Friday 13 May 2016

Penguin Cloud

Auteur: Romain Claret

But

jsLinux de Fabrice Bellard est un émulateur en ligne d'un LFS. Cependant, celui si n'a pas de connections réseau. Le but est de lui donner un réseau.

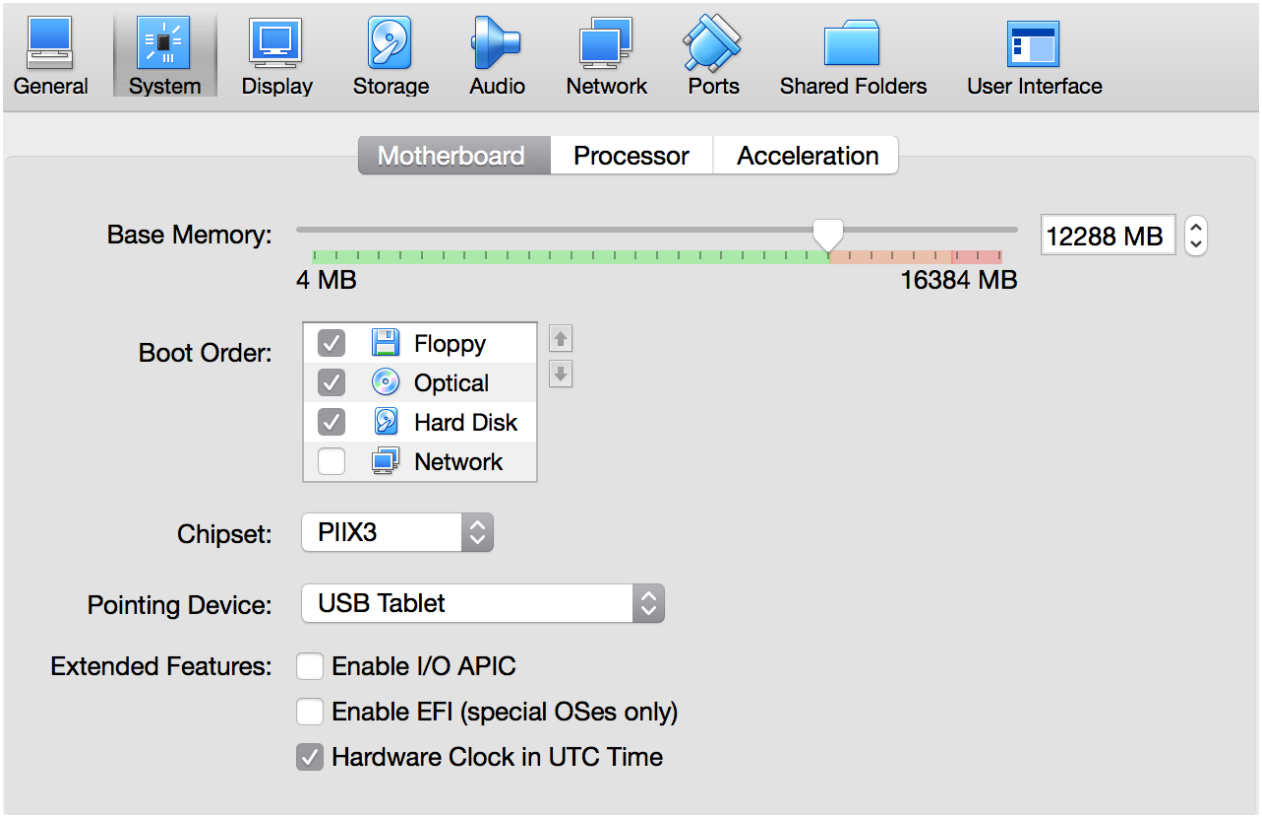
Principe

Le host, le serveur web hébergent jsLinux fait office de passerelle avec le monde extérieur.

Machine Hôte

- Pour commencer nous avons avoir besoin d'une machine virtuelle dans virtualbox tournant avec **Debian 7.8 32-bit et 40 GB d'espace** (la ram et les coeurs peuvent être modifié on-the-fly plus tard). Le nom de l'image d'installation est: *debian-7.8.0-i386-xfce-CD-1.iso* trouvable ici au moment de la rédaction de ce rapport: http://debian.nctu.edu.tw/debian-cd/7.8.0/i386/iso-cd/
- Ajouter un Disk Dur supplémentaire avec **20GB** qui contiendra LFS. Concernant l'espace mémoire de ce disque, il est précisé dans la documentation qu'il faut au moins 4 GB pour la partition LFS elle-même, et il est recommandé 10 GB. Dans notre cas il a été pris le double.
- Nous allons également donner un peu de puissance à notre machine virtuelle. Ici nous mettrons à disposition 12GB (12GB * 1024 = 12'288 MB) de ram:

- La machine est configurée et contrôlée par SSH à l'aide du port forwarding:



Particularités:

-

-

Étape -1 : Installation de la machine (si besoin)

- Installer Debian 7.2 32-bit dans virtualbox avec les propriétés suivantes:
  - hostname: cloudy-penguin
  - root password: cloud
  - Use entire disk and all files in one partition
  - Software sélection (espace pour sélectionner ou désélectionner):

- Installer le grub
- Penser à faire une snapshot une fois l'installation terminée.
- Mettre à jour l'os
  - Se connecter en root avec le password: cloud
  - `apt-get update`
  - `apt-get upgrade`

Étape 0 : Initialisation (Chapitre 0)

- Se connecter en root avec le password: lfs
- Installer les packages suivants: git
  - `apt-get install git build-essential`
- S'il y a une demande pour "Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 xfce-CD Binary-1 20150110-13:31" lors d'une installation, commenter cette ligne dans sources.list
  - `vi /etc/apt/sources.list`

```
# deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 xfce-CD Binary-1 20
150110-13:31]/ wheezy main

# deb cdrom:[Debian GNU/Linux 7.8.0 _Wheezy_ - Official i386 xfce-CD Binary-1 20
150110-13:31]/ wheezy main
```

Étape 1: Récupérer jsLinux

- On va récupère le contenu de jsLinux à l'aide du script suivant. À exécuter depuis la racine qui contiendra notre dossier jslinux.
  - wget -r -p -np -k http://bellard.org/jslinux
  - mv ./bellard.org/jslinux ./jslinux
  - rm -rf ./bellard.org
  - cd jslinux
  - wget https://github.com/levskaya/jslinux-deobfuscated/blob/master/vmlinux-2.6.20.bin?raw=true
  - mv vmlinux-2.6.20.bin?raw=true vmlinux-2.6.20.bin

- for i in $(seq -f "%09g" 0 999)
- do
- wget http://bellard.org/jslinux/hda$i.bin
- done
- tar xvfz linuxstart-20120111.tar.gz
- cd tmp/linuxstart-20120111/
- make
- mv linuxstart.bin ../../
- cd ../../
- rm -rf tmp
- rm -f linuxstart-20120111.tar.gz

Étape 1: Essayer jsLinux

- Exécuter la commande
  - python -m SimpleHTTPServer
- Depuis le navigateur, allez sur l'URL localhost:8000
- Linux devrait se lançait au chargement de la page.
- Dans mon cas j'ai que du kernel panic…

Résultats:

Il avait été prévu au début d'utiliser la technologie webRTC pour permettre une connexion peer-to-peer entre les différents utilisateurs utilisant jsLinux, cependant le temps à disposition et les difficultés techniques ont fortement ralenti le projet. J'ai donc cherché d'autres solutions comme créer un driver de modem et le bind au serveur host.

Bref, un fiasco…

Sources:

- Ce qui ma principalement aidé à comprendre plus ou moins comment jsLinux fonctionne.
  - https://github.com/levskaya/jslinux-deobfuscated

- Tentatives de faire fonctionner un serveur PPTP
  - https://help.ubuntu.com/community/PPTPServer
- Web sockets
  - https://github.com/kanaka/websockify
- jsModem, il utilise la même idée, il semblerait que ça marche, mais pas chez moi
  - https://github.com/ewiger/jsmodem
- D'autre projet de virtualisation Linux dans le browser
  - http://s-macke.github.io/jor1k/demos/main.html
  - http://copy.sh/v86/ (avec support du network via udhcpcd)
  - http://52.32.189.224/angel-simulator/
  -