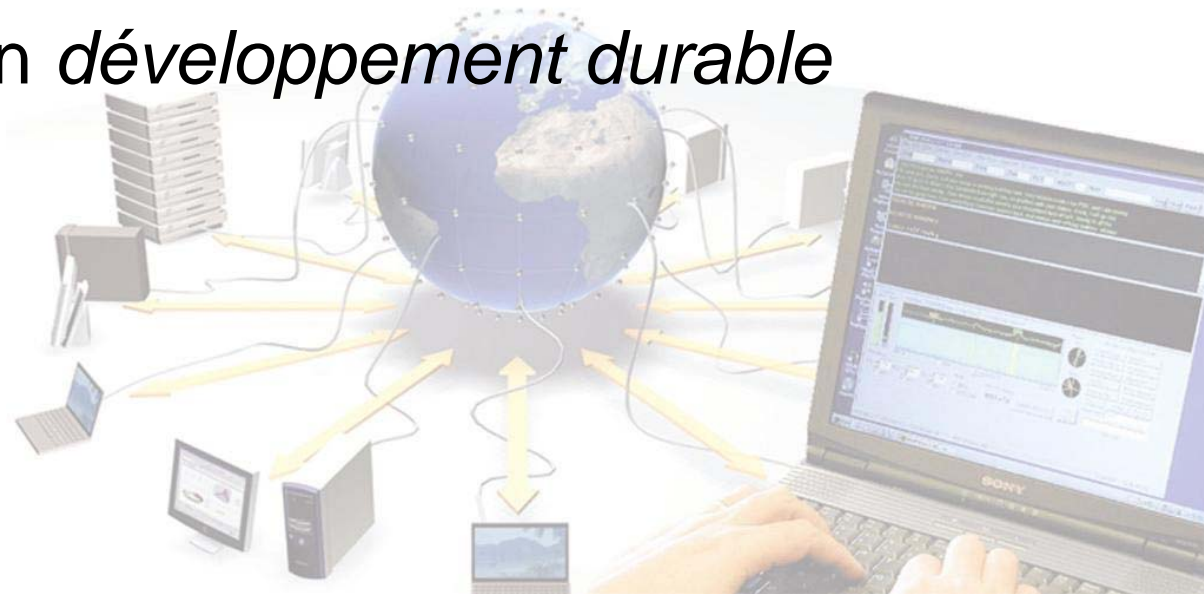


Introduction brève à la Grille LCG/EGEE

Charles Loomis

27 septembre 2005

Réunion *développement durable*



Agenda

- Sécurité
 - Authentification
 - Autorisation
- Architecture de la Grille
- Descriptions des tâches (JDL)
- Tâche triviale
- Tâche qui requies des donnés
- Tâche MPI
- Documentations

- Cette présentation concerne la version du logiciel LCG 2.6.0. Les commandes pour gLite (le prochain système) sont différentes mais avec des fonctionnalités similaires.

Sécurité

- L'utilisation de la grille est sécurisée à l'aide de PKI (public key infrastructure).
 - PKI utilise une clé publique et une clé privée.
 - Les clés doivent être signées par une autorité certificat (CA) approuvée.
 - Le "designated name" (DN) est équivalent à nom d'utilisateur pour la grille.
- **Authentification:**
 - Les clés sont comme une pièce d'identité.

Autorité Certificat GRID-FR

- En France on utilise la CA GRID-FR.
 - <http://igc.services.cnrs.fr/GRID-FR/>
 - **Les certificats CNRS-Standard ne marchent pas pour la grille!**
 - La demande est vérifiée par le responsable de cette CA. Normalement l'attente est de 1 jour environ.
- Votre certificat :
 - Le certificat signé est délivré par browser web.
 - On doit extraire ce certificat pour l'utiliser dans la grille.
 - Voir informations pour "sauvegarder" ou "backup" le certificat pour votre browser web. (Voir lien "Documentation/Aide".)
 - La sauvegarde crée un fichier dans la format PKCS12 avec une extension "pfx" or "p12".

Préparation du Certificat

- Mettez le fichier “pfx” ou “p12” dans le répertoire ~/.globus
- On doit changer la format de PKCS12 à PEM.
 - openssl pkcs12 -nocerts -in **cert.p12** -out userkey.pem
 - openssl pkcs12 -clcerts -nokeys -in **cert.p12** -out usercert.pem
- Changez les permissions pour userkey.pem :
 - chmod 0600 userkey.pem
 - Sinon, s'affiche un diagnostic d'erreur “bad passphrase”.
- Testez la création d'un “proxy”.
 - grid-proxy-init
 - C'est OK si vous avez “Your proxy is valid until: ...”.

Organisations Virtuelles (VO)

- Pour utiliser la grille, vous devez être membre d'une organisation virtuelle (VO).
 - Equivalent d'une expérience pour HEP.
 - Chaque VO a une manager qui contrôle la liste des membres.
 - Pour enregistrer dans une VO, faire une demande avec votre browser web et certificat au serveur VOMS pour cette VO.
- **Autorisation :**
 - C'est comme une visa ou un permis de séjour.
 - L'accès aux ressources est contrôlé par les VO.
- **Pour le moment, il n'est possible d'être membre que d'une seule VO à la fois!**

La VO “vo.lal.in2p3.fr”

- Il y a un serveur VOMS pour la VO “vo.lal.in2p3.fr”.
 - <https://grid12.lal.in2p3.fr:8443/voms/vo.lal.in2p3.fr/>
 - Ouverte à tous les gens au LAL.
 - Les membres de cette VO sont autorisés à utiliser les ressources grilles au LAL.

Création d'un Proxy

- Un proxy est un nouveau certificat signé avec votre certificat normal mais avec une durée de vie beaucoup plus courte (12-24 heures).
- Une copie de votre proxy est envoyée avec chaque tâche et permet aux tâches d'utiliser les services grilles comme vous.
- Les commandes pour gérer votre "proxy" sont :
 - `grid-proxy-init`
 - `grid-proxy-info`
 - `grid-proxy-destroy`
- La commande "destroy" supprime uniquement la copie de proxy locale. Elle n'affecte pas les autres copies qui existent ailleurs et qui sont utilisées par les tâches.

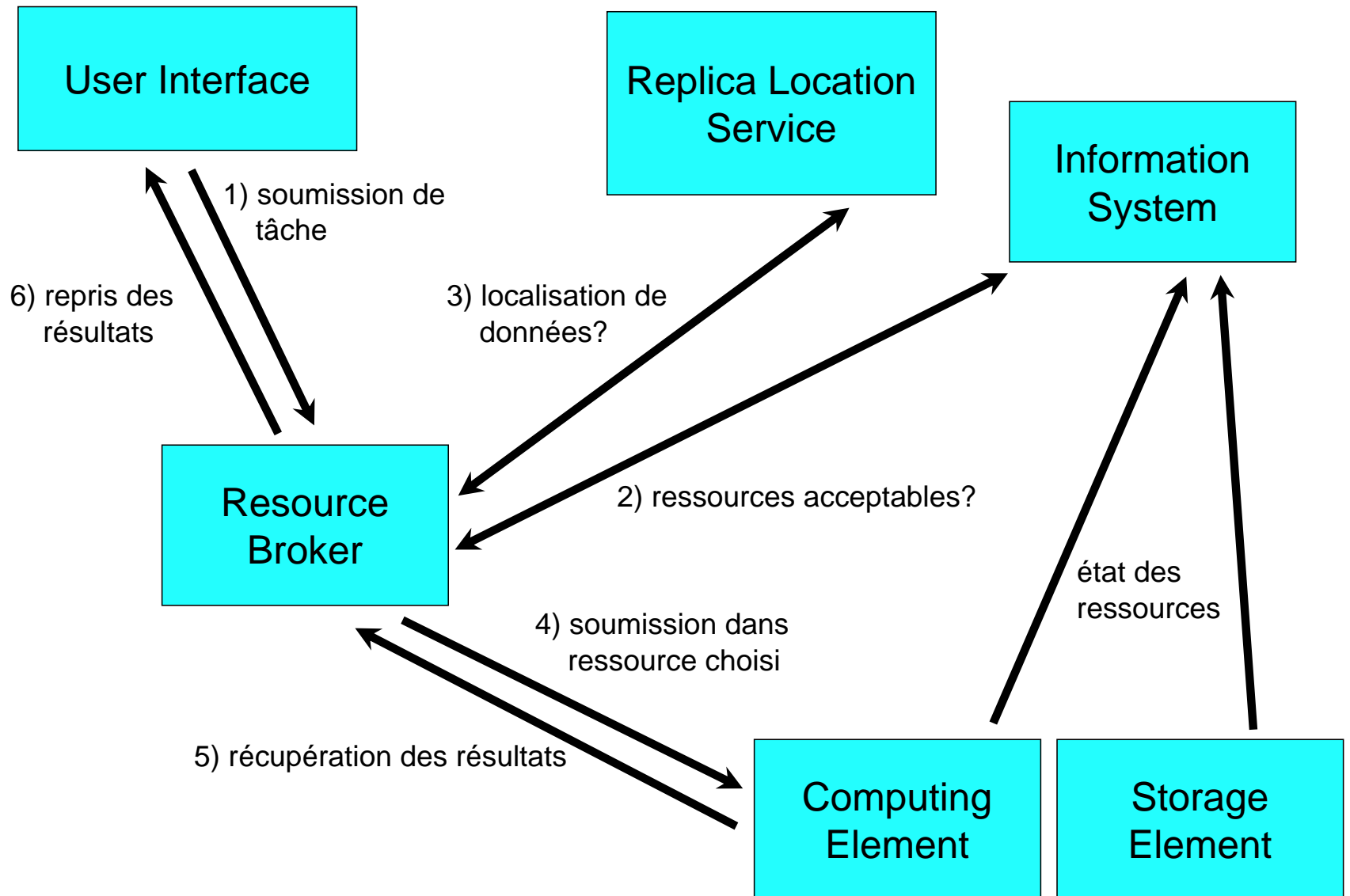
Autre Commandes Proxy

- Avec VOMS c'est possible d'être un membre des plusieurs VO en même temps. Cette fonctionnalité n'est pas encore bien assurée par les services grilles.
 - `voms-proxy-init`
 - `voms-proxy-info`
 - `voms-proxy-destroy`
- Pour les tâches longues il est nécessaire de renouveler les proxies automatiquement. Le serveur "MyProxy" permet ce renouvellement.
 - `myproxy-init`
 - `myproxy-info`
 - `myproxy-destroy`

Architecture

- La grille LCG/EGEE est une collection des services qui travaillent ensemble pour fournir une centre de calcul virtuel.
- Sécurité :
 - Certificate Authorities (CA)
 - VO Membership Service (VOMS)
- Système des informations :
 - RGMA (Relational Grid Monitoring Architecture)
 - MDS (basé sur LDAP)
- Gestion des tâches :
 - Resource Broker (RB)
- Gestion des données :
 - Storage Element (SE)
 - Replica Location Service (RLS)

Traitement de Tâche



Commandes

- `edg-job-list-match --vo my_vo job.jdl`
 - Une liste de tous les ressources pour satisfaire les besoins et dans l'ordre de "rank".
- `edg-job-submit --vo my_vo job.jdl`
 - Soumission d'une tâche.
- `edg-job-status jobid`
 - Status bref d'une tâche.
- `edg-job-get-logging-info jobid`
 - Le status détaillé donne toute l'histoire d'une tâche.
- `edg-job-get-output jobid`
 - Récupération les résultats stockés dans le RB d'une tâche.

Description des Tâches

- La description d'une tâche et de ses besoins est insérée dans un fichier avec le format JDL (Job Description Language).
 - Basé sur "ClassAds" de Condor.
 - Essentiellement paires de clé/valeur avec expressions.

```
Executable = "hsscript.sh";
```

```
Arguments = "arg1 arg2";
```

```
StdOutput = "std.out";
```

```
StdError = "std.err";
```

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

```
OutputSandbox = {"std.out", "std.err"};
```

```
Requirements = (other.GlueCEPolicyMaxCPUtime > 10) &&  
                (other.GlueCEPolicyMaxWallClockTime > 10);
```

```
Rank = other.GlueCEFreeCPUs;
```

Clés Typiques

- GlueCEInfoTotalCPUs
- GlueCEStateFreeCPUs
- GlueCEStateTotalJobs
- GlueCEStateRunningJobs
- GlueCEStateWaitingJobs
- GlueCEStateEstimatedResponseTime
- GlueCEPolicyMaxCPUTime
- GlueCEPolicyMaxWallClockTime
- GlueHostApplicationSoftwareRunTimeEnvironment

Tâche Triviale

- Utilise la commande “echo” pour afficher le message “HelloWorld”.

```
Executable      = "/bin/echo";  
Arguments       = "Hello World";  
StdOutput       = "std.out";  
StdError        = "std.err";  
OutputSandbox  = {"std.out", "std.err"};  
Requirements    = (other.GlueCEStateFreeCPUs > 0);  
Rank            = other.GlueCEStateFreeCPUs;
```

Ressources Disponibles

```
$ edg-job-list-match --vo atlas HelloWorld.jdl
```

```
Selected Virtual Organisation name (from --vo option): atlas  
Connecting to host grid09.lal.in2p3.fr, port 7772
```

```
*****
```

```
COMPUTING ELEMENT IDs LIST
```

```
The following CE(s) matching your job requirements have been found:
```

```
*CEId*
```

```
lxgate15.cern.ch:2119/jobmanager-lcglsf-grid_atlas
```

```
fal-pygrid-18.lancs.ac.uk:2119/jobmanager-lcgpbs-atlas
```

```
...
```

```
ce001.grid.bas.bg:2119/jobmanager-lcgpbs-atlas
```

```
lcg06.sinp.msu.ru:2119/jobmanager-lcgpbs-atlas
```

```
*****
```


Soumission

```
$ edg-job-submit --vo atlas HelloWorld.jdl
```

```
Selected Virtual Organisation name (from --vo option): atlas  
Connecting to host grid09.lal.in2p3.fr, port 7772  
Logging to host grid09.lal.in2p3.fr, port 9002
```

```
*****  
*****
```

JOB SUBMIT OUTCOME

```
The job has been successfully submitted to the Network Server.  
Use edg-job-status command to check job current status. Your job identifier  
(edg_jobId) is:
```

```
- https://grid09.lal.in2p3.fr:9000/7EDeL\_gfjbcpgmolP3U7IA
```

```
*****  
*****
```

Statut

```
$ edg-job-status https://grid09.lal.in2p3.fr:9000/7EDeL\_gfjbcpgmolP3U7IA
```

```
*****
```

```
BOOKKEEPING INFORMATION:
```

```
Status info for the Job :
```

```
https://grid09.lal.in2p3.fr:9000/7EDeL\_gfjbcpgmolP3U7IA
```

```
Current Status:      Scheduled
```

```
Status Reason:      Job successfully submitted to Globus
```

```
Destination:        fal-pygrid-18.lancs.ac.uk:2119/jobmanager-lcgpbs-atlas
```

```
reached on:         Sun Sep 25 11:54:31 2005
```

```
*****
```

Résultats

```
$ edg-job-get-output https://grid09.lal.in2p3.fr:9000/7EDeL\_gfjbcpgmolP3U7IA
```

```
Retrieving files from host: grid09.lal.in2p3.fr ( for  
https://grid09.lal.in2p3.fr:9000/7EDeL\_gfjbcpgmolP3U7IA )
```

```
*****
```

```
*
```

```
JOB GET OUTPUT OUTCOME
```

```
Output sandbox files for the job:
```

```
- https://grid09.lal.in2p3.fr:9000/7EDeL\_gfjbcpgmolP3U7IA
```

```
have been successfully retrieved and stored in the directory:  
/home/loomis/JobOutput/loomis_7EDeL_gfjbcpgmolP3U7IA
```

```
*****
```

```
*
```

Cette commande marche uniquement quand le status de tâche est “Done (success)”!

On doit créer le répertoire ~/JobOutput une fois.

Résultats

```
$ cd /home/loomis/JobOutput/loomis_7EDeL_gfjbcpgmolP3U7IA
```

```
$ ls -l
```

```
total 4
```

```
-rw-rw-r--  1 loomis  loomis           0 Sep 25 14:01 std.err  
-rw-rw-r--  1 loomis  loomis          12 Sep 25 14:01 std.out
```

```
$ cat std.out
```

```
Hello World
```

HelloWorld MPI

- On doit ajouter le JobType dans le fichier JDL.
- Il est préférable de compiler l'exécutable sur place.
- Il y a des différences entre sites :
 - Le "node file" est différent pour PBS et LSF.
 - Quelques sites a un système de fichiers partagé et quelques sites n'a pas.

```
JobType = "MPICH";  
NodeNumber = 16;  
Executable = "MPIHelloWorld.sh";  
Arguments = "MPIHelloWorld";  
StdOutput = "std.out";  
StdError = "std.err";  
InputSandbox = {"MPIHelloWorld.sh", "MPIHelloWorld.c"};  
OutputSandbox = {"std.err", "std.out"};  
Requirements = (other.GlueCEStateFreeCPUs > NodeNumber);  
Rank = other.GlueCEStateFreeCPUs;  
#Requirements = RegExp("grid.*lal.in2p3.fr.*", other.GlueCEUniqueID);
```

HelloWorld MPI (main)

```
/* Simple "Hello World" program in MPI. */

#include "mpi.h"
#include <stdio.h>

int main(int argc, char *argv[])
{
    int numprocs; /* Number of processors */
    int procnum; /* Processor number */
    /* Initialize MPI */
    MPI_Init(&argc, &argv);

    /* Find this processor number */
    MPI_Comm_rank(MPI_COMM_WORLD, &procnum);

    /* Find the number of processors */
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    printf ("Hello world! from processor %d out of %d\n",
           procnum, numprocs);

    /* Shut down MPI */
    MPI_Finalize();
    return 0;
}
```

HelloWorld MPI (script)

```
#!/bin/sh
EXE=$1

echo "Compiling binary: $EXE"
mpicc -o ${EXE} ${EXE}.c

if [ "x$PBS_NODEFILE" != "x" ] ; then
    HOST_NODEFILE=$PBS_NODEFILE
fi
if [ "x$LSB_HOSTS" != "x" ] ; then
    HOST_NODEFILE=`pwd`/lsf_nodefile.$$
    for host in ${LSB_HOSTS}; do
        echo $host >> ${HOST_NODEFILE}
    done
fi
if [ "x$HOST_NODEFILE" = "x" ]; then
    echo "No hosts file defined. Exiting..."
    exit
fi

CPU_NEEDED=`cat $HOST_NODEFILE | wc -l`
echo "Node count: $CPU_NEEDED"

echo "Executing $EXE with mpirun"
chmod 755 $EXE
mpirun -np $CPU_NEEDED -machinefile $HOST_NODEFILE `pwd`/$EXE
```

HelloWorld MPI (résultats)

- Utilisez les commandes `edg-job-submit`, `edg-job-status`, `edg-job-get-output` normalement.

```
$ cat std.out
```

```
Compiling binary: MPIHelloWorld
```

```
Node count:      16
```

```
Executing MPIHelloWorld with mpirun
```

```
Hello world! from processor 15 out of 16
```

```
Hello world! from processor 7 out of 16
```

```
Hello world! from processor 3 out of 16
```

```
Hello world! from processor 11 out of 16
```

```
Hello world! from processor 6 out of 16
```

```
Hello world! from processor 2 out of 16
```

```
Hello world! from processor 14 out of 16
```

```
Hello world! from processor 10 out of 16
```

```
Hello world! from processor 1 out of 16
```

```
Hello world! from processor 9 out of 16
```

```
Hello world! from processor 13 out of 16
```

```
Hello world! from processor 5 out of 16
```

```
Hello world! from processor 8 out of 16
```

```
Hello world! from processor 12 out of 16
```

```
Hello world! from processor 4 out of 16
```

```
Hello world! from processor 0 out of 16
```


Gestion des Données

- Un exemple simple pour expliquer comment on gère des données dans la grille.
 - Créez un fichier texte.
 - Copiez ce fichier dans la grille.
 - Lancez une tâche qui utilise ce fichier.
 - Supprimez le fichier.
- Points importants :
 - Le quantum des données dans la grille est un fichier.
 - Les fichiers dans la grille sont invariants.
 - Chaque fichier a un identifiant unique (GUID).
 - Un fichier peut avoir un nom logique (LFN).
 - On peut avoir plusieurs copies du même fichier (réplicats).

Enregistrer un Fichier

```
$ perl -e 'print rand() . "\n"' > text.orig
```

```
$ cat text.orig
```

```
0.0312232657606764
```

```
$ LFN=lfn:loomis_20050927
```

```
$ lcg-cr --vo atlas -l $LFN file:`pwd`/text.orig
```

```
guid:52a624e0-561a-4a33-b4c5-a6a6534e7514
```

```
$ lcg-lr --vo atlas $LFN
```

```
sfn://grid11.lal.in2p3.fr/var/storage/LCG/atlas/generated/2005-09-  
25/file6263887d-c785-4182-a67f-4a205aeef1e3
```

JDL avec InputData

- On doit utiliser les deux paramètres InputData et DataAccessProtocol.
- Input data peut être spécifié par LFN ou GUID.
- DataAccessProtocol doit être “gsiftp” ou “rfio”.

```
Executable      = "HelloData.sh";  
Arguments       = "lfn:loomis_20050927";  
StdOutput       = "std.out";  
StdError        = "std.err";  
InputSandbox    = {"HelloData.sh"};  
OutputSandbox   = {"std.out", "std.err"};  
  
InputData       = {"lfn:loomis_20050927"};  
DataAccessProtocol = {"gsiftp"};
```

Script d'Analyse

```
$ cat HelloData.sh
#!/bin/sh

FNAME=text.copy
FURL=file:`pwd`/$FNAME
LFN=$1

lcg-cp --vo atlas $LFN $FURL

cat $FNAME
```

Résultats

- Avec la spécification des fichiers requis, le broker peut trouver la bonne ressource.

```
$ edg-job-list-match --vo atlas HelloData.jdl
```

```
Selected Virtual Organisation name (from --vo option): atlas  
Connecting to host grid09.lal.in2p3.fr, port 7772
```

```
*****
```

```
COMPUTING ELEMENT IDs LIST
```

```
The following CE(s) matching your job requirements have been found:
```

```
*CEId*
```

```
grid10.lal.in2p3.fr:2119/jobmanager-pbs-atlas
```

```
*****
```

```
$ # edg-job-submit, edg-job-status, edg-job-get-output
```

```
$ cd /home/loomis/JobOutput/loomis_EhZnU9iRHwtGLTcITXIAQQ
```

```
$ ls
```

```
std.err  std.out
```

```
$ cat std.out
```

```
0.0312232657606764
```

Résumé

- Si vous avez un certificat et êtes enregistré dans une VO, vous pouvez accéder à la grille LCG/EGEE.
- On a expliqué comment :
 - soumettre une tâche simple
 - utiliser MPI dans la grille
 - gérer un fichier dans la grille et y accéder
- Plus d'informations :
 - <http://cern.ch/lcg>
 - <http://www.eu-egee.org>
 - <http://glite.org> (futur middleware)
 - <http://lcg.web.cern.ch/LCG/users/support.html>