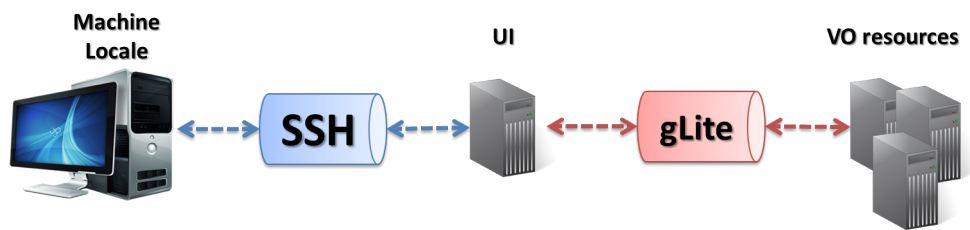


TP2 Grille de Calcul 3^{ème} année F2

Soumission de jobs simples

Préambule

La suite *gLite* est installée sur les machines de type UI (User Interface). Les TP de grille ne peuvent donc être réalisés que sur ce type de machine. Le CRRI nous fournit une machine de ce type à laquelle vous pourrez accéder via cette URL : *193.55.252.161* et les identifiants fournis en TP. La manipulation mise en place dans le premier TP pour vous permettre de vous connecter en SSH sans mot de passe peut être répétée sur l'UI en guise de révision :)



Connexion à l'infrastructure de grille

1 Authentification avec un proxy

Les commandes manipulant des objets sur la grille (jobs, fichiers, ...) impliquent que le propriétaire des ces éléments soit identifié et autorisé à accéder à la grille de calcul. Pour ce faire, on utilise les *proxies*, qui sont des représentations du certificat de l'utilisateur. Ils sont accessibles sans mot de passe et existent pour une durée déterminée et **courte**. La création et la gestion des proxies se fait par l'intermédiaire du serveur *VOMS*, qui vous a été présenté en cours.

Traditionnellement, les utilisateurs de la grille créent leur proxy à partir de leur certificat personnel, afin de le mettre à disposition des commandes *gLite*. Dans notre cas, vous ne disposez pas de certificat, ce qui peut apparaître comme un problème à première vue. Cependant, nous allons outrepasser les règles de base de sécurité et partager une authentification commune. Le proxy que vous utiliserez durant cette séance sera déjà créé. Vous devrez simplement vous l'approprier pour pouvoir l'utiliser :

```
cp /tmp/x509up_u652 $X509_USER_PROXY
chmod 600 $X509_USER_PROXY
```

Quelle commande vous permet de vérifier que la manipulation a fonctionné ?

2 Soumission basique de jobs

1. Écrivez un script permettant d'afficher :
 - le nom de la machine sur laquelle il s'exécute
 - ses caractéristiques de mémoire et de processeur
 - le nom d'utilisateur
 - le contenu du répertoire courant
 - la liste des processus en cours d'exécution sur la machine
2. Vérifiez que votre script fonctionne sur l'UI.
3. Écrivez le JDL permettant d'exécuter ce script sur la grille.
4. Soumettez le job sur la VO *biomed* et **conservez précieusement son identifiant** (une option de la commande de soumission vous sera utile).

3 Monitoring de jobs

1. Quelle commande permet de surveiller l'état d'un job ?
2. Comment pouvez-vous surveiller l'état de plusieurs jobs à l'aide de cette commande ?
3. Une fois votre job terminé, récupérez ses résultats.

4 Contraintes des exécutables sur la grille

1. Récupérez le fichier source C `myExe.c` situé dans le répertoire `/home/jopasserat/enseignement/tp2/`
2. Compilez ce fichier. Pour cela, vous aurez besoin :
 - de la bibliothèque TestU01 (présente dans le répertoire `/home/jopasserat/enseignement/tp2/tools/install_testu01`)
 - du fichier objet (déjà compilé) `/home/jopasserat/enseignement/tp2/tools/tinymt32.o`
 - du fichier d'en-tête `/home/jopasserat/enseignement/tp2/tools/tinymt32.h`
3. Écrivez le JDL permettant d'exécuter ce programme sur la VO *biomed*. Il prend en paramètre l'entier correspondant à votre login (pour `zz01` → 1).
4. Récupérez les sorties de votre job. Que constatez-vous ?
5. Identifiez le problème et résolvez-le de 2 manières (le contenu du répertoire `/home/jopasserat/enseignement/tp2/tools/` devrait vous servir...).
6. Comparez votre résultat avec ceux de vos camarades de classe en envoyant un job pour chaque login utilisé dans la salle.