

Utilisation du dépôt CRDAP

Instruction simplifiée décrivant le Système de synchronisation et de traitement des données en mer

Sommaire

1 Installation.....	2
1.1 Schéma général du système.....	2
1.2 Arborescence générale.....	3
1.3 Configuration d'une session de traitement sous Linux.....	3
1.4 Automatisation des traitements.....	4
2 Traitements des données.....	4
2.1 Traitements des données CTD/LADCP.....	4
2.2 Formatage des données.....	5
2.3 Formatage des données au format NetCDF.....	5
2.4 Plot des données.....	5
2.5 Génération du fichier .kml.....	5
2.5.1 Copie des fichiers sur serveur Nginx en local :.....	5
2.5.2 Fichier de configuration config.toml.....	6
3 Suivi des versions de ce document.....	7

Laboratoire : US191
Implantation : Brest

PROTOCOLE
Version 1
Page 2/7

1 Installation

Installer le logiciel de versions décentralisées git avec la commande :

```
$ sudo apt-get install git
```

Sous Linux, se placer dans le répertoire /mnt/campagnes, puis installer le dépôt CRDAP dans le sous répertoire *CRUISENAME* avec la commande suivante :

```
$ git clone --recurse-submodules https://forge.ird.fr/us191/crdap.git CRUISENAME
```

Basculer sur la branche propre à la campagne :

```
$ git checkout CRUISENAME
```

Créer un lien symbolique sous la racine « /m » :

```
$ cd /
```

```
$ sudo ln -s /mnt/campagnes /m
```

1.1 Schéma général du système

Mise en place de l'environnement

```
-.bashrc  
-local/etc/skel/.bashrc.CRUISENAME  
-contrab
```

Synchronisation des données	Formatage des données 1 (ASCII, XML, ODV)	Formatage des données 2 (NetCDF)	Plots des données	Génération .xml
<pre>-local/sbin/synchro.sh (données bord) -data-processing/ ctdSeaProcessing.ini (CTD/LADCP) -data-processing/ ctdSeaProcessing</pre>	<pre>-local/sbin/process-all.sh -data-processing/ [CTD,XBT,...]/[ctd,xbt, ...].pl -data-processing/ config.ini</pre>	<pre>-local/sbin/process-all.sh -local/sbin/ oceano2oceansites.pl</pre>	<pre>-local/sbin/process-all.sh -local/sbin/python/ python-plots.sh -local/sbin/python/ plots.py -local/sbin/python/ scatter.py</pre>	<pre>-local/sbin/process-all.sh -local/sbin/linux/ cruiseTrack2kml-linux- amd64 -data-processing/ local.toml</pre>

Fichiers à modifier en début de campagne

Laboratoire : US191
Implantation : Brest

PROTOCOLE
Version 1
Page 3/7

1.2 Arborescence générale

└─ data-adjusted	= Données ajustées
└─ data-final	= Produits finis
└─ data-processing	= Données prétraitées
└─ data-raw	= Données brutes
└─ DOCUMENTS	= Documents de campagne
└─ local	= Utilitaires de traitement et de synchronisation
└─ tmp	= Fichiers temporaires

1.3 Configuration d'une session de traitement sous Linux

Le principe de cette opération est d'obtenir dans son environnement shell l'ensemble des alias nécessaires aux traitements et à la mise en forme des données.

Dans le fichier `.bashrc` («`~/bashrc`»), en fin de script, définir les variables d'environnement pour la campagne et le fichier à sourcer :

```
# CRUISE variables
export DRIVE=/mnt/campagnes
export CRUISE=RESILIENCE
export CRUISEid=RES

# Source CRUISE definitions
if( -f ${DRIVE}/local/etc/skel/.bashrc.${CRUISE} ) then
    source ${DRIVE}/local/etc/skel/. bashrc.${CRUISE}
endif
```

Le fichier à sourcer se trouve sous « `CRUISE/local/etc/skel/.bashrc.CRUISENAME` ». Ce fichier contient tous les alias nécessaires aux traitements. Ce script peut être modifié en début de campagne pour ajouter ou supprimer des alias.

Deux scripts shell (sous « `CRUISE/local/sbin/`») permettent de lancer l'ensemble des tâches de traitement :

- Le script `synchro.sh` récupère les données sur le réseau du bord et les copie dans les répertoires de `data-raw` et de `data-processing`. Ce script doit être modifié en début de campagne de manière à s'ajuster à l'architecture réseau du navire.

Laboratoire : US191
Implantation : Brest

PROTOCOLE
Version 1
Page 4/7

- Le script process-all.sh va réaliser l'extraction et l'archivage sous différents formats (ASCII, XML, ODV, NetCDF) une fois la collecte effectuée par le script synchro.sh.

1.4 Automatisation des traitements

La synchronisation des données ainsi que les traitements sont effectués automatiquement à l'aide de la crontab. Pour modifier son contenu, utiliser la commande `crontab -e` et `crontab -l` pour la visualiser.

```
$ crontab -l
```

```
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan, feb, mar, apr ...
# | | | | .----- day of week (0 - 7) (Sunday=0 or 7) OR
sun, mon, tue, wed, thu, fri, sat
# | | | | |
# * * * * * command to be executed
# all processing cruise 3 times per day (local time: 8, 16, 22h)
SHELL=/bin/sh
CRUISE=RESILIENCE
DRIVE=/mnt/campagnes
CRUISEid=RES
5 9,11,17,22 * * * /mnt/campagnes/${CRUISE}/local/sbin/synchro.sh $CRUISE $DRIVE
$CRUISEid > /mnt/campagnes/${CRUISE}/local/logs/synchro.log 2>&1
30 9,11,17,22 * * * /mnt/campagnes/${CRUISE}/local/sbin/process-all.sh $CRUISE $CRUISEid
$DRIVE > /mnt/campagnes/${CRUISE}/local/logs/process.log 2>&1

0 8,12,21 * * * rsync -av --exclude '.git' /mnt/campagnes/${CRUISE}/ /media/science/WD\
2T/campagnes/${CRUISE}/ > /mnt/campagnes/${CRUISE}/local/logs/backup.log 2>&1
```

Lors de la mission, vérifier régulièrement les fichiers de log pour voir d'éventuels problèmes (droits en lecture, variables nouvelles,...)

2 Traitements des données

2.1 Traitements des données CTD/LADCP

Les données de la CTD + LADCP sont enregistrées localement sur le PC d'acquisition puis copiées sur le disque réseau par le programme Matlab « ctdSeaProcessing » après chaque profil. Le script Matlab « ctdSeaProcessing » permet également de réaliser le traitement des profils CTD (via SBEDDataProcessing) et LADCP (via logiciel Visbeck). Le logiciel utilise le fichier de configuration *ctdSeaProcessing_CRUISEid.ini* (sous data-processing) qui doit être modifié en début de campagne. Le logiciel est ensuite lancé à chaque fin de station via Matlab.

Laboratoire : US191
Implantation : Brest

PROTOCOLE
Version 1
Page 5/7

2.2 Formatage des données

Actuellement, il existe un script Perl générique pour chaque instrument. Les scripts sont dans les dossiers relatifs à chaque instrument sous « data-processing ». Ils permettent le formatage des données sous différents formats : ASCII, XML, ODV. La configuration des scripts est décrite dans un fichier externe « config.ini » se trouvant à la racine du répertoire « data-processing ». Ce fichier « config.ini » est à modifier en début de campagne afin de renseigner correctement les attributs globaux et en fonction de la configuration des instruments de mesures. Chacune des sections ([ctd], [ctd-all], [btl],...) décrit la structure des données à extraire dans les fichiers. Ce formatage est effectué à l'aide du script « process-all.sh ».

2.3 Formatage des données au format NetCDF

Les données sont ensuite formatées au format NetCDF via le script perl oceano2oceansites (sous « CRUISE/local/sbin/ »). Les variables créées suivent le code roscop situé en fin du script « oceano2oceansites.pl », des variables peuvent être ici ajoutées le cas échéant. Ce formatage est effectué à l'aide du script « process-all.sh ».

2.4 Plot des données

Des plots python sont ensuite générés. Les scripts sont sous « CRUISE/local/sbin/python ». Ils génèrent d'une part les plots des profils CTD, XBT et LADCP (lancés par le script shell « python-plots.sh », puis « plots.py »), et d'autre part le tracé de la route du TSG (réalisé par « scatter.py »). Ces scripts peuvent être ajustés au cours de la campagne. Ces plots sont effectués à l'aide du script « process-all.sh ».

2.5 Génération du fichier .kml

Enfin, un fichier .kml est généré avec les données de la campagne. Ce fichier est réalisé avec le script « cruiseTrack2kml-linux-amd64 » (sous « CRUISE/local/sbin/linux »). Ce fichier est généré à l'aide du fichier de configuration « local.toml » (sous data-processing) et utilise les figures python précédemment générées.

2.5.1 Copie des fichiers sur serveur Nginx en local :

Dans un premier temps il est nécessaire de créer des liens symboliques sous la racine sur serveur NGINX. Il faut créer un répertoire RESILIENCE sur le serveur web NGINX puis créer les 3 liens suivants avec la commande `ln -s <source> <dest>`

Soit :

```
> cd /var/www/html/RESILIENCE
> ln -s /mnt/campagnes/RESILIENCE/data-processing/CELERITE/plots/ CELERITE
> ln -s /mnt/campagnes/RESILIENCE/data-processing/CTD/plots/python CTD
> ln -s /mnt/campagnes/RESILIENCE/data-processing/THERMO/plots/ THERMO
```

Laboratoire : US191
Implantation : Brest

PROTOCOLE
Version 1
Page 6/7

Démarrer, stopper, redémarrer, le serveur web:

```
$ sudo service nginx <start|stop|restart>
```

2.5.2 Fichier de configuration config.toml

Le script « cruiseTrack2kml-linux-amd64 » utilise le fichier de configuration « config.toml » (sous data-processing). Ce fichier doit être modifié en début de campagne en fonction de la campagne et des fichiers utilisés. Le serveur Web ne permet pas de « browser » le contenu d'un répertoire, il faut donc entrer l'URL complète pour visualiser une image depuis son navigateur.

Laboratoire : US191
Implantation : Brest

PROTOCOLE
Version 1
Page 7/7

3 Suivi des versions de ce document

Rédacteur		Approbateur	
Nom :	Pierre Rousselot	Nom :	
Fonction :	IE	Fonction :	

Dates	Versions	Chapitres concernés	Commentaires et modifications
Février 2022	01	tous	Création

Relecteur	Date