

# Déploiement simplifié de stations sans disque avec FaDDeF

Mohammed Khabzaoui

UMR 8524 Université Lille1

13 fevrier 2014

# Qu'est-ce qu'un poste "diskless" ?

Un poste "diskless" peut se résumer à un poste qui (source Wikipedia) :

- Démarre sur le réseau
- N'a pas besoin d'un disque dur mais qui peut en avoir
- Exploite pleinement, peu, ou pas du tout les ressources locales

# Qu'est-ce qu'un poste "diskless" ?

Un poste "diskless" peut se résumer à un poste qui (source Wikipedia) :

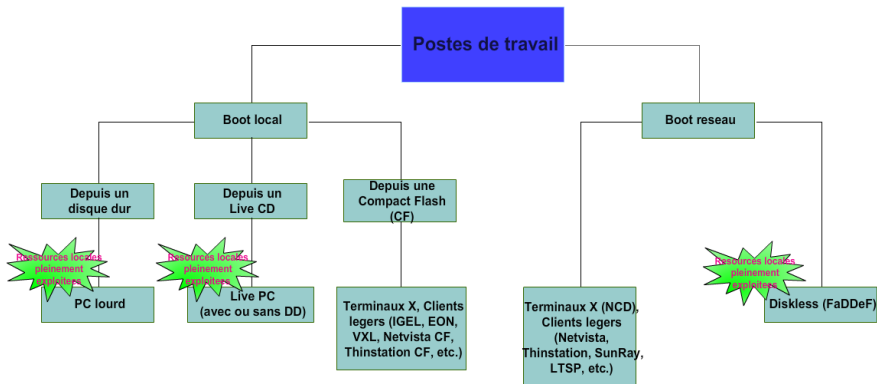
- Démarre sur le réseau
- N'a pas besoin d'un disque dur mais qui peut en avoir
- Exploite pleinement, peu, ou pas du tout les ressources locales

# Qu'est-ce qu'un poste "diskless" ?

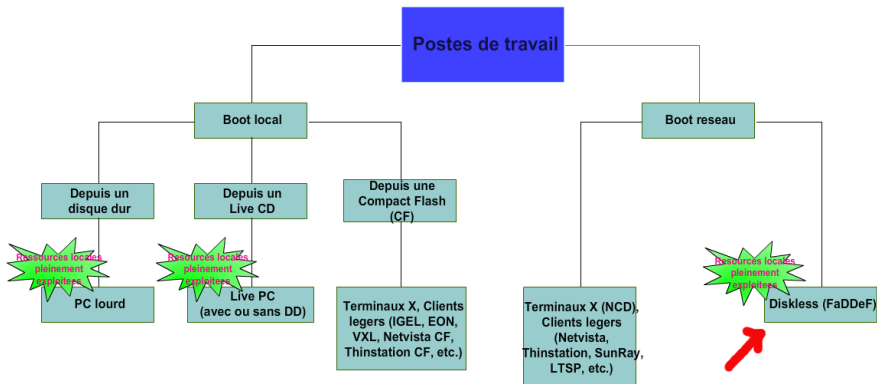
Un poste "diskless" peut se résumer à un poste qui (source Wikipedia) :

- Démarre sur le réseau
- N'a pas besoin d'un disque dur mais qui peut en avoir
- Exploite pleinement, peu, ou pas du tout les ressources locales

# Positionnement du poste "diskless" par rapport aux autres types de postes de travail



# Positionnement du poste "diskless" par rapport aux autres types de postes de travail



# Principe de fonctionnement du "diskless"

Dans notre cas, un système "diskless" fonctionne de la manière suivante :

- Chargement du noyau et d'un ramdisk initial via le réseau (PXE)
- Montage de l'arborescence racine en lecture seule (readonly root) via un protocole réseau (NFS)
- Initialisation de l'environnement "diskless"

Contrainte : le système doit pouvoir écrire (fichiers systèmes, fichiers temporaires, etc.)

# Principe de fonctionnement du "diskless"

Dans notre cas, un système "diskless" fonctionne de la manière suivante :

- Chargement du noyau et d'un ramdisk initial via le réseau (PXE)
- Montage de l'arborescence racine en lecture seule (readonly root) via un protocole réseau (NFS)
- Initialisation de l'environnement "diskless"

Contrainte : le système doit pouvoir écrire (fichiers systèmes, fichiers temporaires, etc.)



# Principe de fonctionnement du "diskless"

Dans notre cas, un système "diskless" fonctionne de la manière suivante :

- Chargement du noyau et d'un ramdisk initial via le réseau (PXE)
- Montage de l'arborescence racine en lecture seule (readonly root) via un protocole réseau (NFS)
- Initialisation de l'environnement "diskless"

Contrainte : le système doit pouvoir écrire (fichiers systèmes, fichiers temporaires, etc.)

# Principe de fonctionnement du "diskless"

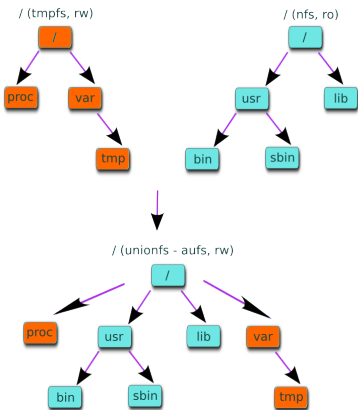
Dans notre cas, un système "diskless" fonctionne de la manière suivante :

- Chargement du noyau et d'un ramdisk initial via le réseau (PXE)
- Montage de l'arborescence racine en lecture seule (readonly root) via un protocole réseau (NFS)
- Initialisation de l'environnement "diskless"

Contrainte : le système doit pouvoir écrire (fichiers systèmes, fichiers temporaires, etc.)

# Union de systèmes de fichiers

Une union de systèmes de fichiers est une "superposition" d'un ensemble de systèmes de fichiers appelés **branches** de l'union :



**lecture** : première occurrence dans une branche en parcourant de gauche à droite

**création** : dans la première branche accessible en lecture/écriture

**modification** : si l'entrée est dans une branche en lecture seule alors copie de celle-ci dans la première branche accessible en lecture/écriture puis modification

**suppression** : si l'entrée est dans une branche en lecture seule alors création d'une entrée spéciale dans la première branche accessible en lecture/écriture matérialisant la suppression

FaDDeF repose sur :

- Un module noyau **unionfs** ou **aufs**
- Un script **mkreadonlyroot.diskless** qui met en place l'arborescence racine à exporter en lecture seule
- Un script **mkinitrd.diskless** qui crée un ramdisk adapté pour le fonctionnement en diskless

Site du projet : <http://projets.mathrice.fr/faddef>

Distributions Linux supportées : CentOS, Fedora, Mandriva et Ubuntu

FaDDeF repose sur :

- Un module noyau **unionfs** ou **aufs**
- Un script **mkreadonlyroot.diskless** qui met en place l'arborescence racine à exporter en lecture seule
- Un script **mkinitrd.diskless** qui crée un ramdisk adapté pour le fonctionnement en diskless

Site du projet : <http://projets.mathrice.fr/faddef>

Distributions Linux supportées : CentOS, Fedora, Mandriva et Ubuntu

FaDDeF repose sur :

- Un module noyau **unionfs** ou **aufs**
- Un script **mkreadonlyroot.diskless** qui met en place l'arborescence racine à exporter en lecture seule
- Un script **mkinitrd.diskless** qui crée un ramdisk adapté pour le fonctionnement en diskless

Site du projet : <http://projets.mathrice.fr/faddef>

Distributions Linux supportées : CentOS, Fedora, Mandriva et Ubuntu

FaDDeF repose sur :

- Un module noyau **unionfs** ou **aufs**
- Un script **mkreadonlyroot.diskless** qui met en place l'arborescence racine à exporter en lecture seule
- Un script **mkinitrd.diskless** qui crée un ramdisk adapté pour le fonctionnement en diskless

Site du projet : <http://projets.mathrice.fr/faddef>

Distributions Linux supportées : CentOS, Fedora, Mandriva et Ubuntu

FaDDeF repose sur :

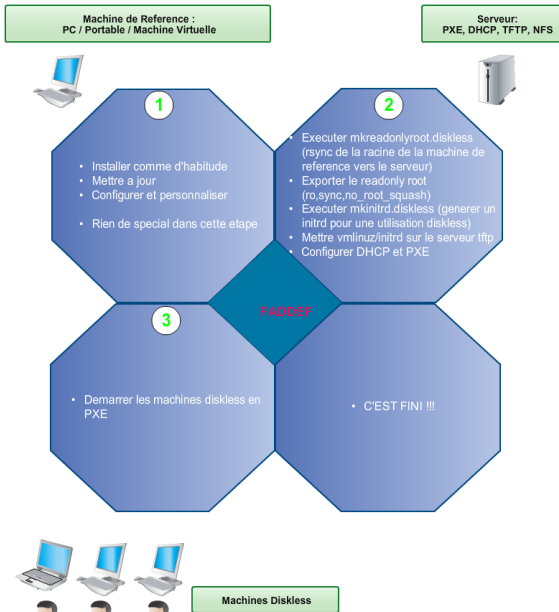
- Un module noyau **unionfs** ou **aufs**
- Un script **mkreadonlyroot.diskless** qui met en place l'arborescence racine à exporter en lecture seule
- Un script **mkinitrd.diskless** qui crée un ramdisk adapté pour le fonctionnement en diskless

Site du projet : <http://projets.mathrice.fr/faddef>

Distributions Linux supportées : CentOS, Fedora, Mandriva et Ubuntu



# Mise en œuvre de FaDDeF



- Un réseau local (LAN) ou correctement segmenté (confinement du DHCP et regroupement des postes concernés dans un VLAN)
- Un réseau commuté au moins à 100Mb/s
- Un serveur DHCP
- Un serveur dédié NFS/PXE
- Un serveur de fichiers pour les comptes utilisateurs (de préférence NFSv4)
- Des PCs standards (bien détectés par la distribution GNU/Linux) avec suffisamment de RAM (au moins 2 Go)

**Ajoutez la configuration de votre serveur DHCP (pour utiliser le protocole PXE) :**

```
next-server IPserveurPXE;  
filename "/tftpboot/pxelinux.0"
```

**Configuration des hôtes avec IP fixée**

```
host diskless1 {  
    hardware ethernet 00:0f:75:af:eb:44;  
    fixed-address 192.168.1.100;  
    filename "/tftpboot/pxelinux.0"  
    # definit le serveur qui servira le fichier pxelinux.0  
    next-server 192.168.1.1;  
}
```

**export NFS read-only de /diskless/madistribution**

```
/diskless/madistribution *(ro,sync,no_root_squash)
```

**Installez le package correspondant à syslinux et tftp-server et créez le fichier /tftpboot/pxelinux.cfg/default :**

```
default diskless  
label diskless  
KERNEL vmlinuz-xxx.diskless  
APPEND initrd=initrd-xxx.img.diskless NFSROOT=IP_NFS  
:/diskless/ma_distribution
```

# Configuration de la machine de référence

- Installez une distribution récente sur un PC de référence, configurez là pour permettre une utilisation normale (authentification, montage des comptes utilisateurs, etc.)
- Installez ou veillez à bien avoir le module unionfs ou aufs dans les modules disponibles (apt-get install aufs-tools), ou compilez-le.
- Veillez à ce que le PC ne loggue que vers un serveur syslog et plus en local
- Installez rsync et le service SSH
- Générez un biché SSH (clé privée sur le serveur NFS/PXE et clé publique pour le root sur la machine de référence)

## Sur le serveur NFS :

- Récupérez FaDDeF :  
**<http://projets.mathrice.org/faddef/dist/faddef-last.tar.gz>**

- Puis :

```
tar xzf faddef-last.tar.gz
cd faddef
cd build-root
./mkreadonlyroot machine_reference:/ /diskless/madistribution dist
cd ../initrd
./updateDiskless.diskless -a /diskless/madistribution 3.2.XX-XXXX /tftpboot
```

- -a : choisir selon la disponibilité de unionfs ou aufs dans votre distribution de référence
- 3.8.XX-XXXX : la version du noyau dans votre distribution de référence
- dist : correspond à la distribution de votre machine de référence dans build-root/add-ons
- Adaptez selon vos besoins /diskless/madistribution/etc/faddef/faddef.sh

FaDDeF peut être utilisé dans diverses situations :

- Pour des postes de travail dans un laboratoire
- Dans des salles d'enseignement / TP / formation
- Dans des salles en libre accès
- Dans une salle machine pour les besoins d'une conférence / congrès
- Pour faire du calcul
- Pour faire du dépannage

FaDDeF est en production depuis 2006 :

- A l'Institut de mathématiques de Bordeaux (laboratoire, bibliothèque) : 2 serveurs Xeon, 200 postes, 1 Gb/s et 100 Mb/s, Mandriva
- Au Laboratoire de mathématiques Paul Painlevé de Lille (laboratoire, bibliothèque) : NetApp, 200 postes, 1 Gb/s et 100 Mb/s, Fedora
- A l'UFR de mathématiques, Université Lille1 (salles de cours + salles de TP Calculs scientifiques ) : plus de 60 postes, 100 Mb/s, Fedora
- A l'UFR de Mathématiques, Université de Tours (salles de cours) : 100 postes, 1 Gb/s et 100 Mb/s, Fedora
- A l'Institut d'Etudes Politiques de Lyon (salle en libre accès du service documentation) : 20 postes, Ubuntu
- Au CREMI Université Bordeaux 1 (salles de cours) : 2 serveurs, plus de 100 postes
- Au Laboratoire de Mathématiques, Nice : 200 postes, Mint



# Conclusion

# Avantages pour l'utilisateur

- Fonctionnement identique au poste lourd avec un peu moins de bruit
- Pleine exploitation des ressources matérielles locales
- Meilleure prise en charge des périphériques locaux que les clients légers (son, graveur, etc.)
- Meilleure prise en charge logicielle que les clients légers (3D, OpenGL, fontes, etc.)

# Avantages pour l'administrateur

- Gestion centralisée
- Pas besoin de serveurs de terminaux (XDMCP par exemple)
- Déploiement simplifié
- Portabilité, sécurité
- Coût de possession optimal

- Requiert suffisamment de RAM (2 Go minimum)
- Dépendance vis à vis du réseau, nécessité d'un serveur NFS connecté en Gb/s ( Min =100 Mb/s)
- Limité aux distributions GNU/Linux (pas de Windows, BSD, Solaris)