ASA virtuel d'exécuter une grande variété de profils de ressources VM.

Utilisation de vCPU dans ASA virtuel

L'utilisation de vCPU dans ASA virtuel affiche la quantité de vCPU utilisée pour le chemin de données, le point de contrôle et les processus externes.

L'utilisation de vCPU signalée par vSphere comprend l'utilisation de l'ASA virtuel comme décrit plus :

- Délai d'inactivité d'ASA virtuel
- %SYS surdébit utilisé pour la machine virtuelle ASA
- Surdébit du déplacement des paquets entre les vSwitches, les vNIC et les pNIC. Ce surdébit peut être assez important.

Exemple d'utilisation du processeur

La commande **show cpu usage** (afficher l'utilisation du processeur) peut être utilisée pour afficher les statistiques d'utilisation du processeur.

Exemple

```
Ciscoasa#show cpu usage

Utilisation du processeur pendant 5 secondes = 1 %; 1 minute : 2 %; 5 minutes : 1 %
```

Voici un exemple dans lequel l'utilisation de vCPU signalée est sensiblement différente :

- Rapports ASA virtuel : 40 %
- DP : 35 %
- Processus externes : 5 %
- ASA (comme les rapports ASA virtuel) : 40 %
- Interrogation ASA inactive : 10 %
- Frais généraux : 45 %

Le surdébit est utilisé pour effectuer des fonctions d'hyperviseur et pour déplacer des paquets entre les NIC et les vNIC à l'aide du vSwitch.

Rapport d'utilisation du processeur (CPU) de KVM

La

`virsh cpu-stats` *`domain`* `--total start count`

fournit les renseignements statistiques du CPU sur la machine virtuelle invitée indiquée. Par défaut, elle affiche les statistiques de tous les CPU, ainsi que le total. L'option `--total` n'affiche que les statistiques totales. L'option `--count` n'affiche que les statistiques du *nombre* de CPU.

Des outils comme OProfile, top, etc. donnent l'utilisation totale du processeur d'une VM KVM particulière, qui inclut l'utilisation du processeur de l'hyperviseur et de la VM. De même, des outils comme XenMon, qui sont propres à Xen VMM, attribuent une utilisation totale du CPU de l'hyperviseur Xen, c'est-à-dire du domaine 0, mais ne le séparent pas en fonction de l'utilisation de l'hyperviseur par VM.

À côté de cela, certains outils existent dans les cadres d'informatique en nuage comme OpenNebula, qui ne fournit que des renseignements approximatifs sur le pourcentage de CPU virtuel utilisé par une VM.

Graphiques ASA virtuel et KVM

Il existe des différences dans les numéros de % du CPU entre l'ASA virtuel et KVM :

- Les numéros de graphique KVM sont toujours supérieurs aux numéros d'ASA virtuel.
- KVM l'appelle « %CPU usage »; l'ASA virtuel l'appelle « %CPU utilization ».

Les termes « %CPU utilization » et « %CPU usage » ont des significations différentes :

- CPU utilization fournit des statistiques sur les CPU physiques.
- CPU usage fournit des statistiques sur les CPU logiques, qui sont basées sur l'hyperthreading du processeur. Mais, comme un seul vCPU est utilisé, l'hyperthreading n'est pas activé.

KVM calcule le pourcentage d'usage du CPU comme suit :

Nombre de processeur (CPU) virtuels utilisés activement, spécifié en pourcentage du nombre total de CPU disponibles

Ce calcul est la vue de l'hôte de l'utilisation du CPU, et non la vue du système d'exploitation invité, et est l'utilisation moyenne du CPU sur tous les CPU virtuels disponibles dans la machine virtuelle.

Par exemple, si une machine virtuelle avec un CPU virtuel fonctionne sur un hôte qui a quatre CPU physiques et que l'usage des CPU est de 100 %, la machine virtuelle utilise complètement un CPU physique. Le calcul de l'usage du CPU virtuel est l'usage en MHz/nombre de CPU virtuels x fréquence du cœur

À propos de la traduction

Cisco peut fournir des traductions du présent contenu dans la langue locale pour certains endroits. Veuillez noter que des traductions sont fournies à titre informatif seulement et, en cas d’incohérence, la version anglaise du présent contenu prévaudra.