Compte Rendu

Projet Reseaux : Conception Implémentation d'un Mini DataCenter

Etudiant
M. Jake MORISON
M. Eric Michel FOTSING

Encadrant
Dr.Olivier RICHARD
Dr Didier DONSEZ

Avril 2015

1. Sommaire

1. Sommaire 2
1. Objectif global 3
2. Cas d'utilisation 4
2.1 Demarrage d'une carte Jetson TK1 en réseau 4
2.2 Allocation/désallocation dynamique de la carte jetson 4
3. Architecture du mini datacentre 5
4. Demarrage de la carte jetson TK1 sur un NFSRoot 7
5. Allocation/désalloction dynamique d'une carte jtk 8
6. Difficultés rencontrés et perspectives 9
7. Conclusion 10

1. OBJECTIF GLOBAL

L'objectif global de notre projet était de provenir une plateforme d'expérimentation y compris nombreuses cartes Jetson TK1. Un cluster comme ça est utile pour les développeurs des choses comme mobile applications, jeux vidéo, multimédia, et autres choses qui manquent l'architecture de leur propres plateformes. Ce projet était bien s'inspiré par Grid5000.

Dans la suite de ce document nous aborderons les points suivants concernant le projet :

- Presentation des cas d'utilisations
- Présentation de l'architeture cible du mini datacenter
- Mise en ouvre du démarrage de la carte Jetson TK1 sur un partage NFSRoot
- Basculement automatique d'un NFSRoot d'une Carte Jetson TK
- Diffcultés rencontrés et piste d'amélioration
- Conclusion

2. CAS D'UTILISATION

Nous commencons ici par rappelé que le projet Mini Data Cente à base de carte Jetson TK1 est composé de deux équipes :

- Une équipe pour s'occuper des aspects systèmes du projet
- Une autre équipê pour gérér tous les apects portail et interface avec l'utilisateur

Le cas d'utilisation qui suivent ne concernent que les cas que nous avons inventoriés dans la partie système. Ainsi donc nous avons recensés les cas d'utilisation suivants :

- Démarrage d'une carte Jetson sur une image monté sur un partage NFS
- Allocation/Désallocation dynamique de la carte
- Préparation, ajout et suppression des images OS
- Ajout et suppression des utilisateurs du mini datacente

2.1 Demarrage d'une carte Jetson TK1 en réseau

Il s'agit dans cette partie de permettre à un utilisateur quelconque ayant créé un partage NFS dans lequel il a préalablement disposé son image OS, démarrér la carte en utilisant ce partage comme image OS

2.2 Allocation/désallocation dynamique de la carte jetson

A traver ce cas d'utlisation, nous devons non seulement permettre à plusieurs utilisateurs de partager la même carte, mais aussi à un utilisateur possédant plusieurs images OS de démarrer de manière flexible la carte avec l'image OS de son choix.

2.2 Préparation, ajout et suppression des images OS

Ce cas d'utilisation vise à offrir à l'utilisateur la possiblité de préparer, d'enregistrer et supprimer ses images OS auprès de middleware centrale

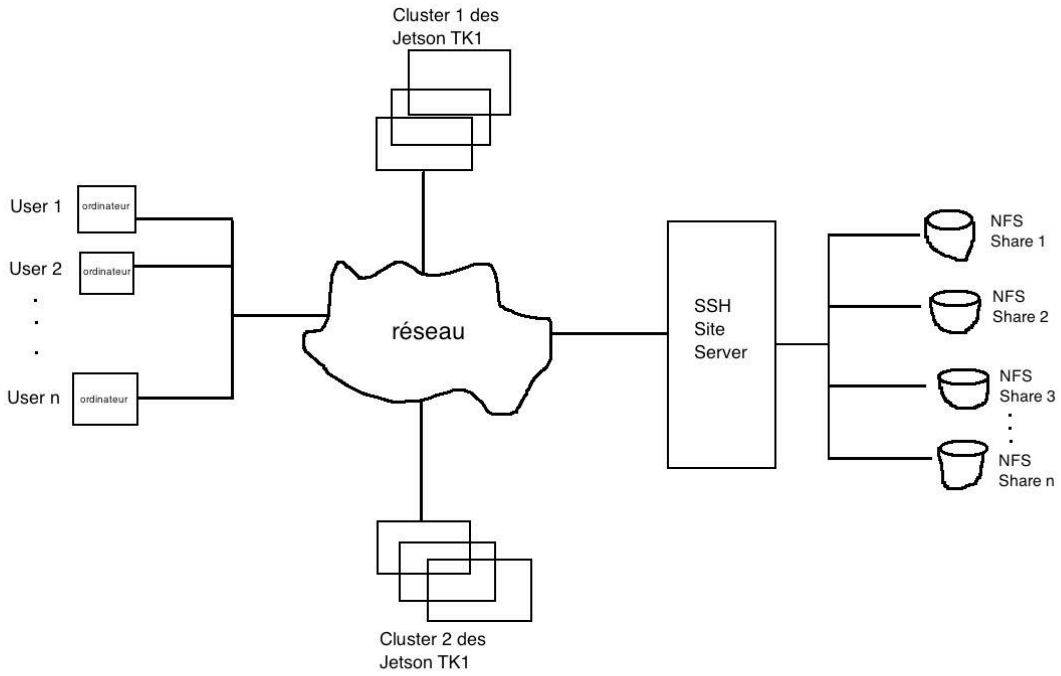
2.2 Ajout et suppression des utilisateur dans le mini datacente

Ce cas d'utilisation vise à offrir à l'adminisrateur la possiblité de creer et de supprimer les utilisateurs de la plate forme sur chaque middleware central

3. ARCHITECTURE DU MINI DATACENTRE

3.1 Vue d'ensemble de l'architecture

Le schema suivant donne idée globale de l'ossature de notre data centre



Comme il est possiblde le constater, les principaux composante de cette architecture sont

- Les poste des utilisateur des la plateforme
- Le réseau proprement configuré
- Dans cartes Jetson TK1 connecté au réseau reproupés en cluster ou pas
- Un serveur ou plusieurs serveur de site

Dans la suite nous détaills le rôle et la composition de chacun de ces composants dans notre architecture

3.1 Poste utilisateur

Cest un ordinateur quelconque doté essentiellement d'un client SSH qui permetra à l'utilisateur de se connecter à ses images OS. Le client SSH devra être configuré pour s'authentifier sur le serveur au moyens d'une clé privée. Et la clé publique correspondante doit préalablement avoir été installé sur le serveur de site.

3.3 Cartes Jetson TK1

C'est la ressource principalement géré dans le datacenter. Il s'agit d'un SOC Tegra K1 doté d'un processeur spécifique.

Chaque carte de la plate forme doit être flashé pour démarrer sur partage NFS dont le point exacte comme illustré sans cet exple de code

```
sudo ./flash.sh -N 192.168.137.130:/NFSRoot1 -n
192.168.137.131:192.168.137.130:192.168.137.1:255.255.255.0 jetson-tk1 eth0
```

Vous pouvez utiliser les liens internet suivants pour avoir plus de détails sur les cartes Jetson TK1 et la procédure de flash :

- https://developer.nvidia.com/get-started-jetson
- http://elinux.org/index.php?title=Jetson_TK1&redirect=no
- https://devtalk.nvidia.com/default/board/139/embedded-systems/1
- https://plus.google.com/114318922342198493952/posts

3.3 Serveur Jetson TK

C'est le composant princpal de l'architecture du mini data cente : Il sert d'intermédiaire entre l'utilisateur et la carte cible. Il est reponsable de la fourniture des fonctions définies dans la senctions cas d'utilisations définie plus haut.

Le serveur Jetson TK est doté des composants logiciels suivants :

> Un serveur SSH configuré pour supporté uniquement l'authentification a base de clé publique

> Un serveur NFS qui exportera les partages NFS qui contiendrons les images de chaques untilisateur

> Une base de données SQLITE qui stocke les métadonnées du système

> La liste des comptes utilisateurs du systèmes préconfigurés

> L'executable jtkoar qui est le proramme que l'utilisateur va uiliser pour exectuer les fonction attendu du système

4. DEMARRAGE DE LA CARTE JETSON TK1 SUR UN NFSROOT

Ayans prsi connaissance de tous les composantes de notre datacenter, nous allons décrire comment le processus de démarrage d"une carte Jetson TK1 en réseau se configure.

Pour démarrer une carte Jetson TK1 en reséu, il faut exécuter la série des actions suivantes :

1. Configurer un point de montage (Exple /NFSRoot) le serveur NFS
2. Configurer un RootFS
3. Copier le RootFS sur le NFSRoot
4. Flasher la carte Jetson TK1 pour quelle boot en réseau

4.1 Configurer le serveur NFS et y créer un point de montage

Le bout de code suivant illuste les commande qui doivent être exécuté sur le serveur Jetson TK1 pour configurer le point montage NFS qui abritera l'image NFS

```
$ sudo apt-get install nfs-common nfs-kernel-server

$ cat "/nfsroot *(rw,nohide,insecure,no_subtree_check,async,no_root_squash)" >> /etc/exportfs
$ sudo /etc/init.d/nfs-kernel-server restart
$ sudo mkdir /nfsroot
$ cd ./rootfs
$ sudo cp –a * /nfsroot
$ sudo exportfs -a
```

4.2 flasher la carte Jetson TK1

La procédure pour flasher les carte Jetson est décrite par la suite des action suivante :

- Télécharger le le package Linux Driver for Tegra (L4T) sur le site de Nvidia
- Extraire ce paquet dans dossier de votre choix
- Brancher la carte Jetson TK1 en Recovery Mode puis exécuter le script de flash fournie avec le paquet L4T
-

```
sudo ./flash.sh -N 192.168.137.130:/NFSRoot1 -n
192.168.137.131:192.168.137.130:192.168.137.1:255.255.255.0 jetson-tk1 eth0
```

- Redémarrer la carte et elle est prête pour charger son OS a pratir du partage NFS spécifié dans la procédure de flash

5. ALLOCATION/DÉSALLOCTION DYNAMIQUE D'UNE CARTE JTK

La procédure que nous avons mise en places pour allouer et désallouer une cartes peut se traduire dans l'alogirthme suivant :

1. Charger les détails sur la carte et sur l'image actuellement configuré pour booter la carte sélectionné
2. Charger les détails sur l'image sélectionné par l'utilisateur
3. Initiation du redémarrage de la carte
4. Attendre la termination de la connexion NFS au partage. A travers un mecanisme de synchronisation pour s'assurer que dans les commandes qui suivront carte sera bien arrêté et le montage NFS libre de toute manipulation.
5. Desactivation du partage jadis exploité par la carte
6. Sauvegarde de l'image actuellement référencé par le NFSRoot dans /home/jtkusr/NFSRoots/RootFS1
7. Redirection de l'image /homejtkusr/NFSRoots/RootFS vers le partage /NFSRoot
8. Reactivation du partage /NFSRoot
9. Attente du demarrage de la carte a travers un mecanisme de synchrnisation pour s'assurer que dans les commandes qui suivront carte sera bien démarré
10. Enregistrement de la session Carte - Image OS dans la base de données de jtkoar.
11. Ouverture et transfert session ssh utilisateur sur la carte

Cette algorithme est implémenté dans l'éxécutable jtkaor. Mais le bout de code suivants illuste un exemple de son éxécution .

```
ssh  root@192.168.131.1 reboot

#step 1
sudo mv /home/jtkuser/NFSRoots/RootFS1 /NFSRoot1
sudo chown root:root /NFSRoot1
sudo exportfs -o rw,nohide,insecure,no_subtree_check,async,no_root_squash *:/NFSRoot1

#step 2
sudo exportfs -u *:/NFSRoot1
sudo chown jtkserveur:jtkserveur /NFSRoot1
sudo mv /NFSRoot1 /home/jtkserveur/NFSRoots/RootFS1

#step 3
sudo mv /home/jtkserveur/NFSRoots/RootFS2 /NFSRoot1
sudo chown root:root /NFSRoot1
sudo exportfs -o rw,nohide,insecure,no_subtree_check,async,no_root_squash *:/NFSRoot1          $
sudo exportfs -a
```

6. DIFFICULTÉS RENCONTRÉS ET PERSPECTIVES

Les difficultés rencontrés dans le cadre de ce projet on été relative au fait que c'est un projet vaste de par la diversité des composants qui sont manipulés et manipulables dans ce type d'infrastructure.

Particulierement, la prise en main des composant suivants n'a pas été facile :

- Carte Jetson TK
- Plate forme Grid5000
- OAR scheduler
- Serveur NFS
- Wake On LAN démarrage de carte Jetson TK1 en reséau n'étant pas encore implémenté sur ces carts.

Les perspective de ce projet pourrait être les suivantes :

- Intégrer Oar en grandeur nature pour l'allocation et la désallocation des cartes
- Implémenter le Wake On Lan sur les cartes Jetson TK 1
- Porter d'autre distributions linux que le processeur Tegra TK1.

7. CONCLUSION

Nous remercions l'encadrement pour ce type de projet, qui nous a vraiment permis d'aborder les problématiques uselles dans le cadre des datacenter. Partant du boot en réseau, jusqu'à l'allocation et la désallocation des ressouces du datacenter. Nous nous confrontés à une diversité des problématiques desquelles nous pensont être ressorties avec beaucoup de savoir.