

Portage d'applications sur EGI:
Retour d'expériences sur
Le middleware et quelques extensions

David Weissenbach, IE CNRS / IPGP
david.weissenbach at upmc.fr

Université de Pau et des Pays de l'Adour
3 décembre 2010

le middleware

le JDL

le middleware

Quelques extensions

applications orientées job management

les environnement complets

Soumission d'un job: le JDL

Sur EGI, un job est représenté par un fichier additionnel en JDL (job description language)

Ce n'est qu'une suite de clauses

Attribut = Valeur ;

Une fois pris en charge il est identifié par un [jobid](#).

`https://<LB_hostname>:<port>/<unique_string>`

le JDL le plus simple?

```
StdOutput = "/dev/null";  
StdError = "/dev/null";  
Executable = "/bin/bash";  
Arguments = "-version";
```

Pas de jobs interactifs.

Un JDL utile pourrait comporter des `Requirements`.

Donnant à l'utilisateur une liberté totale

```
StdOutput = "out.txt";  
StdError = "err.txt";  
Executable = "/usr/bin/perl";  
Arguments = "mywrapper.pl";  
InputSandbox= {"mycommand.pl", "mypkg.tgz"};  
OutputSandbox= {"out.txt", "err.txt"};
```

On peut utiliser des serveurs GRIDftp pour les Sandboxes.
Les Sandboxes sont de taille réduite, mais le job peut utiliser le
data management gLite, internet ...

dans environnement très contraint

Les contraintes ont 2 origines:

la grille

- ▶ mauvais batch
- ▶ ressources floues
- ▶ complexité / goulets (petites VOs)

l'application

- ▶ code & services
- ▶ *flow* de l'application
- ▶ disponibilité pour l'adaptation

les points forts du middleware

les facilités du job management offertes par le middleware (JDL étendu pour collections, jobs paramétriques ou à dépendances) sont suffisantes et immédiatement ou rapidement¹ très rentables pour:

- ▶ les applications *Monte Carlo* ou statistiques
- ▶ l'exploration d'espaces de paramètres
- ▶ l'exploitation de grands jeux de données

¹encadrées dans des scripts et ou jobs cron sur l'UI

les limites middleware

Il persiste(ra?) toujours un petit taux de jobs échoués.
En se donnant quelques moyens logiciels supplémentaires on peut utiliser l'application *gridifiée* dans des situations opérationnelles.

Le debogage puis le réglage de l'application se déroulent dans un environnement plutôt opaque.

Les applications accédant à des bases de données, les codes MPI nécessitent plus de soins.

les revers du middleware

Les jobs doivent être biens proportionnés (en France de 16 à 20 heures et jusqu'a 42 pour l'Europe).

L'environnement d'exécution sur le Worker Node est assez spartiate². La VO peut disposer de zones permanentes ou installer du code.

Les 4 expériences du LHC utilisent toutes leurs extensions au job management classique via les jobs pilotes et les VObox.

²on trouvera C,C++,jdk,perl,python,tcl,...

GANGA, DIANE

Ces deux outils sont basés sur python et sont d'une approche simple.

GANGA: ganga est un ensemble de classes, une API et une surcharge de l'interpréteur python orienté job management offrant une alternative à la ligne de commande standard. Il peut utiliser un batch local et gLite.

<http://ganga.web.cern.ch/ganga/>

DIANE est un moteur pour faire tourner des jobs grilles en mode worker/master (le job est un worker qui effectue toujours la même tâche avec des paramètres fournis par master).

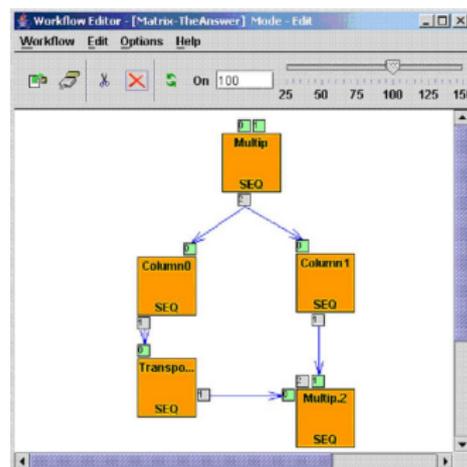
[http:](http://it-proj-diane.web.cern.ch/it-proj-diane/)

[//it-proj-diane.web.cern.ch/it-proj-diane/](http://it-proj-diane.web.cern.ch/it-proj-diane/)

le portail p-grade

p-grade est un portail graphique permettant de facilement construire des workflows (semblables au DAGs gLite) sur Grille.

Les codes peuvent-être scalaires ou MPI.



On peut aussi:

- ▶ conduire des études paramétriques;
- ▶ utiliser des m/w gLite ou globus;
- ▶ utiliser du code hors grille (vu comme un service externe) dans le workflow via le moteur additionnel GEMMLCA

<http://p-grade.hu>

g-Eclipse

g-Eclipse dérive de l'environnement de programmation eclipse.
Il contient entre autres:

- ▶ le management d'un cluster local
- ▶ le job management
- ▶ un debugger
- ▶ une IHM pour manipuler ses fichiers
- ▶ ...

<http://www.geclipse.org>

DIRAC

DIRAC (Distributed Infrastructure with Remote Agent Control) est un environnement complet qui se place entre l'utilisateur et le middleware: on soumet ses jobs et ses opérations sur les données à DIRAC plutôt qu'au middleware et DIRAC s'en occupe.

On s'assure ainsi de plus de robustesse dans ces opérations essentielles.

DIRAC propose un portail web, une API et un jeu de commandes pour orchestrer tout ceci.

`http://lhcbweb.pic.es/DIRAC/`

`http://mareela.in2p3.fr:`

`9200/dirac/wiki/DIRAC/Tutorials`

Références, liens utiles

FranceGrilles: <http://www.france-grilles.fr>

gLite user manual:

<https://edms.cern.ch/document/722398/>

Goc Wiki: [http:](http://goc.grid.sinica.edu.tw/gocwiki/User_Guides)

[//goc.grid.sinica.edu.tw/gocwiki/User_Guides](http://goc.grid.sinica.edu.tw/gocwiki/User_Guides)

Merci pour votre attention !

Questions ?