RAPPORT DE STAGE

MISE EN PLACE DU SYSTEME DE GESTION D'OBSERVATIONS EN AMONT DE LA CHAINE OPERATIONNELLE DU MODELE ALADIN

Stage effectué par	Sous la supervision de
BOUZOUITA Nihed	**PUECH Dominique**
IP/DSMB	**TAILLEFER Françoise**
INM-Tunisie	GMAP/SCEM/Météo France

Du 30 Avril au 28 Juin 2002 Au GMAP/SCEM de METEO-France (Toulouse/France)

INTRODUCTION

Certes l'un des maillons les plus importants dans la chaîne de la prévision numérique du temps est l'Assimilation des données d'observation qui constituent le point de départ dont l'exactitude et la précision sont fort influentes sur la qualité de la prévision qui sera élaborée par la suite. D'où l'importance d'y attacher un intérêt capital pour assurer une bonne utilisation et gestion des données requises.
Les dernières années témoignent du développement permanent du traitement des données à Météo-France, notamment dans les modèles ARPEGE et ALADIN.

Le stage que j'ai effectué du 30 Avril au 28 Juin 2002 au service GMAP (Groupe de Modélisation pour l'Assimilation et la Prévision) à Météo-France (Toulouse - France) sous la supervision attentionnée de *Françoise TAILLEFER* et *Dominique PUECH*, relate de la base de données ODB (Observational DataBase) développée et implémentée par *Sami Saarinen* (CEPMMT-GB) dans les modèles ALADIN et ARPEGE. Ce concept de nouveau format de données est encore en cours de développement.

Ainsi, en première étape, mon objectif était de me familiariser avec la nouvelle ODB à travers les requêtes SQL et les scripts d'assimilation de données CANARI puis on s'est proposé d'élaborer une procédure de conversion de formats de ces données pour une manipulation plus conviviale.

FAMILIARISATION AVEC LA BASE DE DONEES ODB

DEFINITION ET BASE DE L'ODB

ODB, Observational DataBase, est un concept récemment développé et adopté dans les modèles numériques ARPEGE et ALADIN de Météo-France grâce à son optimisation du stockage et de la manipulation de tout type d'observation.

En effet cette technique permet de réduire considérablement le temps d'accès et de manipulation des diverses données, notamment grâce au langage *ODB/SQL*. En effet, ce concept de base de données organise les données d'observations selon une distribution relationnelle flexible et économique dont les avantages sont :
- Espace disque bien géré et économie en mémoire
- Sélection interactive des données
- Accès parallélisé aux données
- Changement facile de la structure des données
- Intégrité et connectivité (liens)
- Structure propice au développement

Par contre, il importe de noter que ce concept défavorise la précision des données ainsi que le format des données puisque chacun peut le reconfigurer à son aise.

La base ODB est constituée de plusieurs structures de données nommées *tables*. Chaque table est constituée de colonnes et autant de lignes que de données observées disponibles. Ainsi, la table HDR (Header Data Report) contient des informations d'identification des données (type d'observation, identifiant de la station, coordonnées géographiques...)et la table BODY contient des informations sur un seul type de d'observation (type de variable, valeur observée...).

ODB se divise en plusieurs types de bases de données (ensemble de tables) dont les plus utilisés en assimilation : ECMA (forme élargie de l'ancien format CMA de la base de données), ECMASCR et CCMA (formes réduites du format CMA).

Les données d'observation sont rangée dans l'ODB telles dans des vecteurs de tables de même type connectées entre elles par des colonnes contenant les divers liens qui montrent où sont stockées les données dans les autres tables. Ainsi, la colonne body@link de la table HDR indique les autres tables BODY liées à l'information qu'elle contient.
Les tables sont rangées en cascade (One-looper) ou alignées dans le cas où elles contiennent le même nombre d'éléments. La base de données est partagée en sous-ensembles nommés *pools* qui traduisent la répartition des données aux divers processeurs du calculateur sur lequel tourne le modèle. Ce partage se fait suivant les bandes latitudinales, temporelles ou autres.

LES VUES

Les données de la base ODB sont accessibles en utilisant des requêtes ODB/SQL pour une éventuelle extraction. Une requête fournit un ensemble de données nommé *vue*.

Exemple

SET $airep=2	*(Initialisation)*
CREATE VIEW visu **AS**	*(Nom de la vue)*
SELECT	
obstype, lat, lon, varno, obsvalue	*(Colonnes souhaitées)*
FROM hdr, body, update	*(Tables utilisées)*
WHERE (lat>0)	*(Sélection et Options*

AND obstype=$airep ; *spéciales)*

Le résultat obtenu de cette requête est une matrice de 5 colonnes correspondant aux 5 données sélectionnées (partie SELECT) et autant de lignes que de valeurs observées transmises dans les messages AIREP dans l'hémisphère nord.
Les vues sont placées dans des fichiers .sql.

Une base de données ODB est un répertoire contenant deux fichiers ASCII (**db_type.ddl** qui contient les valeurs par défaut de toutes les colonnes et les variables ODB, et **db_type.sch** qui contient le schéma de création de la base) et un ou plusieurs sous-répertoires selon les *pools*.
ODB DANS ARPEGE

L'accès à la base de données dans ARPEGE se fait en utilisant des procédures telles que **OPENDB** (pour l'ouverture et l'initialisation de la base), **CLOSEDB** (fermeture et mise à jour de la base), **GETDB** (Extraction de données dans des matrices allouées temporairement) et **PUTDB** (restauration des données dans la base).

Pour ARPEGE, la base de données est structurées sur DELAGE dans des répertoires **/chaine/mxpt/mxpt001/arpege/oper/assim/yyyy/mm/dd/rhh** où yyyy/mm/dd désigne la date et rhh le réseau des observations. Les fichiers *.dd et *.sch ainsi que les pools sont archivés dans des fichiers **db_type.tar**.

Pour la familiarisation avec ces manipulations de la base ODB, j'ai utilisé l'interface "**Viewer**" (implémentée sur ANDANTE et KAMI) pour la compilation et la visualisation des vues passant par les étapes suivantes:

1- Rapatriement des données dans le répertoire d'espace temporaire: $UTEMP/DATA
2- Rédaction de la requête SQL sous $UTEMP/DATA/Locview.sql
3 - Extraction des fichiers *.dd et *.sch et les pools (14 sur le VPP5000) de l'archive .tar
4 - Lancement du Viewer qui fournit un fichier rapport *.rpt contenant la vue en question.

SCRIPT D'ANALYSE PAR INTERPOLATION OPTIMALE: CANARI

Dans une seconde étape, j'ai procédé au lancement du script utilisé dans l'assimilation des données d'observation et nommé **CANARI** (Code d'Analyse Nécessaire à Arpège pour ses Rejets et son Initialisation). J'ai utilisé uniquement la configuration **701** qui est la plus utilisée et qui correspond à l'Interpolation Optimale des données.

Cette configuration fournit une combinaison des informations contenues dans l'ébauche du modèle et les données d'observation. Après l'analyse par cette configuration, le fichier d'observation contient l'information sur l'écart entre l'ébauche et l'analyse ainsi que les "flags" (codes qualité) générés durant l'analyse et le contrôle et qui fournissent une information supplémentaire quant à l'utilisation et la fiabilité des données initiales. Les principaux fichiers obtenus par l'exécution de cette configuration 701 sont archivés sur DELAGE dans le fichier
/chaine/mxpt/mxpt001/ald_horaire/oper/yyyy/mm/dd/CMAFOCF.tar.

La ligne de commande du lancement de la configuration 701 est:

$MASTER -c701 -vmeteo -maladin -eANAL -t1800 -ft0 -aeul

Ainsi, la partie dans ce script concernant la configuration 701 comporte les étapes suivantes:

1- Sélection des observations et initialisation des statistiques d'erreurs
2- Calcul des résidus et des écarts
3- Contrôle de la qualité des observations
4- L'analyse des champs
5- Statistiques d'erreurs et listing de sortie.

Cette configuration utilise:
- un fichier d'ébauche de 6h: **ICMSHANALINIT**
- un fichier de couplage **ELSCFANALBC0000**
- un fichier d'observations **cmat01**
- des fichiers de corrélations verticales et paramètres

Ainsi on obtient:
- Un état analysé **ICMSHANAL+0000**
- Des informations sur les observations (écarts obs/model, flags d'utilisations et de fiabilité) et des diagnostics.

REFORMATTAGE DES DONNEES D'OBSERVATIONS DANS ODB

Les fichiers d'observations ne sont accessibles qu'à travers une interface de visualisation telle le **Viewer**. Ils sont stockés sous forme non-ASCII (CMA, Central Memory Array). Ainsi, on se propose dans la suite de mon stage d'établir une procédure (que j'ai nommé ***mandaodb_ascii***) qui permet de reformater les données extraites dans une vue d'ODB en format ASCII à fin d'une éventuelle visualisation. Les variables sélectionnées pour cette transformation sont les suivantes, mais la liste n'est pas exhaustive et peut être éventuellement étendue:

- TABLE HDR

Nom Variable en ODB	*Nom Variable dans le Code*	
obstype	MDBOTP	type d'observation
date	MDBDAT	date de l'observation
time	MDBETM	heure de l'observation
statid	MDBSID	numéro d'identification de la station
lat	MDBLAT	latitude de la station (en radians)
lon	MDBLON	longitude de la station (en radians)
stalt	MDBALT	altitude de l'observation
numlev	MLNKH2B	nombre des niveaux de pression concernant l'observation dans Body

- TABLE BODY

Nom Variable en ODB	*Nom Variable dans le Code*	
varno	MDBVNM	type de paramètres (1: géopotentiel, 2: T, 3: u, 4: v, 11: Tsurface, 29: Hu, 39, T2m, 41: U10m, 42: V10m, 58: Hu2m, 80: RR6h, 110: Psurface)
anflag	MDBFLG	code qualité analyse (contenant les 11 flags de qualité figurant dans le tableau ci-dessous.
press	MDBPPP	pression correspondant à la coordonnée verticale en Pa
mf_log_P	MDBPIO	pression d'utilisation en CANARI
obsvalue	MDBVAR	valeur observée

Les Flags qualité analyse:
Ces codes sont représentés sur un bit dans anflag.

varGC	flag variationel de qualité
final	flag final
fg	flag de l'ébauche (first-guess)
blacklist	flag de la liste noire (liste des observations rejettées)
ups	flag d'utilisation par analyse de pression de surface
uvt	flag d'utilisation par analyse de vent et température
uhu	flag d'utilisation par analyse d'humidité
ut2	flag d'utilisation par analyse de température à 2m
uh2	flag d'utilisation par analyse d'humidité à 2m
uv1	flag d'utilisation par analyse de vent à 10m
usst	flag d'utilisation par analyse de température de surface de la mer

- TABLES UPDATE ET ERRSTAT

Nom Variable en ODB	*Nom Variable dans le Code*	
fg_depar	MDBOMF	écart entre l'observation et l'ébauche
an_depar	MDBOMN	écart entre l'observation et l'analyse
final_obs_error	MDBFOE	erreur d'observation finale
fg_error	MDBFGE	erreur de first-guess

Ces paramètres sont extraits dans deux vues **select_hdr** et **select_body** selon les tables qui les contiennent..
Cette définition des vues est insérée dans la programme **CTXINIT.F90**.
Le code de la procédure *mandaodb_ascii* est inséré dans la programme pilote **ODB_VISU.F90**.
Ces deux programmes ainsi que les requêtes SQL (select_hdr.sql et select_body.sql) sont copiés dans les répertoires relatifs sous **$HOME/PACK** pour leur compilation et exécution..
La compilation se fait en utilisant **PACK.ksh** (script Shell de compilation)
L'exécution permettra de fournir des listings des observations sous la forme:
YYYYMMDD HH NTOTOBS NTOTPARAM
MDBOTP MDBLAT MDBLON MDBDAT MDBETM MDBSID MDBALT MDBNLV MDBVNM MDBFLAG(bit/bit) MDBPIO MDBVAR MDBOMF MDBOMN MDBFOE MDBFGE (ligne répétée autant de fois que de paramètres)

Exemple:

20020607 0 993 5369

1 53.250 14.533 20020606 235000 03302 11. 7

1 1 1 1 1 1 0 0 0 0 0 0 100190 100190 0.00 15.86 15.69 78.45 82.38

39 1 1 1 1 0 1 0 1 0 0 0 100060 99815 288.85 0.90 0.90 1.40 3.00

58 1 1 1 1 0 0 1 0 1 0 0 100060 99815 0.87 0.08 0.08 0.10 0.20

...... (Exemple tronqué)

CONCLUSION

Durant ce stage de deux mois, j'ai pu connaître la structure de la base de données ODB. Les diverses