Document de développement du service courbe de lumière ASTEP

27 juin 2014

Partie 1 : installation

Ce service est actuellement installé sur la machine virtuelle astep-vo.oca.eu. Cette machine, à ce jour, n'est pas accessible depuis l'extérieur de l'OCA, ni en wi-fi sans VPN. L'ouverture vers l'extérieur est du ressort du SIT.

Parmi les éléments périphériques nécessaires au fonctionnement du service, on trouve :

-MySQL : le gestionnaire de base de donnée. On peut accéder en ligne de commande et entrer des requête à la main, si l'on connaît le langage SQL. Le nom de la base de donnée est ASTEP_v6, et il faut veiller (en cas de réinstallation par exemple) qu'elle ai un compte utilisateur (en plus de root) nommé 'reader' avec le mdp 'readerpwd' ayant les permissions de lecture (select), mais pas d'écriture. Ce compte est utilisé par le service pour consulter les données. La définition de la structure de la base se trouve dans le fichier astep_database_v6.sql (voir sur le trac du SVN)

-Tomcat : un outil de gestion de serveur. C'est lui qui fait tourner le code du service. On peut accéder à son manager via une page web : http://astep-vo.oca.eu/manager/html (login : 'tomcat', mpd : '23%s3cret') Tomcat installe les services dont il a la charge dans le répertoire /var/lib/tomcat6/webapps. Tomcat dispose également d'un répertoire ROOT (tout en majuscule) destiné a accueillir les page web autres que celles des services. C'est là que ce trouve la page de garde que l'on voit en tapant http://astep-vo.oca.eu

-Vizquery : un outil client du CDS qui permet d'interroger Vizier a distance. Il est actuellement installé dans /usr/local/bin/. Plus d'infos sur fonctionnement de cet outil ici : http://cdsweb.u-strasbg.fr/doc/vizquery.htx

-L'existence d'un dossier de travail 'temp' se trouvant dans le 'webapps' de tomcat, avec permission pour lire et écrire. C'est dans ce répertoire que sont créé ou copié tout les fichiers (votable, etc) qu'un utilisateur pourra télécharger par la suite.

Partie 2 : fonctionnement général

Le service est constitué de six classes dont deux servlets :

-RequestServlet : interprète les données entrées par l'utilisateur
-RetrieveServlet : répond à une demande récupération de fichiers courbes de lumières
-DatabaseManager : interagit exclusivement avec la base de données
-FileGenerator ; lit et écrit toutes sorte de fichiers
-RessourcesManager : exécute des commandes externes au service (e.g : Vizquery)
-PageGenerator : construit le code HTML des page affiché par le service

Les deux classes qui réagissent au contact de l'utilisateur sont RequestServlet et RetrieveServlet.

RequestServlet récupère les données du formulaire entrées par l'utilisateur. A ce stade, il n'y a pas beaucoup de contrôle du type de donnée (venant d'un formulaire, tout est 'String')

Cas numéro 1 : le servlet va réagir en priorité si le champ *'search by name'* est remplis : il va considérer que c'est une demande pour la fiche d'un objet individuel, peu importe s'il y a autre chose dans le formulaire. Il va tenter une reconnaissance de nom, en faisant appel à la méthode checkVizier() puis interroge la base avec searchByName(). Si cette dernière retourne quelque chose, il copie les fichiers courbes de lumières associées à l'objet concerné et les convertis en votable avec writeVOFile(). Il utilise ensuite detailPageSetup() pour créer la page résultat.

Cas numéro 2 : c'est une recherche 'normale' : le servlet passe les paramètres du formulaire au DatabaseManager via selectStars(), puis affiche le résultat correspondant avec fullPageSetup().

Cas numéro 3 : RequestServlet gère aussi la création d'une page de plot, s'il détecte la présence des paramètres *LcfileToPlot* et *period*, auquel cas il appelle plotPageSetup() pour note ; cette fonctionnalité aurait plus sa place logique dans RetrieveServlet.

RetrieveServlet n'intervient qu'après l'execution du cas numéro 2. Après que l'utilisateur aie coché les noms des courbes de lumière qui l'intéresse, ce servlet se charge de copier et convertir en votables les fichier correspondant, à l'aide des méthodes copyFileFromTaurus() et astepVOtable(). Il crée dans le répertoire 'temp' une votable générale appelée astep_votable.vot, contenant toutes les informations, et une table par objet avec leur courbes respectives (nom de l'objet .vot)

Parmi les autres composantes du code du service, on trouve des éléments Javascript :

-Aladin lite : http://aladin.u-strasbg.fr/AladinLite/
-Un code (assez rudimentaire) permettant de communiquer avec des noeuds SAMP
-Un code permettant de plotter un fichier de donnée avec la librairie Flot

Partie 3 : gestion de la BDD

Le fichier de définition

La base de donnée est composée de 6 tables :

-STARS
-COMPANION_OBJECT
-FIELDS
-USERS
-LIGHTCURVES
-PIPELINE_RESULTS

Ces tables sont définies dans le fichier astep_database_v6.sql (disponible sur le Trac) qui ajoute également quelques lignes à la table USERS (commande INSERT INTO). Ce fichier sera utilisé pour initialiser (ou reconstruire) la base de donnée (voir plus bas)

Manipulation dans l'environnement MySQL

En se connectant en ssh sur la machine astep-vo, on peut entrer dans la base de donnée mySQL et effectuer des commandes à la main. On y entre de la manière suivante :

mysql -u root -pmp3c072012 ASTEP_v6

Où root est évidemment l'administrateur, 'mp3c072012' le mot de passe (il doit être collé au '-p', et porte ce nom pour des raisons historiques) et ASTEP_v6 le nom de la base. On entre ensuite dans l'environnement MySQL, signalé par un 'mysql>' ; Voici quelques commandes importantes à connaître (les point-virgules sont significatifs):

- \q (pour quitter)
- \help
-show tables ; (affiches les noms des tables de la base)
-describe 'table'; (liste les champs de la table 'table')
-select * from USERS ; (liste les utilisateurs reconnu dans le champs 'restricted access' du service)
-insert into USERS values (x, 'username','pwd','mail@mail',c);
Crée un utilisateur qui pourra être reconnu par le service, avec un numéro 'x' qui doit être unique et un niveau d'accès 'c'
-select count(*) from 'table' ;
Donne le nombre de lignes d'une table
-select * from STARS where id_star='numero_astep' ; (ici, les quotes sont importantes)
Renvoie la ligne correspondant a ce numéro astep. On peut remplacer le '*' par des noms de colonnes séparée par une virgule (voir commande 'describe table')

Faire une sauvegarde de la base de donnée :

A faire par exemple avant une mise à jour. En dehors de l'environnement MySQL, entrer la commande suivante :

mysqldump -u root -pmp3c072012 ASTEP_v6 > bdd_astep_save_année_mois_jour.sql

Et pour restaurer :

mysql -u root -pmp3c072012 ASTEP_v6 < bdd_astep_save_année_mois_jour.sql

Mettre à jour les données avec AstepDataLoader :

AstepDataLoader est l'autre code disponible sur le SVN. S'il faut ajouter une nouvelle année : il suffit d'éditer la classe (mal nommée) TestReader et ajouter :

mp.yearProcessing("2012") ;

Dans le corps de sendProcesser(), puis recompiler.

Pour se servir du loader, il faut télécharger AstepDataLoader_6.jar sur astep-vo et executer la commande suivante :

java -jar AstepDataLoader_6.jar > data_batch.sql

Le loader va parcourir l'arborescence et créer un fichier batch contenant toutes les données formatées au format sql. Des warnings à propos de tags 'comments' vont s'afficher, on peut les ignorer, ils sont générés par la bibliothèque de lecture de fichiers .fits

Il est possible de contrôler le résultat en ouvrant le batch avec un éditeur de texte, ou en faisant grep A-045-0402-1878 data_batch.sql pour avoir un échantillon et vérifier que les commandes sql sont bien formatées.

Ensuite il faut ingérer ce batch, avec les commandes suivantes :

mysql -u root -pmp3c072012 ASTEP_v6 < astep_database_v6.sql

Cela va « flusher » les anciennes données et recréer la structure de la base. Puis :

mysql -u root -pmp3c072012 ASTEP_v6 < data_batch.sql

Le temps d'exécution total doit-être de quelques minutes.

Partie 4 : Modifier le service

L'opération la plus courante consiste à vouloir ajouter une colonne dans le tableau résultat.

Dans la classe DatabaseManager, il faut éditer la constante correspondant à la requête SQL que l'on souhaite modifier (COL_STAR, COL_STAR_BLS, ou COL_TRANSIT) et ajouter le nom de la colonne. Par exemple, pour ajouter la colonne période :

private static final String COL_STAR = "STARS.id_star, ra_j2000, dec_j2000, period, F_mag_UCAC, date_obs_start, date_obs_end, FIELDS.id_field, [...] as path"

Une nouvelle colonne ne doit jamais être ajoutée à la fin de la ligne : réservée au path du fichier courbe de lumière (signalé par concat(...)) ni au début, réservé à la clé primaire.

Il faut ensuite recompiler le code du service (qui doit générer un fichier archive nommé ASTEP_service.war)

Puis aller sur la page du Tomcat Manager (http://astep-vo.oca.eu/manager/html), désinstaller la version courante du service ('undeploy') et uploader la nouvelle version (WAR file to deploy) le reste de l'installation est automatique.