GUIDE D'UTILISATION FEATURE DETECTOR APPLICATION

THEME : Permettre la migration de machines Virtuelles dans un environnement hétérogène.

A. INTRODUCTION

Xen est un moniteur de machine virtuelle (VMM) de paravirtualisation, ou "hyperviseur", pour l'architecture de processeur x86, il peut exécuter en toute sécurité plusieurs machines virtuelles sur un seul système physique avec des performances proches de celles d'origine. Xen est né d'un projet de recherche mené par Ian Pratt à l'université de Cambridge et la première version a été publiée en 2003.
La technologie de la machine virtuelle facilite les fonctionnalités de niveau entreprise, notamment
- Des machines virtuelles aux performances proches du matériel natif.
- La migration en direct des machines virtuelles en cours d'exécution entre les hôtes physiques.
- Une excellente prise en charge du matériel (prise en charge de la plupart des pilotes de périphériques Linux).
- Des pilotes de périphériques redémarrables et protégés par un bac à sable.

Ainsi La paravirtualisation permet une virtualisation très performante, même sur des architectures comme x86, qui sont traditionnellement très difficiles à gérer. Le support de Xen est disponible pour un nombre croissant de systèmes d'exploitation et est intégré dans de nombreuses distributions telles que : Debian, Ubuntu, Suse, Red Hat, Fedora, CentOS, etc

CARACTÉRISTIQUES DE LA MACHINES

Xen ne fonctionne actuellement que sur l'architecture x86, ce qui nécessite un processeur "P6" ou un processeur plus récent (par exemple, Pentium Pro, Celeron, Pentium II, Pentium III, Pentium IV, Xeon, AMD Athlon,AMD Duron).
Les machines multiprocesseurs sont prises en charge, Xen peut actuellement utiliser jusqu'à 4 Go de mémoire. Il est possible pour les machines x86 d'adresser jusqu'à 64 Go de mémoire physique
Processeur de type IA64, PPC

Portage en cours pour les processeurs de type ARM
Le port x86/64 est la voie prévue pour supporter des tailles de mémoire plus importantes.
Xen décharge la plupart des problèmes de support matériel sur le système d'exploitation invité qui s'exécute dans le domaine 0.

Xen lui-même ne contient que le code nécessaire à la détection et au démarrage des processeurs secondaires, à la mise en place du routage des interruptions et à l'exécution des tâches de gestion de la mémoire. le routage des interruptions et l'énumération du bus PCI. Les pilotes de périphériques s'exécutent dans un système d'exploitation invité privilégié plutôt que dans Xen lui-même. Cette approche permet d'assurer la compatibilité avec la majorité des périphériques matériels pris en charge par Linux. La version par défaut de XenLinux contient la prise en charge de matériel réseau et de disques relativement modernes de type serveur, mais il est possible d'ajouter le support pour d'autres matériels en configurant le noyau XenLinux.

B. PREREQUIS

Notre système est utilisé pour déterminer si la migration d'une Machine virtuelle, exécutant des applications données, à l'aide d'un hyperviseur donné sur une machine physique, vers une autre machine physique est possible. Pour note cas d'étude nous utilisons l'hyperviseur Xen présenté plus haut, exécutant l'application PostgreSQL. Il est également à noter que le système est basé sur des langages de programmation arbitraires.

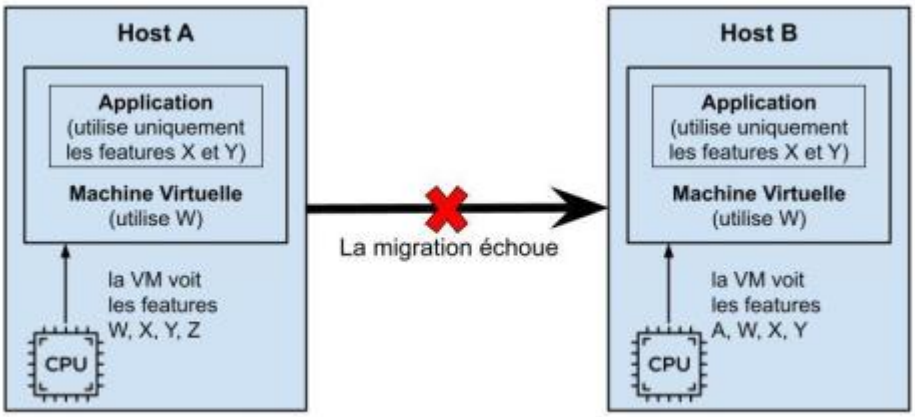

Figure 1Illustration Migration

Nous avons par conséquent pour se faire besoin

- D’une machine ayant un OS avec l’hyperviseur Xen Installé, et d’une connexion Internet pour la suite de la procédure. L’installation et la configuration de Xen est expliqué dans le guide accessible via ce lien :
- De l’interpréteur Python pour pouvoir exécuter la partie en python de notre application. Pour installer python nous utilisons les commandes suivantes :

sudo apt update && sudo apt upgrade -y

sudo apt install python3

C. PROCEDURE

1. CREATION DE LA VM

- Cloner le projet de github sur cette machine.
- Une fois Xen installé, démarrez la machine en choisissant l’OS avec Xen dans le grub
- Sur le terminal lancer la première exécution, celle du script **CreationVM.sh**, une fois positionné dans le répertoire du projet. Utilisez la commande suivante :

./CreateVM.sh

Ceci entrainera la création de la VM, accessible via un terminal (le nom d’utilisateur et le mot de passe sont données lors de la création.

2. INSTALLATION DE POSTGRESQL SUR LA MACHINE VIRTUELLE

- Cloner le projet sur cette machine virtuelle.
- L’application PostgreSQL ainsi que les Test Suites de l’application devront être installées sur la machine virtuelle. Ces deux étapes sont possibles en suivant les instructions des deux liens suivants :
  install postgresql from sources et run postgresql tests

3. OBTENTION DES FEATURES

Sur le terminal de la machine virtuelle :

- Installer g++, pour la compilation de codes c++ de l’application

sudo apt update
sudo apt install build-essential

- Compiler le fichier c++ **GetFeatures.c++** :

g++ GetFeatures -o GetFeatures

- Executer GetFeatures :

./GetFeatures

4. OBTENTION DES FEATURES NECESSAIRES

- Sur le Dom0 (Le système initial, sélectionné dans le grub), se placer dans le répertoire du projet
- Exécuter le fichier **header.py :**

Python3 header

- Une fois la commande exécutée, se placer dans le répertoire **home/postgres/résultats-tests/résultats**
- L'ensembles des features utiles est contenues dans le fichiers **features_utiles**

5. ANNEXE

Le fichier **main.py,** contenant l'interface graphique, entièrement réalisée de l'application n'est pas encore intégré à la couche métier de celle-ci et n'est par conséquent pas encore utilisable pour interagir avec elle.