



par Guido Socher (homepage)

Exécution d'applications distantes avec X11



L'auteur:

Guido adore Linux non seulement parce qu'il est intéressant de découvrir le fonctionnement des systèmes d'exploitation mais aussi à cause des personnes impliquées dans sa conception.

Résumé:

Beaucoup de nouveaux venus à Linux pensent que son bureau graphique est juste un autre système "à la Windows" où vous pouvez lancer des applications qui vont apparaître dans des fenêtres séparées. Certains remarquent la possibilité d'avoir plusieurs bureaux mais ça semble s'arrêter là. Le système de fenêtrage X Window de Linux (X11) est beaucoup plus que cela ! C'est un système de fenêtrage réseau. Nous verrons les nouvelles et puissantes fonctionnalités que cela offre.

Traduit en Français par:

Olivier Pons

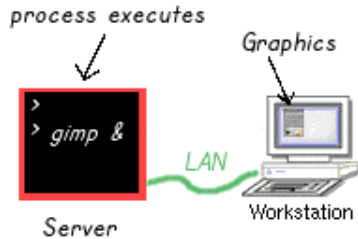
<pons/at/cnam.fr>

Le concept d'affichage

Lors de son lancement, chaque application graphique de X Window lit la variable d'environnement DISPLAY pour découvrir l'écran d'ordinateur auquel envoyer ses graphiques. Si on ajoute les capacités réseau du système X Window, ceci permet d'exécuter des applications graphiques à distance.

C'est-à-dire que vous utilisez la puissance du processeur d'une machine alors que vous exécutez l'application à partir d'une autre. L'interface graphique complète apparaît sur la machine d'où vous l'exécutez. Vous ne remarquez pas que vous utilisez 2 ordinateurs.

La vitesse du réseau est naturellement importante ici mais une connexion locale à 10Mbit/s est plus que suffisante.



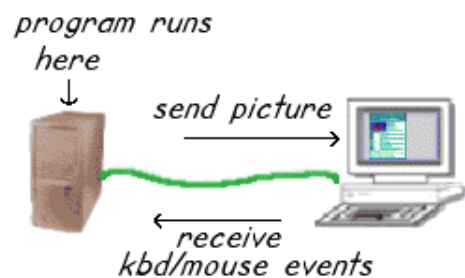
A quoi ça sert ?

Il existe de nombreuses applications de ce "graphisme réseau". Des entreprises l'utilisent pour gérer à distance du matériel pouvant se trouver à des milliers de kilomètres et emploient pour le contrôler, la même application que celle qui aurait servi en étant sur place.

Supposons que vous ayez 2 ordinateurs, une machine rapide à 1GHz et un vieux Pentium à 133MHz. Vous pouvez d'autant plus apprécier la vitesse de votre nouvelle machine que vous n'êtes pas assis devant elle. Votre soeur est peut-être actuellement assise devant la machine rapide et travaille avec. Aucune importance, vous continuez à en bénéficier.

Comment ça marche

Toutes les applications X Window, qu'elles se nomment gimp, xterm, konquerer, Netscape... sont vraiment des clients réseau qui se connectent à un serveur, le serveur X. La tâche du serveur X est de communiquer avec le matériel graphique, de dessiner les images sur votre écran, de lire les entrées souris et clavier. Les clients (vos programmes tels que gimp, Netscape...) envoient les instructions au serveur sur la façon de dessiner les cadres et les boutons. En échange, ils reçoivent du serveur les événements souris et clavier. De toute évidence, vous avez besoin d'un certain type d'authentification, sinon n'importe qui pourrait mettre la pagaille sur l'écran de tout le monde. Deux programmes peuvent contrôler l'accès:



- xhost :

en utilisant ce programme vous pouvez permettre à n'importe quel utilisateur sur une machine donnée d'écrire des graphiques sur votre écran. Exemple : vous êtes assis devant une machine nommée philosophus. Pour permettre à n'importe quel programme sur l'hôte movietux d'accéder à votre écran sur philosophus vous tapez la commande :

`xhost +movietux`

Ceci doit être tapé dans un shell sur philosophus

- xauth : C'est une authentification beaucoup plus sophistiquée basée sur l'usage de clés (cookies). Ici vous pouvez vraiment définir des accès individuels. C'est beaucoup plus sûr que xhost.

L'authentification utilise une clé (cookie) enregistrée dans le fichier `.Xauthority` dans le répertoire home de l'utilisateur. Si la machine distante dispose d'une clé valide dans ce fichier alors l'accès est autorisé. Pour copier la clé de la machine devant laquelle vous êtes assis (philosophus) sur l'hôte duquel vous voulez démarrer le programme (movietux), vous pouvez utiliser une des méthodes suivantes :

`xauth extract - philosophus:0.0 | ssh movietux /usr/X11R6/bin/xauth merge`

ou

scp ~/.Xauthority movietux:

Le procédé à la mise en route d'un programme (client) est le suivant:

1. Le client contrôle la variable d'environnement DISPLAY pour trouver le serveur, à défaut, il essaie de se connecter sur le serveur de cet hôte.
2. Le serveur vérifie que le client est autorisé à lui envoyer des "images". Si le client est reconnu, alors le serveur dessine l'image sur l'écran.

La variable d'environnement DISPLAY a la syntaxe suivante :

```
bash:export DISPLAY=hostname:displaynumber.screennumber
tcsh:setenv DISPLAY hostname:displaynumber.screennumber
```

Je ne parlerai pas ici du displaynumber et du screennumber. Normalement c'est simplement 0:0. Un exemple pour bash serait :

```
export DISPLAY=philosophus:0.0
```

Lancement d'une application à distance

Après toute la théorie voyons maintenant un exemple pratique. Essayez-le. Nous sommes encore assis devant l'ordinateur appelé philosophus et nous voulons lancer quelque chose à distance depuis movietux.

1. Informez votre serveur X que les clients de movietux sont autorisés à dessiner des images sur votre écran :
xhost +movietux
2. connectez-vous à la machine distante movietux :
slogin -l username movietux
3. vous êtes maintenant connecté à movietux et chaque commande que vous tapez est exécutée sur movietux
4. export DISPLAY=philosophus:0.0
5. lancez le programme. Par exemple :
netscape &

Si movietux est une machine puissante, vous noterez que le rendu des pages HTML dans Netscape est beaucoup plus rapide que lorsque vous l'exécutez sur votre propre machine. En même temps vous ne remarquerez pas vraiment, d'un point de vue utilisation, que ce Netscape n'est pas lancé localement car vous le manipulerez de la même manière avec la souris et le clavier.
Sympa, non ?

Variable DISPLAY à emporter

Bien qu'il soit facile d'exécuter une application à distance, il reste ce petit supplément de frappe requis pour définir la variable DISPLAY. Il est possible de l'automatiser :

- Si vous utilisez ssh pour vous connecter à la machine distante, la variable DISPLAY est automatiquement définie. D'autres programmes de connexion à distance supportent la variable DISPLAY mais ssh est très répandu.
- Si vous utilisez slogin ou d'autres commandes, vous pouvez installer les scripts suivants sur la machine distante

Pour tcsh :

```
# emportez votre variable display avec vous lors de la connexion distante :
# mettez-la dans votre fichier ~/.login
set whoami='who -ml'
set remhost='expr "$whoami" : '.*(\.*)''
if ( "$remhost" != "" ) then
setenv DISPLAY "$remhost":0.0
endif
```

Ce script fonctionne en récupérant le nom de l'hôte distant à partir de la commande "who -ml". Cette commande retourne quelque chose du genre

```
>who -ml
movietux!guido pts/3 Oct 26 21:55 (philosophus.tux.org)
```

Si vous utilisez bash, il vous faut le script suivant :

```
# emportez votre variable display avec vous lors de la connexion distante :
# mettez-la dans votre ~/.bash_profile
whoami='who -ml'
remhost='expr "$whoami" : '.*(\.*)''
if [ -n "$remhost" ]; then
DISPLAY="$remhost":0.0
export DISPLAY
fi
```

OpenGL

Si les capacités réseau du système X Window sont très bonnes, les graphiques sont un peu plus lents du fait que vous envoyez les données par un protocole de réseau. Normalement vous ne verrez pas grande différence.

Les applications gourmandes en ressources, telles que les jeux au graphisme intensif sont habituellement basées sur OpenGL (Open Graphics Library) et GLX (extension d'OpenGL au système X Window). Ces bibliothèques fournissent une interface de programmation indépendante du matériel autorisant l'accès direct aux composants d'accélération 3D des cartes graphiques. Autrement dit : l'application envoie la description d'un objet sous forme de points, lignes et

polygones à la carte graphique qui génère tout le rendu en interne. Le graphisme devient très rapide.

Actuellement, la plupart des gestionnaires de cartes graphiques pour Linux ne supportent pas l'accélération GLX/OpenGL pour les applications distantes. Ils ne supportent l'accélération que pour des applications locales. Le résultat, c'est que les applications OpenGL lancées à distance ont du mal à démarrer et sont vraiment lentes. Une exception vient des gestionnaires propriétaires de NVidia. Ils ont une interface de rendu direct qui supporte le rendu indirect pour les applications distantes.

Conclusion

L'utilisation de la puissance de calcul de votre réseau est très facile avec X11. Vous pouvez travailler avec des applications distantes de la même manière qu'avec les applications locales. La seule différence que vous rencontrerez, c'est la possibilité de voir les fichiers et le répertoire home de la machine distante. Cependant, avec NFS et NIS installés, vous pouvez masquer cette petite différence et utiliser la pleine puissance du processeur des machines les plus rapides de vos réseaux sans même y penser.

Links

X11 système, x.org
xfree86.org Le système X11 sous Linux

<p>Site Web maintenu par l'équipe d'édition LinuxFocus © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Guido Socher (homepage) en --> fr: Olivier Pons <pons/at/cnam.fr></p>
---	--