

Introduction à CMSSW_1_0_1

Jérémy Andrea

- CMSSW sur ui2:
 - Les fichiers de configurations .cfg
 - Génération
 - Simulation et reconstruction
 - Les outputs
 - Exemple d'analyse
 - EDAnalyser
 - Un rootTuple simple
- CMSSW sur la grille de Strasbourg
 - Les fichiers job.jdl
 - Les fichiers job.sh
 - Mon premier job!

CMSSW sur ui2

- « Setup » de l'environnement:
 - Se logger sur ui2:
 - cd `/opt/sbg/cms/ui2-data1/`
 - mkdir myname
 - cd myname
 - export `VO_CMS_SW_DIR=/opt/exp_soft/cms`
 - source `$VO_CMS_SW_DIR/cmsset_default.sh`
 - `scramv1 project CMSSW CMSSW_1_0_1`
 - cd `CMSSW_1_0_1/src`
 - `eval `scramv1 runtime -sh``
 - Pas de `service cvs` disponible pour l'instant
- Lancer le job : nécessite un fichier de configuration `.cfg`.
- Exemple de production/analyse:
 - `cp ../../../../CMSSW_test/*.tgz .`
 - `tar -xvzf ExampleAnalysis.tgz`

Le fichier de configuration

- Contient pour la production d'événements:
 - **Succession de modules:**
 - Configuration de Pythia pour la production, nombre d'événements etc...
 - Fichiers de sortie
 - Différents « includes » contenant les algorithmes de simulation, digitization et reconstruction,
 - Vos propres algorithmes
 - **Ordre d'exécution des modules: path**
- Fichiers de configuration pour la production:
- http://cmsdoc.cern.ch/swdev/viewcvs/viewcvs.cgi/CMSSW/Configuration/ReleaseValidation/data/?cvsroot=CMSSW&only_with_tag=BTauTag100
- Un exemples :
 - **data/ttbar_genrec_allReco.cfg** (trop « gourmand » en mémoire vive, ne fonctionne au cern que pour 1 à 2 événements)
 - Exécution : `cmsRun ttbar_genrec_allReco.cfg`

Composition du fichier de sortie

- à ajouter dans le module « `PoolOutputModule` » ,
- Nom des objets = 4 parties séparés par des underscores: `A_B_C_D`

```
untracked vstring outputCommands = {  
    "keep *", #enregistre tous les objets,  
    "drop *_muoncscdigi_*_* #n'enregistre pas tous les objets avec  
B=muoncscdigi  
}
```
- Listes des `objets dans un fichiers` = exécuter `exampleDump.cfg`
- Exemple de fichier root.

Exemple d'analyse

- Exemple EDAnalyser:
 - mkdir Demo
 - cd [Demo](#)
 - makedanlzzr ExampleAnalysis
 - Attention au [BuildFile](#): nécessite des modifications dans le cas des particules générées.
 - Ici tout est près dans ExampleAnalysis
- Construit une arborescence permettant le développement de nouveaux modules.
- Classes d'objets reconstruits et simulés:
 - http://cmsdoc.cern.ch/Releases/CMSSW/latest_nightly/doc/html/
- Exemple dans [exampleAnalysis.cc](#)

Utiliser CMSSW

- Création de modules d'analyse avec **EDAnalyser**? (création de ntuple)
- Avec **EDProducer**? (modification du root tree de CMSSW pour l'analyse)
- Construction d'une **structure d'analyse commune**?
- Optimiser l'utilisation de CMSSW nécessite une bonne connaissance du software.

CMSSW sur la grille de Strasbourg:

Merci à Yannick!

- Après avoir récupérer le **certificat**...
 - Dans votre browser web récupérer **mykey.p12**
 - Pour **mozilla** : Edit->Preference->Privacy&Security->Certificats
->ManageCertificates->Backup-> mykey.p12
 - Création du **./globus**:
 - mkdir ~/.globus
 - openssl pkcs12 -in mykey.p12 -clcerts -nokeys -out
~/.globus/usercert-`date +%Y%m%d`.pem
 - openssl pkcs12 -in mykey.p12 -nocerts -out ~/.globus/userkey-`date
+%Y%m%d`.pem
 - chmod 400 ~/.globus/userkey*.pem
 - chmod 444 ~/.globus/usercert*.pem

CMSSW sur la grille de Strasbourg

- Se placer dans le répertoire data/
- Copier et décompresser [Grid_strasbg.tgz](#)
- [Soumission à la grille](#) (Strasbourg uniquement)
 - Job définit dans le fichier [job.jdl](#)
 - Nécessite un fichier script [job.sh](#)
- `voms-proxy-init -voms cms` (pour utiliser la grille)
 - `export VO_CMS_SW_DIR=/opt/exp_soft/cms`
 - `source $VO_CMS_SW_DIR/cmsset_default.sh`

CMSSW sur la grille de Strasbourg

- **Soumettre un job:**
 - `edg-job-submit job.jdl`
 - « `edg-job-submit --resource sbgat417.in2p3.fr:2119/jobmanager-lcgpbs-cms job.jdl` » (à Strasbourg uniquement)
 - Renvois le numéro d'ID du job.
 - « `edg-job-status jobID` » : status du job (-i pour un fichier contenant les jobID si plusieurs)
 - Une fois le status « Done »:
 - « `edg-job-get-output jobID` » : récupère les outputs dans un répertoire

CMSSW sur la grille de Strasbourg

- Le fichiers Root est **trop volumineux** pour être récupérer avec la **sendbox**
- Dans output/message.txt, chercher la ligne commençant par “ lfn:... ” en fin de fichier
- **Taper** :
 - lcg-cp -v --vo cms “lfn...” file://`pwd`/ttbar.root
 - lcg-del -v --vo cms -a “lfn”.
- Pas de montage NFS pour l’instant, mais prévu.

Quelques liens utiles

- <https://twiki.cern.ch/twiki/bin/view/CMS/September06CMSweekTutorials>
- <https://uimon.cern.ch/twiki/bin/view/CMS/ReleaseSchedule>
- <https://twiki.cern.ch/twiki/bin/view/CMS/WorkBook>
- <http://cmsdoc.cern.ch/swdev/viewcvs/viewcvs.cgi/CMSSW/?cvsroot=CMSSW#dirlist>
- <https://hypernews.cern.ch/HyperNews/CMS/login.pl?&url=%2fHyperNews%2fCMS%2ftop%2epl>
- <http://cmsdoc.cern.ch/swdev/viewcvs/viewcvs.cgi/?cvsroot=CMSSW>
- <http://cmslxr.fnal.gov/lxr/>