Rapport du projet BASH : Rapatriement automatique de ressources

LEDEE Maxime & LEVON Stéphanie

Décembre 2015

Table des matières

1 Structure du programme **2**
- 1.1 Les banques de données . . . 2
  - 1.1.1 Le répertoire Data/ . . . 2
  - 1.1.2 Le fichier Log/get_data.log . . . 2
  - 1.1.3 Le répertoire Reports/ . . . 2
- 1.2 Les scripts . . . 2
  - 1.2.1 main.sh . . . 2
  - 1.2.2 Bin/gestion_user.sh . . . 2
  - 1.2.3 Bin/gestion_repatriations.sh . . . 2
  - 1.2.4 Bin/gestion_strategies.sh . . . 3

2 Notes d'utilisation **3**
- 2.1 Les fonctions liées aux utilisateurs . . . 3
- 2.2 Les fonctions liées aux rapatriements . . . 3
- 2.3 Les fonctions liées aux stratégies . . . 3
- 2.4 Autre fonctionnalité . . . 3

3 Intéraction entre les scripts **3**

4 Génération d'un rapport **3**

5 Informations à propos du logiciel **4**

6 Modules requis **4**

7 Perspectives d'évolution **4**
- 7.1 Fonctionnalités liées à la génération du rapport . . . 4
- 7.2 Fonctionnalités liées au rapatriement d'une ressource . . . 4
- 7.3 Structure du programme . . . 5

1 Structure du programme

1.1 Les banques de données

1.1.1 Le répertoire Data/

Il contient quatre fichiers qui contiennent respectivement les données relatives aux utilisateurs du programme, les demandes de rapatriements de l'ensemble des utilisateurs ainsi que les stratégies de rapatriements (qui rapatrie quoi et où). Le dernier script contient les périodicités suivant la mise en forme cron.

1.1.2 Le fichier Log/get_data.log

Il contient les historiques de tentatives de rapatriement de données.

1.1.3 Le répertoire Reports/

Ce répertoire contient les rapports qui sont générés depuis le menu principal du programme. Il contient également un répertoire caché nommé ".model/Document" stockant des fichiers au format .tex et .sty .

1.2 Les scripts

1.2.1 main.sh

Au sein du répertoire du projet, un script main.sh a été créé. Ce fichier permet :

- d'afficher le Menu Principal par l'intermédiaire d'une fonction "displayMenuGenerationReport". Ce menu liste plusieurs fonctionnalités du programme :
  - Génération d'un rapport
  - Gestion de la liste des utilisateurs
  - Gestion de la liste des ressources à rapatrier
  - Gestion de la liste des stratégies
- de rappatrier une données à partir d'un identifiant d'une stratégie,
- de générer un rapport en fonction des critères de recherche : pour un ou tous les utilisateurs, et optionnellement sur un laps de temps donné. Cette fonction se nomme "generateReport()".
- et d'initialiser l'ensemble des variables (par exemples : chemins vers les autres scripts).

Une brève description des scripts contenus dans le répertoire Bin/, codés également en bash, est présentée dans les sections suivantes. Ces scripts permettent de gérer les utilisateurs, les rapatriements et les stratégies.

1.2.2 Bin/gestion_user.sh

- init : initialise les variables, notamment le fichier DATA/list_users.txt
- displayMenu : affiche le menu "Utilisateurs"
- listUsers : affiche la liste des utilisateurs existants
- addUsers : ajoute un utilisateur avec les champs : Nom, Prénom et adresse mail
- getNewId : permet de récupérer le dernier Id de la liste des utilisateurs et attribue un nouvel Id au nouvel utilisateurs
- delUser : permet de supprimer un ou plusieurs utilisateur(s) à la fois

1.2.3 Bin/gestion_repatriations.sh

- init : initialise les variables, notamment le fichier DATA/list_repatriations.txt
- displayMenu : affiche le menu "Rapatriements"
- listRepatriations : affiche la liste des rapatriements existants
- addRepatriation : ajoute un rapatriement avec les champs : Répertoire de destination, adresse du fichier à rapatrier

— getNewId : permet de récupérer le dernier Id de la liste des rapatriements et attribue un nouvel Id au nouveau rapatriement
— delRepatriation : permet de supprimer un ou plusieurs rapatriement(s) à la fois

1.2.4 Bin/gestion_strategies.sh

— init : initialise les variables, notamment le fichier DATA/list_strategies.txt
— displayMenu : affiche le menu "Stratégies"
— listStrategy : affiche la liste des stratégies existantes
— addStrategy : ajoute une stratégie avec les champs : Utilisateur, source du fichier à rapatrier et choix de la periodicité. Cette interface propose également à l'utilisateur de garder une trace du rapatriement ou non (log).
— getNewId : permet de récupérer le dernier Id de la liste des stratégies et attribue un nouvel Id à la nouvelle stratégies
— delStrategy : permet de supprimer une ou plusieurs stratégies à la fois
— addCron : ajoute au fichier mycron la commande à réaliser pour le rapatriement, conforme à la syntaxe demandée par crontab.

2 Notes d'utilisation

2.1 Les fonctions liées aux utilisateurs

Un utilisateur n'est ajouté que s'il fournit au minimum un des trois champs demandés. Un utilisateur est supprimé si et seulement si aucune stratégie ne lui est associée.

2.2 Les fonctions liées aux rapatriements

Un rapatriement n'est enregistré que si les 2 champs demandés sont remplis. De même, un rapatriement est supprimé si et seulement si aucune stratégie ne lui est associée.

2.3 Les fonctions liées aux stratégies

La modification d'une stratégie est possible pour la périodicité et la tracabilité. L'utilisateur doit remplir ces deux champs pour l'enregistrement de la modification.

2.4 Autre fonctionnalité

./main.sh [–help] [-h]

3 Intéraction entre les scripts

Lorsque l'utilisateur choisit d'afficher la liste des utilisateurs/rapatriements/stratégies, la seule option qui lui est proposée est de revenir au Menu Utilisateur. A partir de ce sous-menu, l'utilisateur peut également revenir au Menu principal.

Lorsqu'un(e) utilisateur/rapatriment/stratégie est ajouté(e)/modifié(e)/supprimé(e), le Menu Principal est affiché.

Au contraire, si l'ajout d'un(e) utilisateur/rapatriment/stratégie ou sa modification/suppression n'a pas été prise en compte, l'utilisateur retourne au sous Menu adéquat.

En résumé :
Reussite de la tâche : retour au Menu Principal.
Echec de la tâche : retour au sous-menu correspondant.

4 Génération d'un rapport

L'utilisateur peut générer un rapport en fonction des critères de recherche :

— Recherche par utilisateur. Par défault, ce champ est sur "ALL"
— Recherche selon une période. Par défault, la date de début vaut "01-01-0001" et la date de fin vaut la date actuel (au format "DD-MM-YYYY"

Une liste déroulante nommé "Type de rapport" est également présent. Actuellement, elle ne contient qu'une seule valeur : "Document". La génération d'un rapport nécessite :

— le fichier "Log/get_date.log"
— l'identifiant d'un utilisateur ou le mot clef "ALL"
— des modèles LaTeX au format .tex . Ces fichiers sont situés dans un répertoire caché "Reports/.model/Document"
— et le paquet pgf-pie.sty qui est aussi stocké dans Reports/.model/Document . Ce paquet est automatiquement copié dans le répertoire ∼/texmf/tex/latex/local/

Le nom du rapport produit prend la forme suivante : USER_date-début_date-fin.pdf.
Exemples : ALL_17-12-2015_01-01-2016.pdf ou Maxime_17-12-2015_01-01-2016.pdf

5 Informations à propos du logiciel

L'utilisateur peut accéder à différentes informations telles que la version du logiciel, le nom des auteurs et l'organisation dont ces derniers dépendent. Le double clique n'est pas autorisé et a pour conséquence la fermeture de l'interface.

6 Modules requis

Le programme requiert pdflatex, zenity et yad.

7 Perspectives d'évolution

7.1 Fonctionnalités liées à la génération du rapport

La génération du rapport pourrait comporter plus d'informations telles que des graphiques (cf. maquette ci-jointe).
Actuellement, un seul type de rapport est accessible ("Document"), mais nous pourrions en définir des nouveaux. Par exemple, les types "Website" et "Presentation (slides)". Ces types nécessiteront des nouveaux modèles au format .html et .tex .

7.2 Fonctionnalités liées au rapatriement d'une ressource

Actuellement nous utilisons "wget", mais d'autres outils pour le rapatriement existent et auraient pu être utilisés. Nous pouvons citer :

— "curl" qui fournit des status d'erreurs suplémentaires par rapport à "wget"
— "rsync" qui est efficace pour télécharger des fichiers imposants
— "ftp" . Exemple d'utilisation :
  ftp <TRAITEMENT> << BYEBYE
  newer
  BYEBYE
— "mirror.pl" qui est un paquet Perl

Une des améliorations envisageables serait de faire appel à un de ces outils selon le protocol, par exemple :

— "http" : utilisation de l'outil wget
— "ftp" : utilisation de l'outil rsync
— "local" : utilisation de la commande cp

Une autre fonctionnalité serait la possibilité d'utiliser des expressions régulières afin de rapatrier, par exemple, des fichiers dont le nom se terminerait par ".fastq", ou qui commencerait par "chrm_".

7.3 Structure du programme

Un fichier de configuration .init pourrait être mis en place pour l'utilisateur afin de paramétrer, à souhait, des chemins de répertoires tels que Data/ , Log/ et Reports/ ou bien aussi des valeurs par défault pour des variables telles que $typeReport.