

1 Généralités

Question 1.1. Quelles fonctionnalités supplémentaires un middleware de grille comme gLite apporte de plus qu'un gestionnaire de jobs tel que Torque ?

Question 1.2. Comment sont organisés les utilisateurs et les ressources ? Pourquoi ?

Question 1.3. On souhaite mettre en place une infrastructure de grille privée (indépendante de l'infrastructure de grille européenne EGI) basée sur le middleware gLite. Quels sont les éléments indispensables à déployer pour avoir une grille fonctionnelle (justifier chacune de vos réponses) ?

Question 1.4. Quel est l'intérêt du système d'information pour l'utilisateur final ? Quel est le fonctionnement des 2 variables Requirements et Rank dans le JDL (donnez des exemples) ?

2 Simulations sur grille

On considère dans la suite une simulation paramétré par 2 valeurs entières. L'exécutable du moteur de simulation `mon_simulateur` prend en paramètre un fichier décrivant le modèle de simulation et les deux paramètres du modèle :

```
mon_simulateur modele1.desc 3 5
```

Le simulateur génère un seul fichier de résultat `modele1.out`.

Question 2.1. Ecrivez le script PBS du job et la commande pour soumettre une simulation (avec des paramètres fixés) sur un cluster Torque/PBS.

Question 2.2. On suppose que l'exécutable, le fichier d'entrée et le fichier résultat sont de petites tailles.

a. Ecrivez le JDL et la commande pour soumettre une simulation (avec des paramètres fixés) sur la grille EGI.

b. Quelle est la commande pour suivre l'état du job ? Quand et comment récupère-t-on le fichier résultat ?

Question 2.3. On se place dans le cas où une simulation a besoin de 3 Go de RAM.

a. Comment poser cette contrainte dans la soumission sur cluster ?

b. Sur grille ?

Question 2.4. On suppose désormais que le fichier résultat est volumineux (> 1Go).

a. Modifier en conséquence votre JDL et le script du job.

b. Quand et comment récupère-t-on le fichier résultat ?

3 Communication entre jobs

En général, les WN ne sont pas accessibles directement depuis l'extérieur : il est donc impossible aux jobs déployés sur la grille de communiquer directement entre eux. On souhaite mettre en place une zone d'échange accessible par tous les jobs qui peuvent y déposer et lire des messages.

On propose d'implémenter un tel mécanisme de communication à l'aide de fichiers et de répertoires sur la grille. Un message d'un job i vers un job j sera représenté par un fichier déposé dans un répertoire (sa boîte aux lettres) créé par le job j au début de son exécution.

Question 3.1. Ecrire la partie initiale du script du job i qui crée sa boîte aux lettres dans le LFC.

Question 3.2. Ecrire la partie du script du job i qui dépose un message au job j . Prêtez attention aux points suivants :

- les jobs ne sont pas nécessairement synchrones,
- il ne faut pas que plusieurs jobs puissent écrire dans le même fichier.

Question 3.3. Ecrire la partie du script du job j qui relève ses messages. Comment éviter de relever plusieurs fois les mêmes messages ?

Question 3.4. Comment un message peut-il faire un broadcast (un message à destination de tous les autres jobs) ? Peut-on s'affranchir de transférer plusieurs fois le message ?

Question 3.5. Modifier votre solution pour que le mécanisme soit tolérant à la panne du SE. Quel est le point unique de défaillance (*Single Point of Failure*) de notre solution ? Est-ce spécifique à ce mécanisme ?