



# Gestion des Données

M. Jouvin (LAL-Orsay)

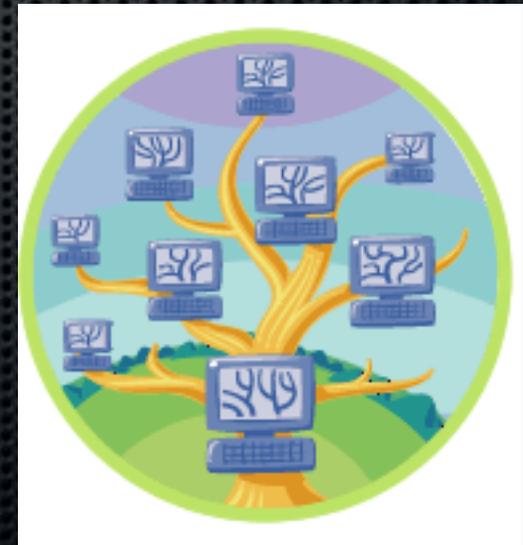
Tutorial Utilisateur Grille

MRM Grille Paris Sud, LAL

2 Juin 2010

# Agenda

- Les principaux composants
- Les commandes de base
  - Mettre des données sur la grille
  - Retrouver ces données
  - Trouver l'information
  - Dupliquer les données
  - Effacer les données
- La gestion des données au cours d'un job
  - Données sur votre bureau
  - Données sur la grille
- Le catalogue LFC



# Les Présupposés

- Les données peuvent se trouver sur un site distant : impossible de garantir un accès type POSIX (via NFS par exemple)
- Besoin de supporter plusieurs protocoles d'accès et d'un nommage indépendant du protocole d'accès
  - De nouveaux protocoles doivent pouvoir être ajoutées en fonction des évolutions technologiques
  - Nécessité d'API permettant la gestion transparente des différents protocoles
- Permettre d'avoir plusieurs replica d'un même fichier
  - Redondance / backup
  - Performance : avoir des copies « proches » de plusieurs sites (CE)



# Les Principaux Composants

- Gestion de l'espace de noms logiques : LFC
  - Garde la trace de la localisation physique des diverses copies (replica) d'un fichier sur la grille
  - Chaque fichier à 1 GUID (Global Unique IDentifier)
  - Chaque replica est désigné par un SURL (Site URL)
  - Arborescence et des commandes façon « système de fichiers »
- Gestion de l'espace de noms physiques : SRM
  - Utilise le GUID (ou le SURL) et fournit le TURL (Transport URL)
  - C'est l'interface avec la DB et les systèmes de stockage
    - ✓ Disques, bandes, MSS, ...
- Les protocoles de transfert de données : rfiio, gsiftp, file...
  - Utilisent le TURL



# Prérequis

- Toutes les commandes liées à la gestion de données supposent d'avoir un proxy valide
  - Détermination de la VO utilisée par l'utilisateur
  - Détermination des droits d'accès



# Transférer sur la Grille...

- Copier le *fichier /home/myhome/toto* (ordinateur local) sur le Storage Element *leSEchoisi.in2p3.fr* et l'enregistrer avec le nom de fichier logique *macopie.test* :
  - `lcg-cr -d leSEchoisi.in2p3.fr \`
  - `-I lfn:/grid/vo.u-psud.fr/myname/macopie.test \`  
`/home/myhome/toto`
- La commande ci-dessus retourne le "GUID" du fichier logique :
  - `guid:76373236-b4c7-11d8-bb5e-eba42b5000d0`
  - Les GUID sont permanents, les LFN non !
- Storage Element – machine accessible par la grille pour le stockage de données



# ... Transférer sur la Grille

- Logical File Name – nom d'un fichier sur la grille indépendant de sa localisation physique
  - Les LFNs peuvent comporter des dossiers et sous-dossiers et l'utilisateur est libre de les organiser à sa guise
  - Les dossiers non existants sont créés
- Il y a plein d'autres options disponibles: 'lcg-cr –help'



# Récupérer les Données

- Localisez les données en utilisant le LFN ou le GUID :
  - `lcg-lr lfn:/grid/vo.u-psud.fr/macopie.test`
  - `lcg-lr guid:76373236-b4c7-11d8-bb5e-eba42b5000d0`
- Ces commandes retournent le/les SURL : nom des réplicas
  - `sfn://leSEchoisi.in2p3.fr/grid/vo.u-psud.fr/tutorial/data/generated/2004-11-10/file7115df45-b4c7-11d8-bb5e-eba42b5000d0`
  - 1 seul replica par SE
- Obtenir une copie locale des données sur la grille : `lcg-cp`
  - `lcg-cp guid:76373236-b4c7-11d8-bb5e-eba42b5000d0 newcopy`
  - `lcg-cp lfn:/grid/vo.u-psud.fr/macopie.test newcopy`
  - `lcg-cp guid:76373236-b4c7-11d8-bb5e-eba42b5000d0 newcopy`



# Découvrir les Ressources

➤ Comment trouver les SEs disponibles pour une VO?

- lcg-infosites --vo atlas se

Avail Space(Kb)	Used Space(Kb)	Type	SEs
845990	1	n.a	sa3-se.egee.cesga.es
102800000	n.a	n.a	se201.grid.ucy.ac.cy
n.a	22095	n.a	ctb04.gridctb.uoa.gr
6200000	5642131	n.a	lxb1921.cern.ch
...			

➤ Beaucoup d'autres informations peuvent être affichées

- Serveurs LFC de la VO : utiliser l'argument lfc (au lieu de se)
- Attention : l'option '--vo' est indispensable
  - ✓ Un SE peut être valide pour une VO et pas pour une autre
- 'man lcg-infosites' pour toutes les options possibles



# Le SE par défaut

- Sur 1 UI ou 1 WN, si votre VO est 'vo.u-psud.fr', le SE par défaut est désigné par la variable:
  - ✓ \$VO\_VO\_U\_PSUD\_FR\_DEFAULT\_SE
  - ✓ C'est le SE sur lequel on écrit par défaut quand aucun n'est spécifié
- Pour utiliser le SE par défaut, il suffit d'omettre l'option -d (lcg-cr et lcg-rep)
  - ✓ Par ex., dans un job qui doit pouvoir s'exécuter sur plusieurs sites en utilisant le SE "local"
- Après écriture du fichier, il est possible de retrouver le SE utilisé avec lcg-lr
  - Rarement nécessaire car on peut utiliser le GIUD ou LFN
  - Essentiellement si plusieurs replicas



# Dupliquer les données

- Dupliquer des données vers un autre SE à partir du LFN (ou GUID ou SURL) du fichier
  - lcg-rep \
    - lfn:/grid/vo.u-psud.fr/myname/macopie.test \
      - -d autreSE.cern.ch
- lcg-lr liste la l'ensemble des replicas :
  - srm://leSEchoisi.in2p3.fr/grid/vo.u-psud.fr/tutorial/data/generated/2004-11-10/file7115df45-b4c7-11d8-bb5e-eba42b5000d0
  - srm://autreSE.cern.ch/pnfs/cern.ch/data/vo.u-psud.fr/generated/2005-11-28/file46034814-3021-47e0-a382-60cde5efbdf0



# Effacer les données

- Pour effacer un replica sur un SE :
  - `lcg-del -s lautreSE.cern.ch \`
  - `lfn:/grid/vo.u-psud.fr/myname/macopie.test`
- Il faut spécifier le SE car s'il y a plusieurs replicas, la commande ne sait pas quel fichier effacer.
- Pour effacer tous les replicas d'un même fichier, il faut rajouter l'option `-a`
  - Dans ce cas le SE doit être omis
- Le catalogue LFC est mis à jour automatiquement



# Job : Accès aux Données

- Disposer des données produites dans le job (output)
  - Récupérer simplement les données sur votre machine de bureau
  - Rendre ces données disponibles/accessibles sur la grille
- Lire des données locales dans le job (input)
  - Spécifier des données au moment de la soumission du job
  - Disposer vos données sur la grille avant un job
  - Accéder à vos données grille depuis l'intérieur du job
- Déplacer des données d'un point à un autre sur la grille



# Job : Récupérer les Données

- Le job génère des données dans le répertoire de travail du WN
  - A la fin du job, les fichiers nouveaux sont placés dans un espace de stockage temporaire sur le WMS appelé « output sandbox »
  - Récupération via “glite-job-get-output”
- Le JDL doit contenir la liste des fichiers à mettre dans l’output sandbox
  - ✓ `OutputSandbox = {"higgs.root", "graviton.HDF"};`
  - Les noms doivent correspondre à des fichiers produits par le job
  - Méthode fortement déconseillée pour de gros fichiers (> 10MB)  
→ espace de stockage temporaire limité



# Job : Stocker sur la Grille

- Il n'existe aucun mécanisme automatique pour stocker les données produites par le job sur la grille
  - L'ancienne clause JDL « OutputData » est obsolète
- Le job (script) doit contenir les commandes lcg-cr nécessaires pour stocker les fichiers à la fin du job.



# Les Inputs

- Pour les données « locales » : attribut InputSandbox du JDL
  - ✓ `InputSandbox = {"input-ntuple.root", "job.sh"};`
  - Attention : on ne peut pas utiliser ce champ pour de trop gros fichiers (> 1 MB)
- Pour les données « grille » :
  - Spécifiez les données grille à l'aide du champ InputData pour influencer la sélection du site:
    - ✓ `InputData = {"lfn:/grid/vo.u-psud.fr/myfile.dat", "guid:f62344d9-ca25-458a-adf7-9d8150031dcd"};`
  - L'accès aux données est de la responsabilité du job : copie locale sur le WN avec `lcg-cp` ou accès direct avec `rfio/xroot`
  - La commande `glite-brokerinfo` permet d'obtenir la liste des fichiers indiquée dans l'InputData



# LCG File Catalog (LFC)

- Le catalogue LFC (LCG File Catalog) présente une arborescence et des commandes façon Posix :
  - il est possible de lister les références, de créer une nouvelle arborescence, de faire des liens symboliques...
- Par défaut, le catalogue utilisé est le catalogue central de la VO.
  - ✓ `lcg-infosites -vo vo.u-psud.fr lfc`  
retourne le nom du LFC central
  - ✓ `lcg-infosites -vo vo.u-psud.fr lfcLocal`  
retourne la liste des catalogues locaux de la VO
- Pour utiliser un catalogue local :
  - ✓ `export LFC_HOST=<hostname du LFC choisi>`
  - ✓ Eviter de définir inutilement `LFC_HOST`



# Commandes LFC...

- Elles permettent la gestion des noms logiques (LFN)
  - Aucun impact sur les données elle-mêmes (replicas)
  - Rarement utilisées directement : plutôt via les `lfc_utils` (`lfc-xxx`)
- Racine de l'arborescence d'un LFC :
  - ✓ `/grid/<nom de la VO>`
  - on peut stocker cette racine dans la variable `LFC_HOME`
    - ✓ `export LFC_HOME=/grid/vo.u-psud.fr`
  - Si on définit cette variable, l'option `'-l lfn:mylfn'` est en fait interprétée comme `'-l lfn:$LFC_HOME/mylfn'`
- Lister les références :
  - ✓ `lfc-ls /grid/vo.u-psud.fr/...`
  - ✓ `lfc-ls $LFC_HOME/...`



# ... Commandes LFC

- Créer un nouveau répertoire :
  - ✓ `lfc-mkdir /grid/vo.u-psud.fr/.../<mon rep>`
- Autres ex. de commandes :
  - ✓ `lfc-ln`, `lfc-rm`, `lfc-rename`, `lfc-getacl`, `lfc-setacl`
  - ✓ `lfc-chmod`, `lfc-chown`, `lfc-setcomment`, `lfc-delcomment`
- Aide en ligne : commande '`man lfc-xx`' ou simplement '`lfc-xx`'



# GFAL

- La copie sur le WN des fichiers n'est pas toujours souhaitable, en particulier si on veut lire/écrire de gros fichiers
  - Surtout s'ils sont plus gros que l'espace disponible sur le WN
- Pour accéder directement à des fichiers distants : GFAL
  - GFAL : 'Grid File Access Library'
    - ✓ Cette librairie offre une API Posix pour C/C++, Python, Perl
    - ✓ Accès aux fichiers disponibles sur un SE depuis une application
  - Les appels possibles sont (liste non exhaustive) :
    - ✓ `gfal_open`, `gfal_read`, `gfal_write`, `gfal_close`, ...
  - La syntaxe des noms de fichiers est la syntaxe `SURL`
    - ✓ Utilise principalement les protocoles `rfile` ou `dcap/gsidcap`
  - Documentation + exemples : `'man gfal'`, `'man gfal_open'`, ...
- Pour les écritures, préférer l'écriture locale + transfert



# Commandes lcg\_utils...

- Ensemble de commandes fournissant des fonctionnalités complètes (interaction LFC + SE/SRM)
  - Pour l'aide en ligne : 'man lcg-xx ' ou 'lcg-xx' (sans argument)
- Commandes de transferts
  - lcg-cp copie simple, y compris from/to local
  - lcg-cr copy and register (in LFC)
    - ✓ Destination nécessairement sur la grille
  - lcg-del delete file
  - lcg-rep replicate file, y compris mise à jour LFC



# ... Commandes lcg\_utils

- Informations sur un fichier ou replica
  - lcg-gt      get TURL
  - lcg-lr      list replicas
  - lcg-la      list alias
  - lcg-lg      list GUID
  
- Opérations spécifiques : à ne pas utiliser normalement...
  - lcg-aa      add alias
  - lcg-ra      remove alias
  - lcg-sd      set FileStatus to done (marquer le transfert terminé)
  - lcg-rf      register file in LFC
  - lcg-uf      unregister file from LFC



# Commande glite-brokerinfo

- Récupérer des informations sur un job soumis via un WMS
  - Ensemble des options possibles : `glite-brokerinfo -help`
- Principales options
  - `glite-brokerinfo getCE`
  - `glite-brokerinfo getDataAccessProtocol`
  - `glite-brokerinfo getInputData`
  - `glite-brokerinfo getSEs`
  - `glite-brokerinfo getCloseSEs`
  - `glite-brokerinfo getSEMOUNTPoint <SE>`
  - `glite-brokerinfo getSEFreeSpace <SE>`
  - `glite-brokerinfo getLFN2SFN <LFN>`
  - `glite-brokerinfo getSEProtocols <SE>`
  - `glite-brokerinfo getSEPort <SE> <Protocol>`

