Exercices 2ième partie

Modification et édition des fichiers JDL

Nous allons maintenant utiliser les sandbox pour importer et exporter des données.

1) Modifier le fichier HelloWorld.jdl pour qu'il n'utilise plus /bin/echo mais les script HelloWorldScript.sh pour cela :
   la ligne Executable doit être HelloWorldScript.sh, la ligne Argument peut rester avec Hello World.
   Vous devez définir le paramètre InputSandbox pour transférer le script du job.

```
InputSandbox = {"HelloWorldScript.sh"};
```

Le job HelloWorld-2.jdl vous donne la solution.

A quoi sert l'inputSanbox ? Excécuter le job et vérifiez que tout fonctionne

Remarque: On peut utiliser n'importe quel script (perl, python,bash ...) cependant le shell utilisé par le script (indiqué par la ligne “#!”) doit exister dans le worker node

2) Pour voir un exemple de récupération d'output par l'output sandbox, on va utiliser le job Output.jdl et le script Output.sh

Comprendre Requirement et Rank

Requirement

```
glite-wms-job-list-match  helloWorld.jdl
```

L'attribut Requirement permet de passer au WMS un ensemble de conditions que rempliront les sites sélectionnés.

Par exemple ajoutez l'expression suivante dans le fichier HelloWorld.jdl pour sélectionnez les sites qui permettent d'avoir plus de 1h de CPU.

```
Requirements = (other.GlueCEPolicyMaxCPUTime >60);
```

L'attribut requirement peut être précisé pour l'ensemble des informations que publie un site (Version OS,librairie disponible ...) Par exemple pour choisir des sites spécifiques on peut utiliser le nom de la ressource comme pré-requis. Pour choisir des sites in2p3, changez la valeur de Requirements comme suit:

```
Requirements = RegExp(“.*\.in2p3.fr:.*”,other.GlueCEUniqueID);
```

Combien de sites avez vous trouvez.
Modifiez la ligne Requirements pour avoir un site in2p3 et plus de 1h de CPU.
Vous pouvez aussi modifiez le Requirement pour tourner au CPPM.(ou sur le site de votre choix).

Rank

L'attribut Rank permet de définir la règle qui sera appliquée pour classer les sites qui répondent aux conditions des pré-requis.
Par exemple pour utiliser la ressource avec le plus de CPU libre, on utilise

Rank = other.GuCEStateFreeCPUs

Environnement sur le Worker Node

Chaque utilisateur de la grille est mappé dans un compte local sur chaque site

a) Générez un jdl permettant de voir sous quelle machine tourne le job (solution Hostname.jdl)
Lancez le job et récupérez l'output. Regardez le fichier std.out. Sur quel compte êtes vous mappé? Comparez avec vos voisins.
b) Visualisez le contenu du script Printenv.jdl.
Soumettez un job qui lance ce script dans la grille . Regardez la liste des variables.

Soumission d'un job MPI

Beaucoup de disciplines utilisent des jobs parallèles. MPI(Message Passing Interface) est un protocole qui permet la communication entre les taches parallèles.
Regardez dans le fichier MPI.jdl MPI.sh (les adeptes de MPI peuvent aussi regarder hello.c)

Il y a 2 paramètres spéciaux pour les jobs MPI: JobType et Node Number.Le NodeNumber est le nombre de CPUs réellement utilisés par le job.
Regardez le script MPI.sh. Le script permet la compilation du code MPI, la vérification de l'existence de la liste des machines et le lancement du job en parallèle.
Vérifiez combien de sites sont suceptible de répondre

Utilisation des nouvelles options du WMS

Soumission de collections de jobs

Objectif: soumettre plusieurs jobs en les gérant comme un seul. Regardez les jobs qui se trouvent dans le répertoire Jobcollection, on retrouve le job HelloWorld, un job qui excecute hostname et un dernier qui fait pwd.
Depuis le répertoire Jobcollection;

glite-wms-job-submit --collection . -o jobid -d delID

avec delID obtenu avec glite-wms-job-delegate-proxy -d delID

glite-wms-job-status -i jobid

Quand le job est fini on récupère les sorties.
glite-wms-job-output –dir result -i jobid

Notez la présence des 3 répertoires correspondant au 3 jobs.

On peut aussi construire un jdl plutot que de mettre les jobs dans un répertoire. Voir JobCollection.jdl

Soumission de jobs multiparamètres

Parfois seul le jeu de paramètres d'un job change, plutôt que d'envoyer x jobs pour couvrir x jeux de paramètres et donc gérer x jobs, on peut envoyer/gérer un seul job.

Ouvrez le job JobPara.jdl et essayez de comprendre ce qu'il doit faire.
Soumettez le

glite-wms-job-submit --o jobid -d delID JobPara.jdl –dir result