

Exécution d'applications X à distance

par Come David ([Site perso](#)) Oliver Van Hoof (ovh) ([Site perso](#))

Date de publication : 22/06/2008

Cet article a pour but de vous présenter l'exécution d'applications X Window à distance

- I - Introduction
- II - Théorie
- III - Pratique
 - III-a - Configuration du serveur Linux
 - III-b - Configuration du client Linux
 - III-b-1 - En console locale
 - III-b-2 - Connexion sécurisée avec SSH
 - IV-b - Accéder au bureau linux depuis Windows
- IV - Conclusion

I - Introduction

X Window, souvent abrégé X11 ou encore X est le système de fenêtrage sur tous les systèmes Unix et dérivés. Ce système est très puissant car il est totalement indépendant du système d'exploitation sur lequel il tourne ou encore du matériel utilisé par ce système.

II - Théorie

La plus grande caractéristique du système X window est son architecture client/serveur qui permet une totale séparation entre les programmes qui gèrent le matériel et ceux qui doivent afficher des choses à l'écran. Ainsi le client et le serveur peuvent être sur des machines totalement différentes et très éloignées sans que cela ne gêne.

C'est le serveur X qui se charge de la gestion du matériel. Actuellement la principale implémentation de X est Xorg, fork de Xfree86. La gestion du matériel ainsi que des unités d'affichage se fait via le fichier xorg.conf situé dans le répertoire `/etc/X11/`.

Le matériel se décompose en périphériques d'entrée (InputDevice) avec le clavier et la souris, en moniteurs qui représentent les écrans connectés aux différentes cartes graphiques (Device) présent sur la machine. Le regroupement d'un moniteur avec une carte forme un écran. Chaque serveur X de lancé forme ce que l'on appelle une unité d'affichage ou encore DISPLAY.

Chaque unité d'affichage est caractérisée par un identifiant de la forme `nom_machine:numero_sequence` ou `nom_machine/unix:numero_sequence` avec `nom_machine` qui est le nom de la machine sur laquelle elle tourne. Le numéro de séquence est un numéro qui commence à toujours 0 et permet de repérer les différents serveurs X de lancés . Dans l'identifiant, le nom de la machine peut être enlevé signifiant ainsi que l'on est sur localhost. La présence de `/unix` sert en autre à garder une compatibilité avec les anciennes versions mais surtout à dire que l'on va écouter sur un socket unix (donc sur localhost) et non tcp.

Le client X est un simple logiciel graphique qui se connecte au serveur X pour lui envoyer ses requêtes d'affichage via le protocole X au travers de la bibliothèque X (Xlib).

III - Pratique

Dans toute cette partie le serveur aura comme nom david (adresse serv.ath.cx) serveur et le client izu (adresse cli.ath.cx) .

III-a - Configuration du serveur Linux

Pour qu'une personne puisse se connecter à un serveur X, il faut qu'ils partagent entre eux un secret, un magic-cookie pour être précis. En détail, un magic-cookie est une chaîne de 128 bits, soit 32 caractères en hexadécimal. N'importe quelle chaîne répondant à ce critère de longueur peut être utilisée en tant que cookie.

La liste des personnes autorisées à se connecter au serveur X est dans le fichier spécifié via l'option -auth de X. Ce fichier est écrit par le login manager (GDM,KDM,...), il varie donc en fonction de celui ci. La visualisation ou l'édition de ce fichier ne peut se faire que via le programme xauth en root. Pour spécifier le fichier que l'on va utiliser , il faut passer par l'option -f de xauth. Pour le moment regardons qui est autorisé à se connecter à X.

```
debian@debian:~$ xauth -f :/var/lib/gdm/:0.Xauth list
#ffff##:0 MIT-MAGIC-COOKIE-1 43cacbf710e9d4763869bbaa0c17bfac
```

Donc si on décortique tout cela, #ffff## signifie que tout le monde peut se connecter. 0 est le numéro du serveur X. MIT-MAGIC-COOKIE-1 est le protocole d'autorisation utilisé et la longue chaîne est la cookie en question.

Le serveur X, comme tout bon serveur écoute sur un port. De façon générale, un serveur X avec comme numéro de séquence N va écouter sur le port 6000+N. En d'autres termes, le 1er serveur X va écouter sur le port 6000 (son numéro de séquence vaut 0), le 2eme sur le port 6001, ... Si vous vous situez derrière un routeur, il faut donc veiller à ce que ce dernier redirige bien le bon port vers le serveur. Pour cela, je vous renvoie au manuel de ce dernier.

Enfin, il faut aussi s'assurer que le serveur X n'a pas été lancé avec l'option -nolisten tcp, option qui empêche le serveur d'écouter sur les ports. Actuellement, cette option est souvent mise par défaut. Pour le vérifier, exécutons un ps aux | grep X

```
david@debian:~$ ps aux | grep X
root      2855  5.1 11.6 79996 60408 tty7      Ss+  20:13   0:40 /usr/bin/X :0 -audit 0 -auth
          /var/lib/gdm/:0.Xauth -nolisten tcp vt7
```

On voit donc que Xorg a été lancé avec cette option. Pour contourner le problème, 2 options: soit lancer un 2eme serveur X soit enlever l'option du serveur X actuel. Pour la 1ere option, c'est assez simple, il suffit de lancer en root :

```
X :1 vt8
```

Ainsi on lance un 2eme serveur X avec comme numéro de séquence 1 sur le terminal virtuel numéro 8. Il sera accessible via les touches Ctrl-Alt-F8.

Pour la 2eme solution, elle est dépendante du *login manager*. Par exemple pour GNOME, il faut lancer en root `gdmsetup`, puis aller dans l'onglet sécurité, puis décocher la case "Refuser les connexions TCP au serveur X". Sous KDE, il faut éditer le fichier `/etc/kde3/kdm/kdmrc` et retirer `-nolisten tcp` de la ligne `ServerArgsLocal=-nolisten tcp`.

III-b - Configuration du client Linux

III-b-1 - En console locale

La configuration est beaucoup plus simple que celle du serveur. Tout d'abord, on configure `~/Xauthority` via `xauth`. Pour cela, il suffit d'ajouter le serveur via la commande `add`. Son synopsis est le suivant : `add` identifiant typeprotocole cookie en veillant à utiliser le même cookie que celui du serveur. Ainsi, cela donne :

```
david@cli:~$ xauth add serv.ath.cx:0 MIT-MAGIC-COOKIE-1 43cacbf710e9d4763869bbaa0c17bfac
david@cli:~$ xauth add serv.ath.cx/unix:0 MIT-MAGIC-COOKIE-1 43cacbf710e9d4763869bbaa0c17bfac
```

En ensuite, il suffit de tester sur le client en passant à l'option `display` l'identifiant du serveur de la manière suivante:

```
xeyes -display=serv.ath.cx:0
```

Mais l'inconvénient de cette méthode, c'est que l'option `display` n'est pas supportée par tous les programmes. Des lors, une solution un peu plus contraignante mais plus universelle est de fixer la valeur de la variable d'environnement `DISPLAY` avec l'identifiant du serveur. Sous le shell `bash`, la commande est:

```
david@debian:~$ export DISPLAY=serv.ath.cx:0
```

Ensuite, il suffit de tester avec une application quelconque et si tout est bien configuré, vous devriez voir le résultat en quelques secondes.

III-b-2 - Connexion sécurisée avec SSH

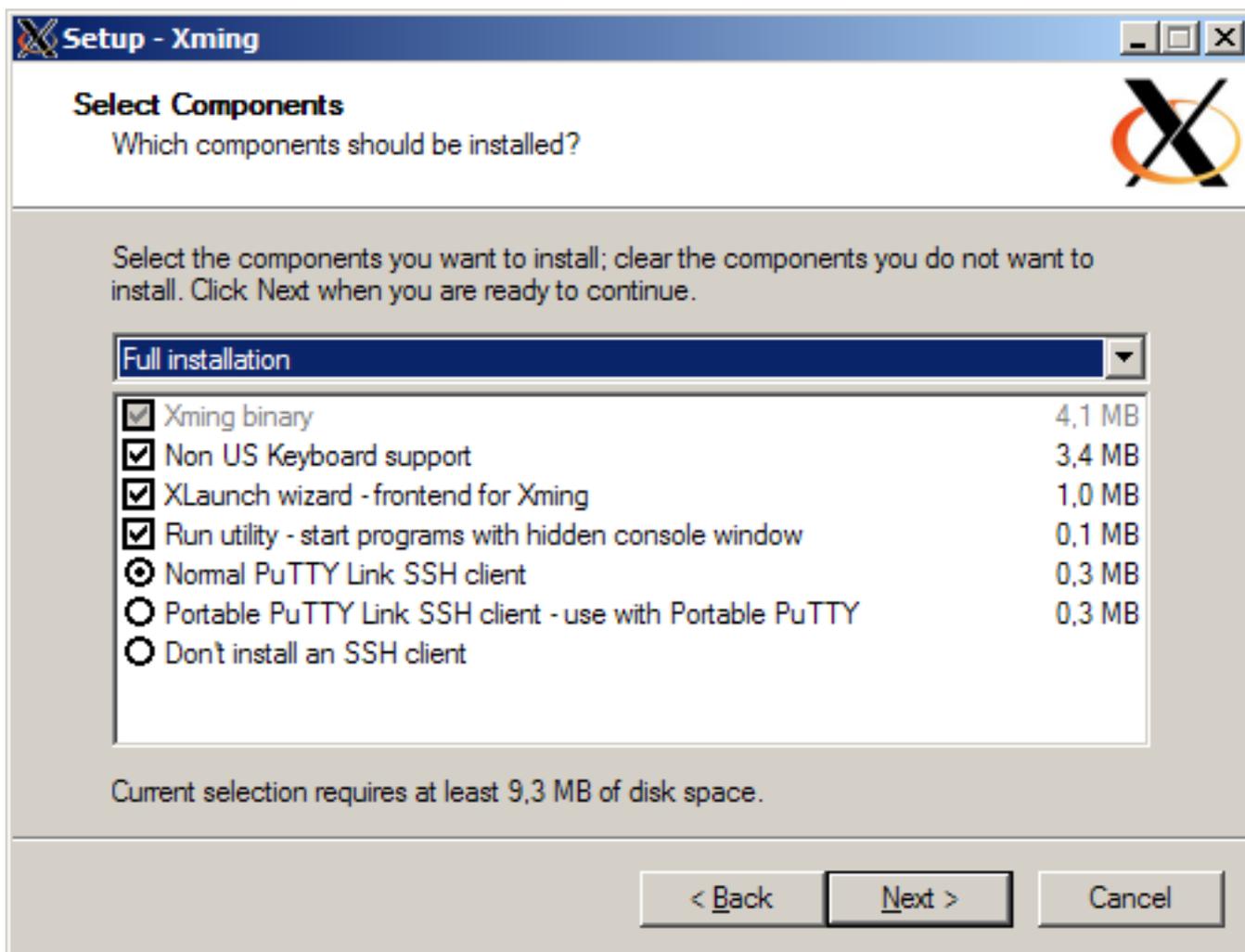
 *Attention, la méthode vue précédemment n'est à utiliser que dans un réseau sûr. En effet, le cookie va transiter en clair sur le réseau. Dès lors, il est très facile de le sniffer pour le récupérer. Si vous souhaitez utiliser cette technique sur internet, utilisez `ssh` pour transférer le cookie même si dès lors la technique devient désuète avec l'option `-X` de `ssh` qui permet d'activer le X11 forwarding, c'est à d'activer la redirection de X11 qui permet alors de lancer de façon naturelle des applications X à distance. On peut aussi noter l'option `-Y` de `ssh` qui sert de joker à X quand celui ne marche pas en activant le `trusted X11 forwarding` (redirection d'X11 forcée). L'inconvénient avec cette dernière, c'est la sécurité qui est moindre puisque cette redirection ne va pas suivre les extensions de sécurité de X11.*

Ainsi avec `ssh`, exécuter une application à distance revient à taper :

```
david@debian:~$ ssh -X test@cli.ath.cx
test@cli.ath.cx:~$ xeyes
```

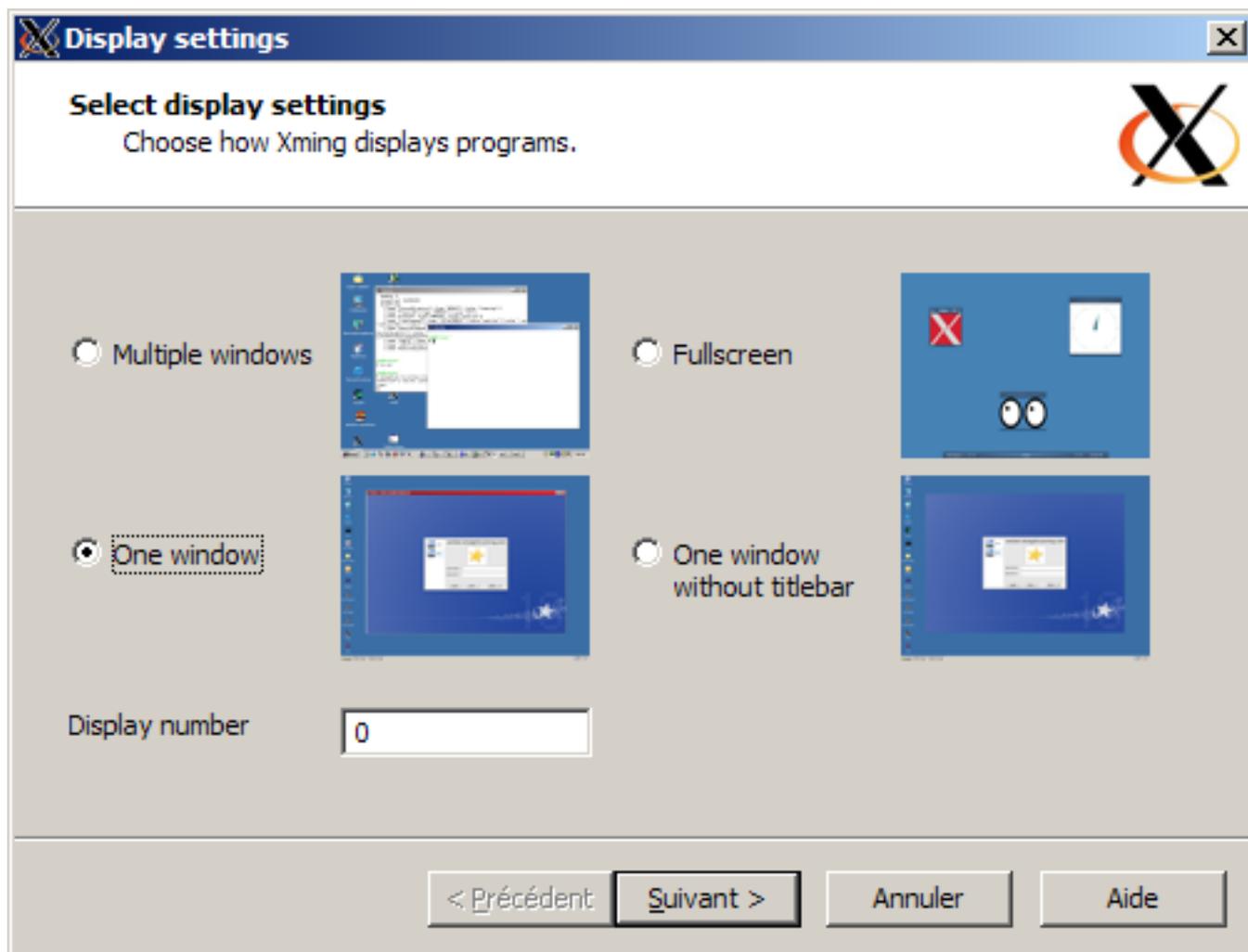
IV-b - Accéder au bureau linux depuis Windows

Le protocole d'accès au serveur X à distance transmet uniquement les instructions d'affichage des fenêtres à travers le réseau, dans le but d'optimiser les transferts de données qui sont ainsi très rapides. Mais cela a pour conséquence que la machine voulant afficher des applications X distantes doit également être équipée d'un serveur X local à même d'exécuter les instructions d'affichage. Il faudra donc émuler un serveur X sous Windows. **Cygwin** en propose un, mais c'est une solution lourde puisque cygwin est une émulation complète de l'environnement linux sous windows. Il existe un projet de serveur X sous Windows appelé **Xming**, c'est cette solution que j'ai choisi ici. D'autre part, l'accès au serveur distant se fait via SSH. Il nous faut donc également un client SSH. Xming intègre un client SSH en utilisant le logiciel libre **Putty** (client ssh sous windows bien connu), mais vous pouvez choisir de ne pas l'installer si vous avez déjà Putty sur votre système. Néanmoins avoir les 2 en parallèle ne pose aucun problème.

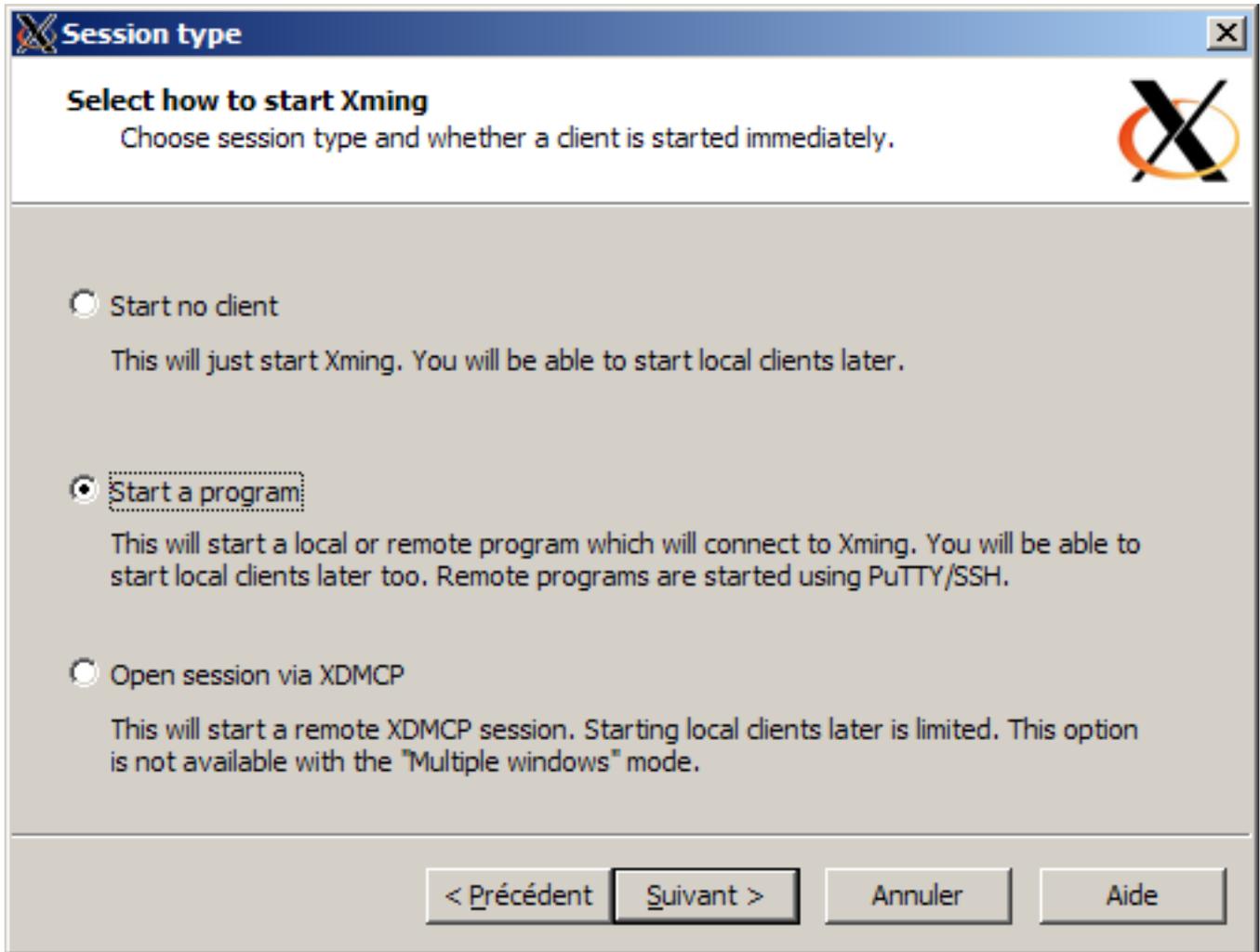


Vous aurez ensuite 2 raccourcis : xlaunch et xming. Xming se contente de lancer le serveur X localement, tandis que xlaunch présente un assistant pour définir la façon dont nous désirons interagir avec le serveur X distant. C'est donc via xlaunch que nous ferons chaque fois la connexion.

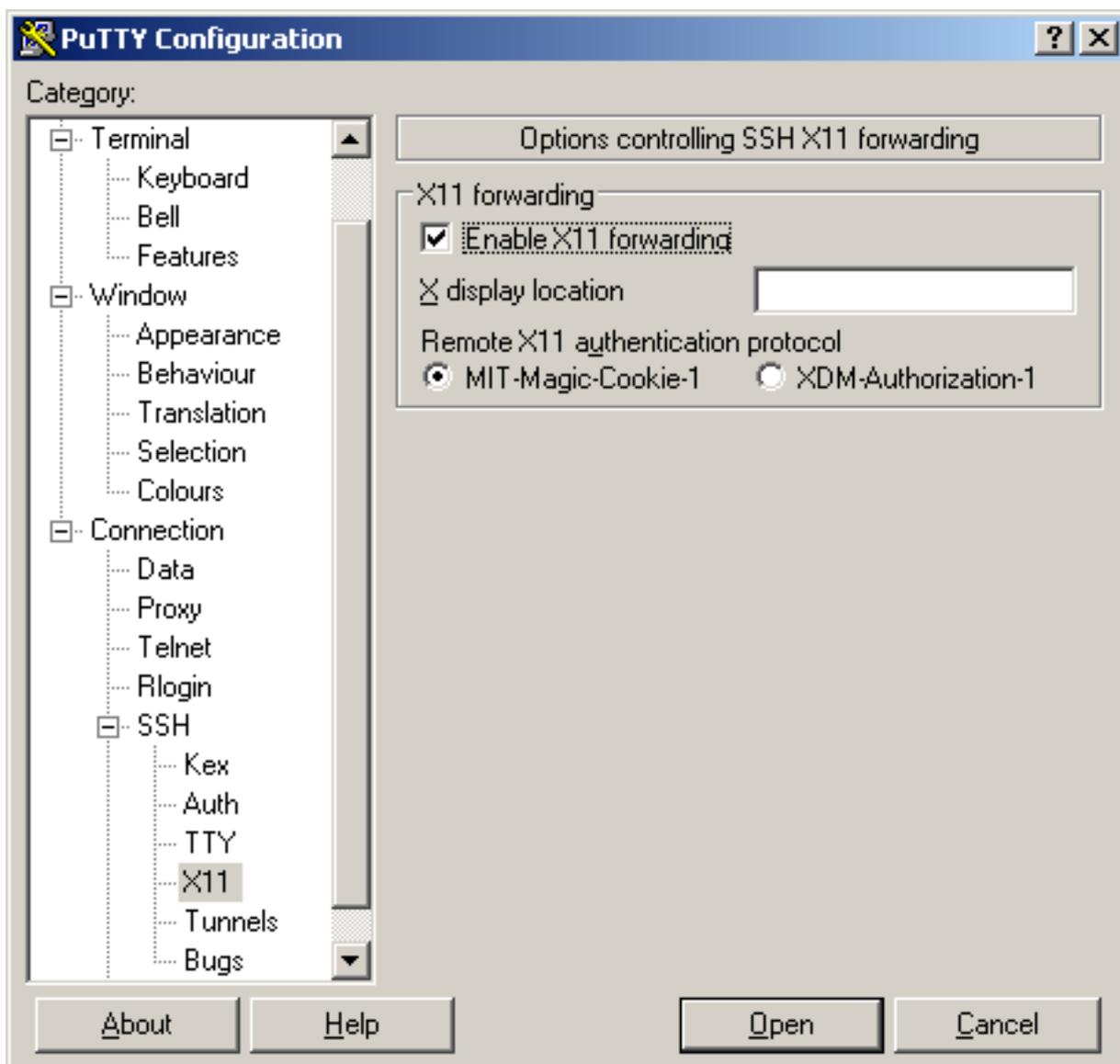
Voici ce qu'il donne au premier lancement :



Vous pouvez donc choisir la façon dont les fenêtres seront affichées. En fait cela va dépendre également de la manière dont on se connecte au serveur X distant. Soit par SSH, à ce moment-là on n'a pas le bureau entier, mais on peut démarrer chaque application graphique une par une en ligne de commande, chaque application s'ouvrant à ce moment-là dans une fenêtre séparée. Soit en utilisant le protocole XDMCP qui permet d'ouvrir une session entièrement graphique qui reproduit tout le bureau X (équivalent du **Terminal Server** sous Windows). L'écran suivant permet d'ailleurs de sélectionner le mode souhaité :

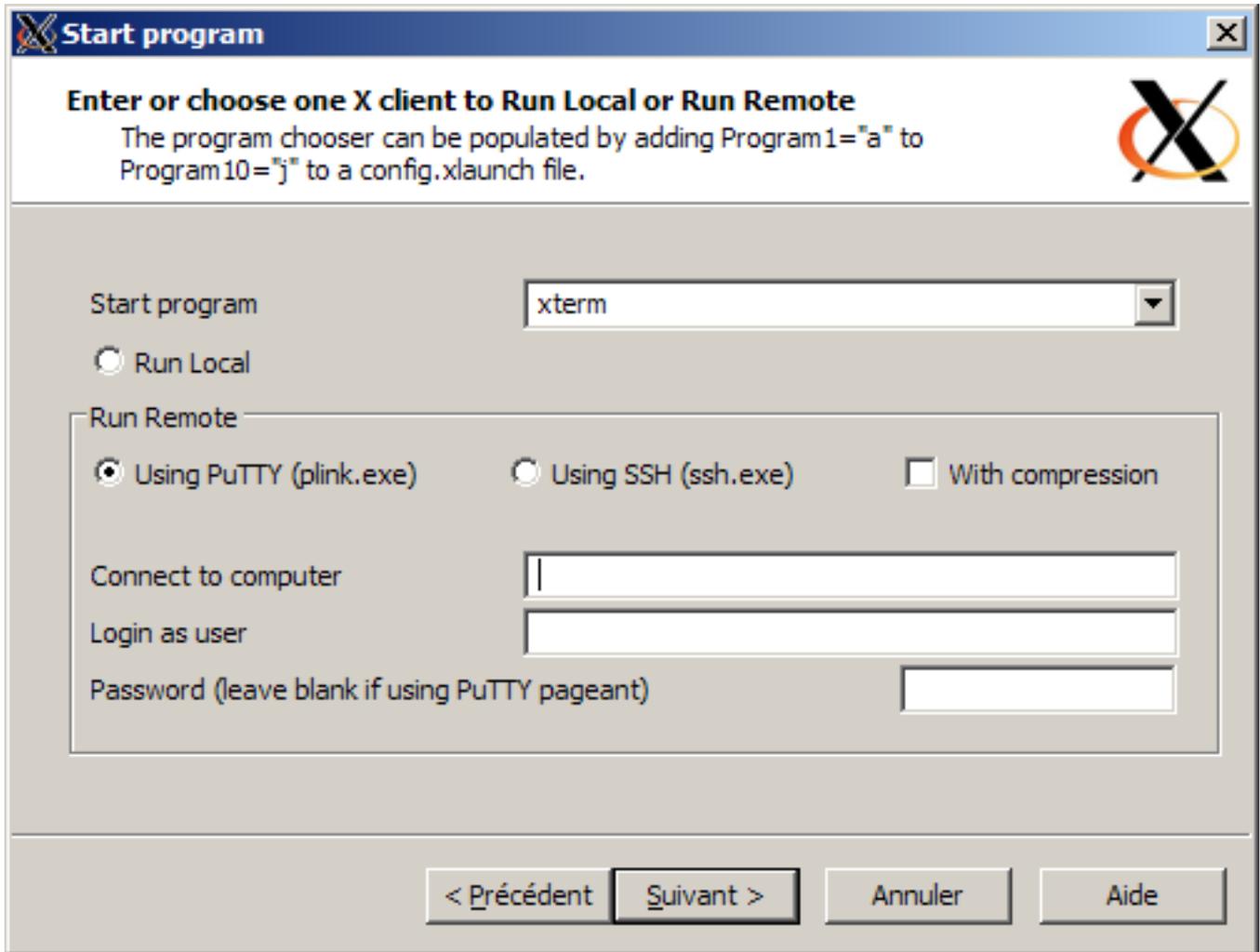


- **Start no client** : démarre le serveur X en mode démon seulement, nécessite de démarrer un client SSH soi-même (par exemple Putty) pour effectuer la connexion et lancer ensuite les programmes graphiques souhaités via la console. Attention dans Putty il faut activer le **X11 forwarding** dans les options de votre connexion :



- **Start a program** : utilise le client SSH intégré de Xming pour effectuer la connexion distante au serveur X linux. Cela évite de lancer putty séparément.
- **Open session via XDMCP** : ici même le login sera graphique et géré par le serveur X distant; vous aurez en outre accès à l'entièreté du bureau comme si vous étiez physiquement devant la machine. C'est probablement le mode le plus pratique d'utilisation. Attention ne soyez pas surpris : dans ce mode le délai entre le login graphique et l'apparition du bureau est assez long (parfois plus d'une minute).

Si vous avez sélectionné le mode "start a program", Xming propose différents clients SSH, prenez l'option par défaut qui consiste à utiliser le client Putty intégré au package Xming. Vous devrez ici mentionner les informations de connexion (adresse du serveur, login et mot de passe).



The screenshot shows a window titled "Start program" with a close button in the top right corner. The main text reads: "Enter or choose one X client to Run Local or Run Remote". Below this, a note states: "The program chooser can be populated by adding Program1='a' to Program10='j' to a config.xlaunch file." To the right of this text is the X logo. The dialog is divided into two main sections: "Run Local" and "Run Remote". In the "Run Local" section, there is a "Start program" dropdown menu currently set to "xterm". The "Run Remote" section is active and contains three radio buttons: "Using PuTTY (plink.exe)" (selected), "Using SSH (ssh.exe)", and "With compression" (checkbox). Below these are three text input fields labeled "Connect to computer", "Login as user", and "Password (leave blank if using PuTTY pageant)". At the bottom of the dialog are four buttons: "< Précédent", "Suivant >", "Annuler", and "Aide".

Vous pouvez spécifier des paramètres supplémentaires en cas de besoin, normalement vous pouvez laisser vide :

Additional parameters

Specify parameter settings
Enter clipboard, remote font server, and all other parameters.

Clipboard **No Access Control**
Start the integrated clipboard manager Disable Server Access Control

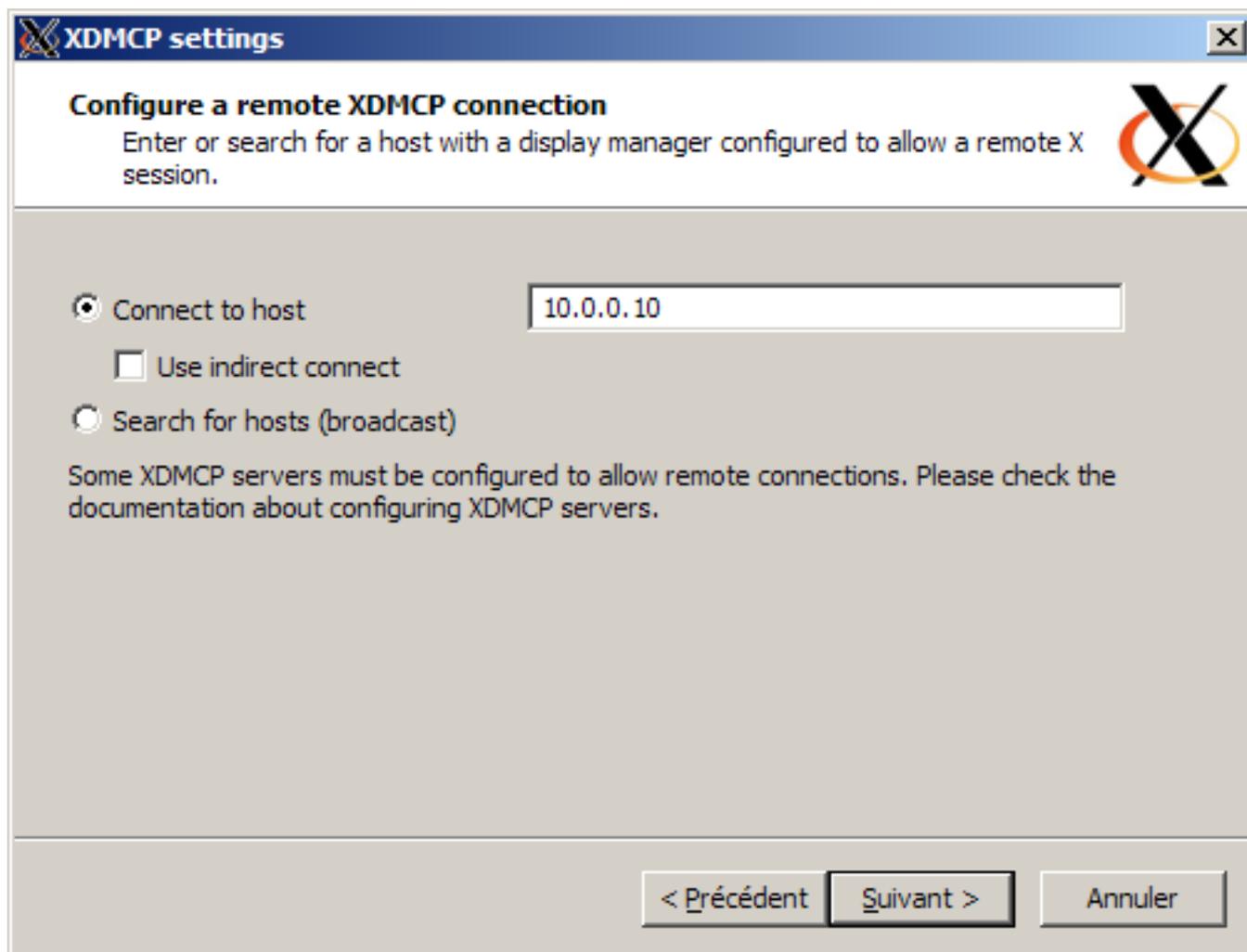
Remote font server (if any)

Additional parameters for Xming

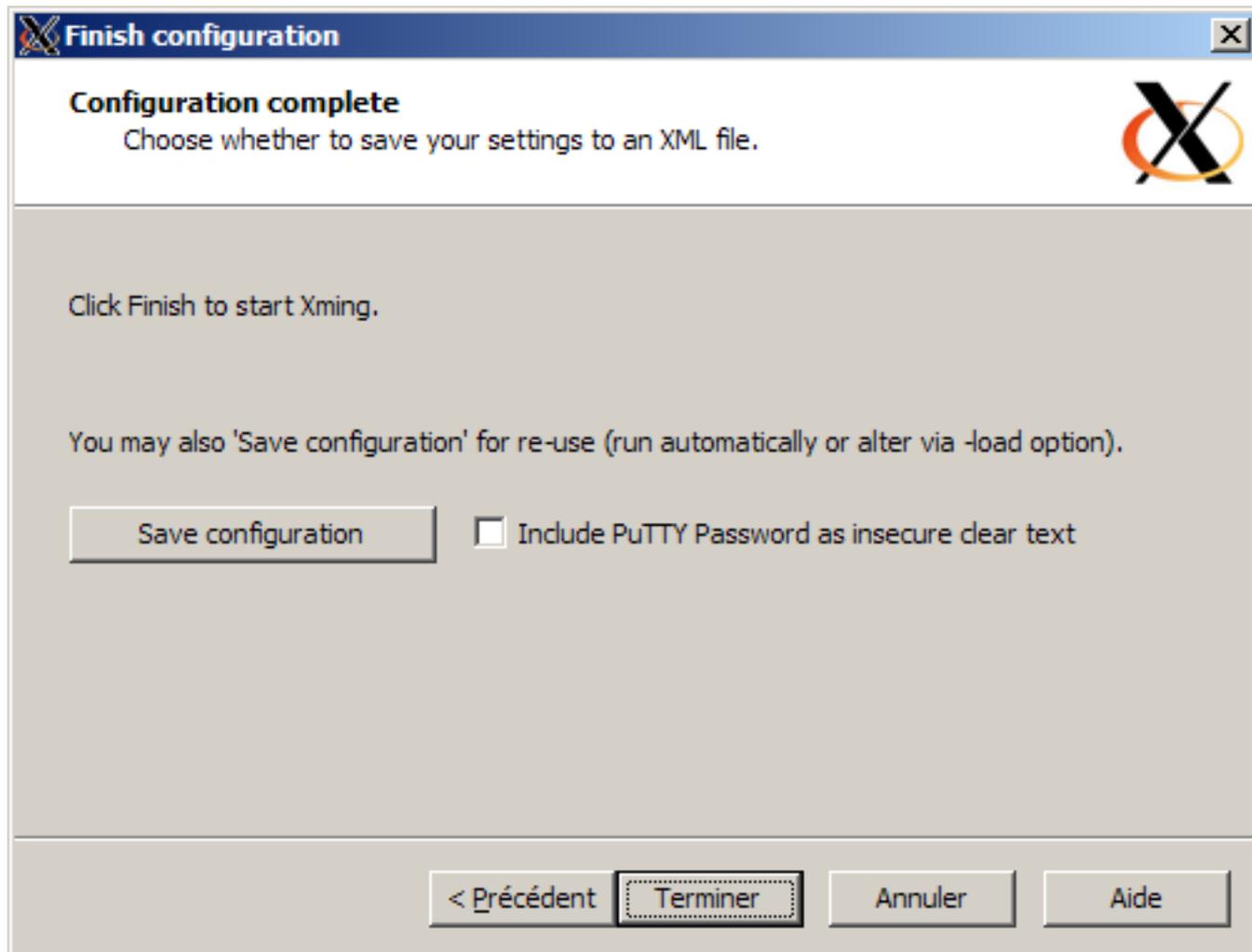
Additional parameters for PuTTY or SSH

< Précédent Suivant > Annuler Aide

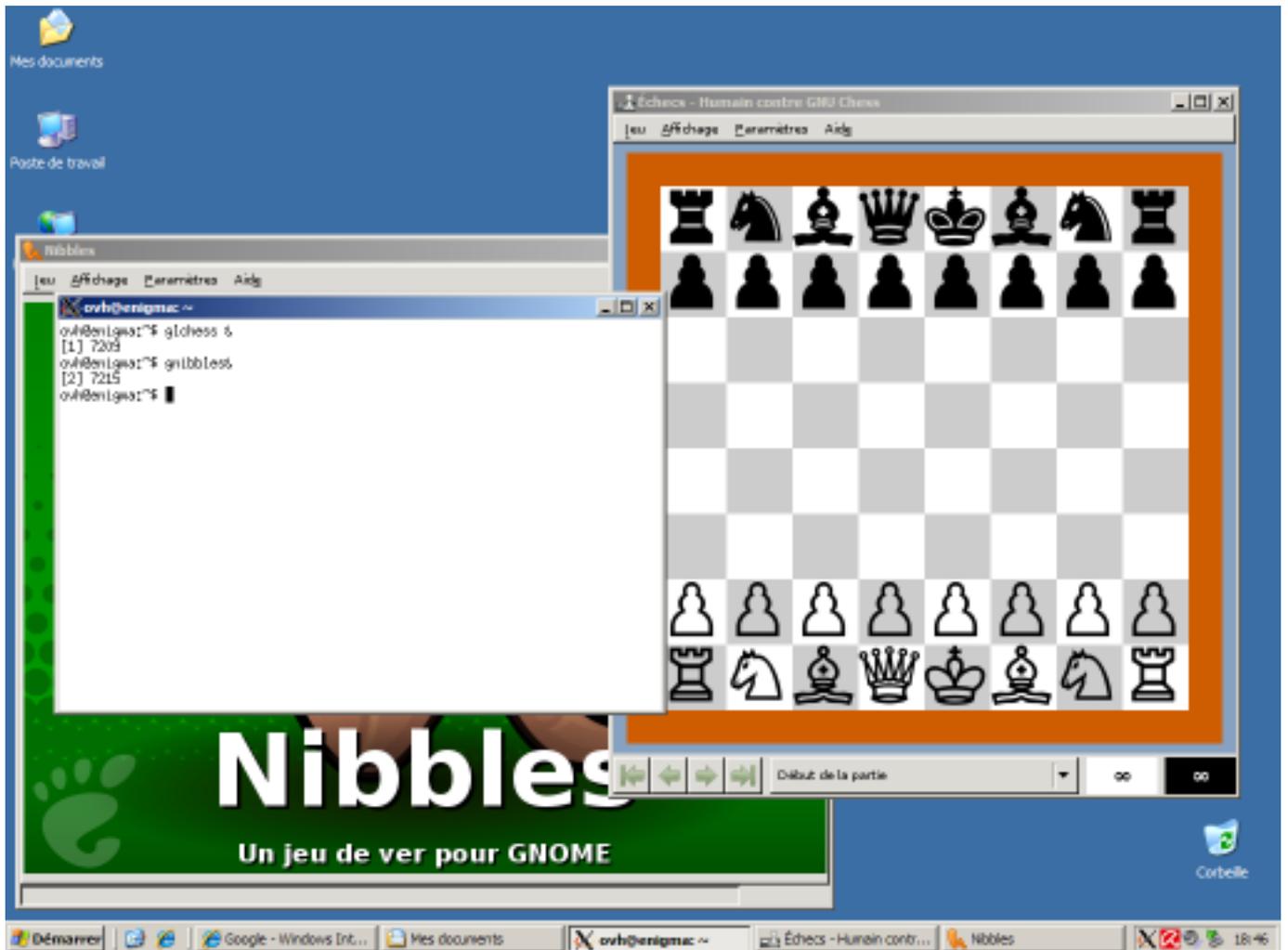
Dans le cas d'une connexion XDMCP, vous devrez seulement préciser l'adresse du serveur où se connecter puisque l'authentification se fera au travers du serveur X :



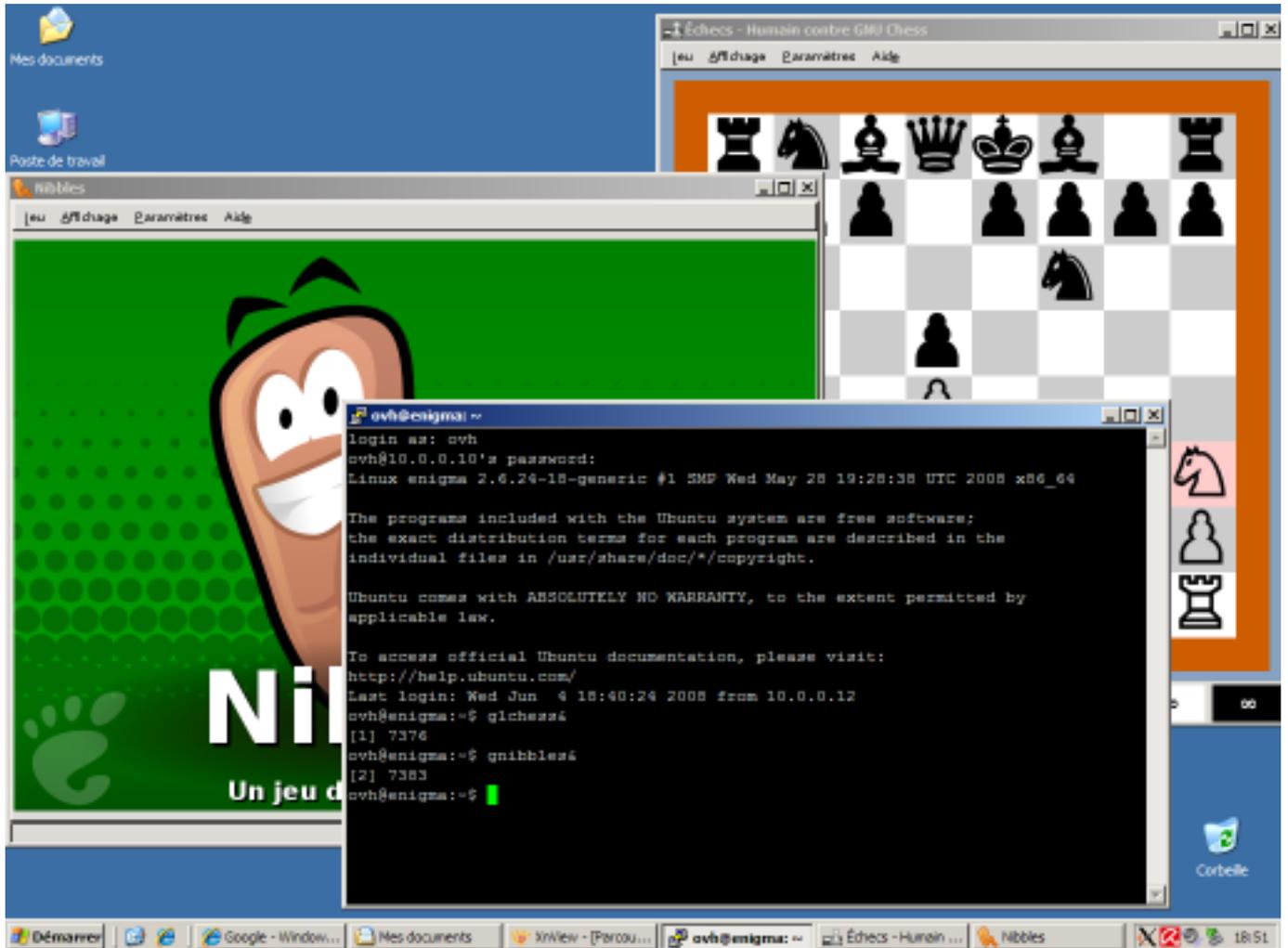
L'écran final permet de sauvegarder la configuration pour un prochain lancement et de démarrer le serveur X local :



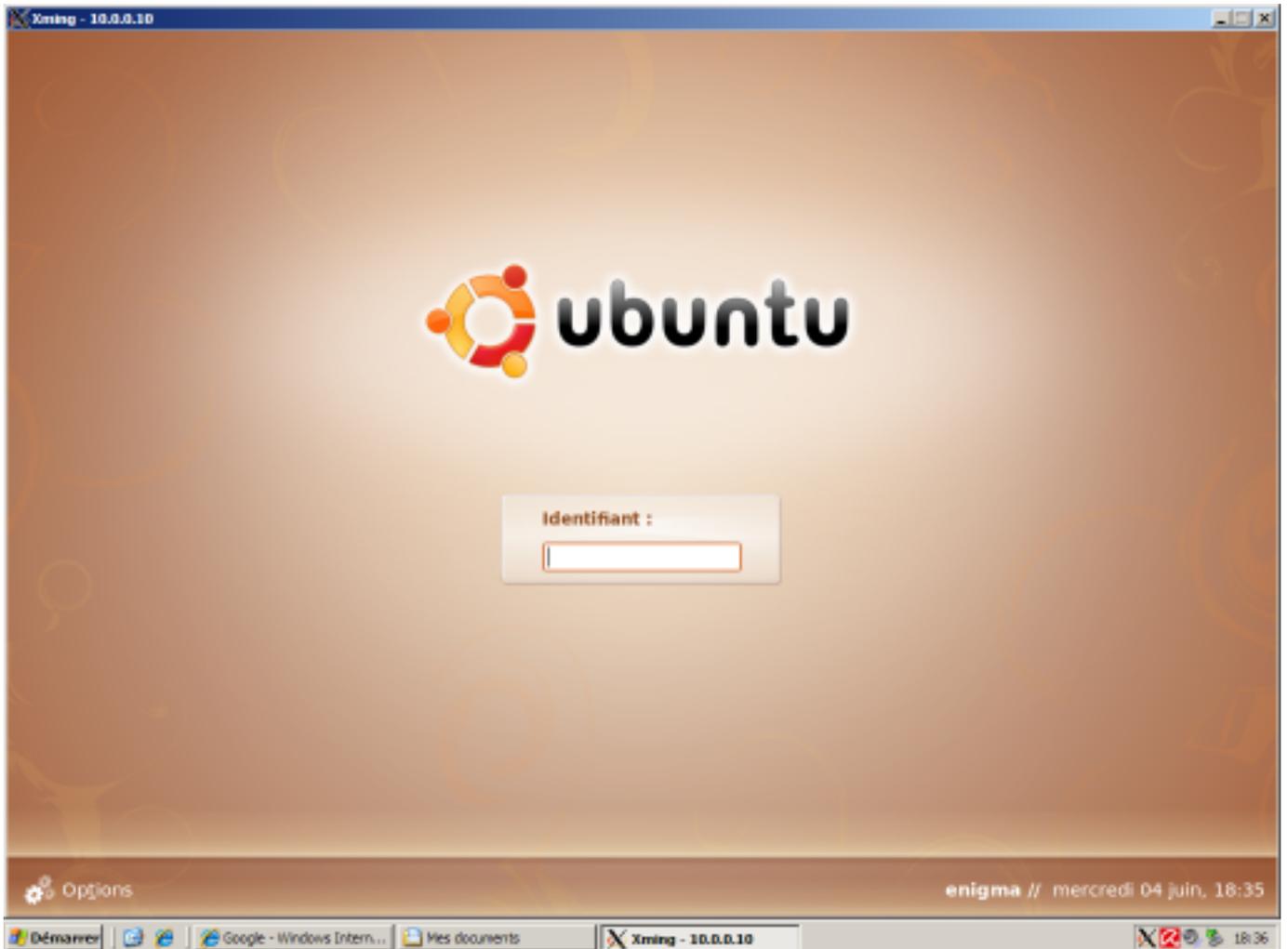
La configuration est maintenant terminée, voici les différents résultats suivants les modes de connexion (cliquez sur chaque image l'agrandir en taille réelle) :



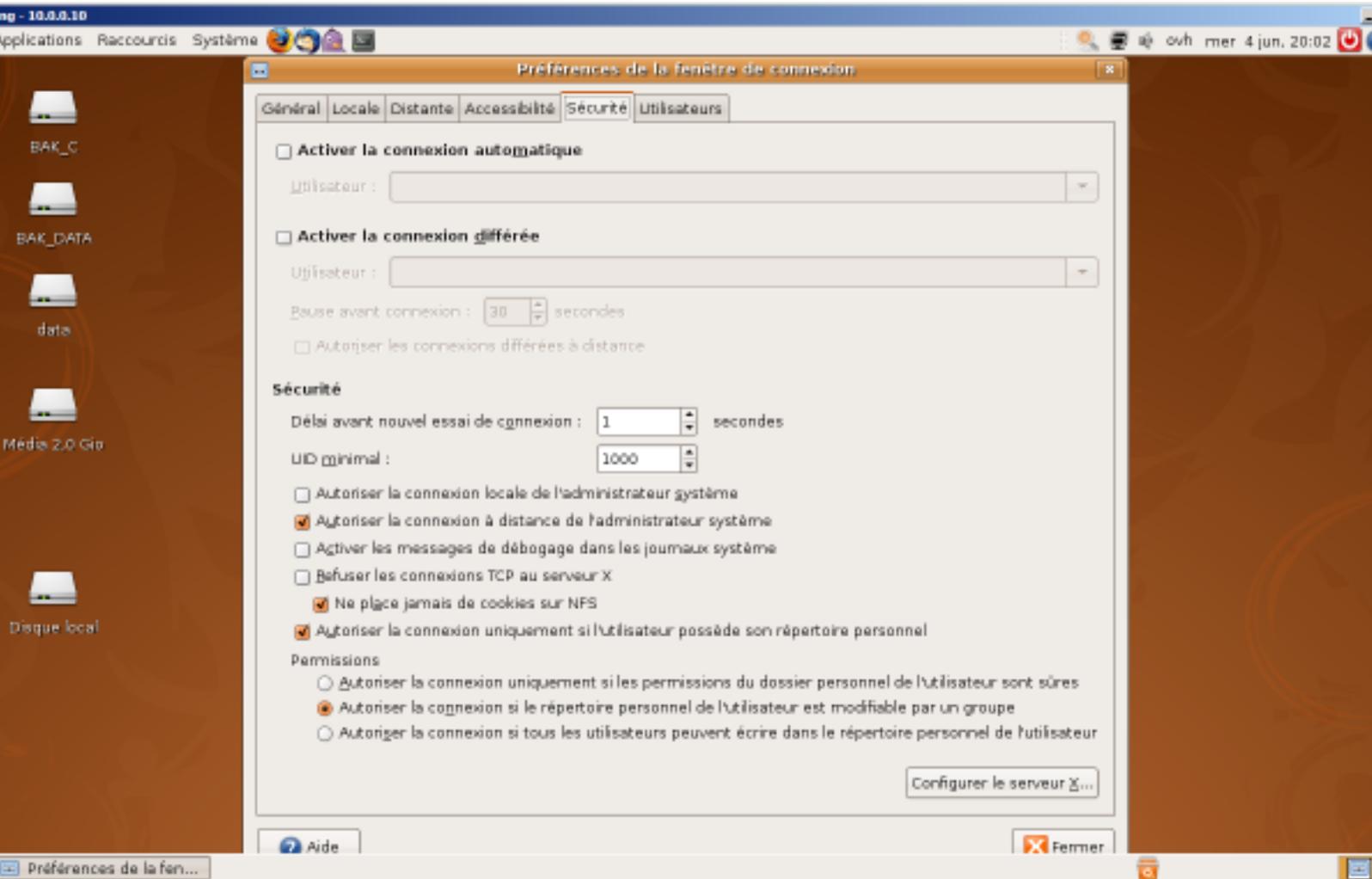
Connexion avec client SSH intégré



Connexion avec client Putty lancé à la main



Login graphique en XDMCP



Bureau complet en XDMCP, permet de voir au passage la configuration à effectuer pour le serveur X sous Ubuntu Linux 8.04.

IV - Conclusion

J'espère qu'avec cet article, vous avez pris conscience de la puissance X ainsi que des applications que l'on peut en faire.

Remerciements: je souhaite remercier remram44 pour son aide en général ainsi que Heureux-oli pour la relecture qu'il à effectuée.

