IASI

THALES INFORMATION SYSTEMS

IA-DP-2100-9549-THA
Edition : 02 Date : 22/08/2002
Révision : 02 Date : 25/06/2004
MT : X Code diffusion : E

Réf. : -

DOSSIER DES PERFORMANCES
DOSSIER DE PERFORMANCES DU LOGICIEL OPS IASI

Rédigé par : PASCAL Jean-Luc	THALES IS	le : 27/7/4	
Validé par : AYER Patrick	THALES IS	le : 27/07/04	
Pour application : MORENO Richard	DTS/MID/VM/TD	le : 27/7/4	

18, avenue Edouard Belin 31401 Toulouse CEDEX 4 Tél. 05 61 27 31 31

BORDEREAU D'INDEXATION

CONFIDENTIALITE : NC	**MOTS CLES : Performances, Optimisation, méthodologie, IASI, Traitement d'images**

TITRE DU DOCUMENT : Dossier des Performances
Dossier de Performances du Logiciel OPS IASI

AUTEUR(S) : PASCAL Jean-Luc THALES IS

RESUME : Ce document décrit la démarche globale, les procédures et les essais mis en oeuvre tout au long du projet pour le suivi et la maîtrise des performances de la chaîne de traitement IASI de niveau 1.

DOCUMENTS RATTACHES : Ce document vit seul.	**LOCALISATION :**

VOLUME : 1	**NBRE TOTAL DE PAGES : 47** **DONT PAGES LIMINAIRES : 6** **NBRE DE PAGES SUPPL. : 0**	**DOCUMENT COMPOSITE : N**	**LANGUE : FR**

GESTION DE CONF. : F	**RESP. GEST. CONF. : GOMEZ MH**

CAUSE D'EVOLUTION : Mise à jour suite aux tests de validation algorithmiqsue

CONTRAT : 01/8937

SYSTEME HOTE : Microsoft Word 8.0b, \\NWTLS10\PROJETS\PROJETS\IASI\Modèles CNES\CnesIndustriel97.dot v2.0.4

DIFFUSION INTERNE

Nom	Sigle	BPi	Observations
BLUMSTEIN Denis	DCT/PO/EV	2504	
~~CHALON Gilles~~	~~DCT/PO/EV~~	~~2504~~	
PONCE Ghislaine	DCT/PO/EV	2504	
~~SEGALEN Barbara~~	~~DCT/PO/EV~~	~~2504~~	
MARQUIER Henry	DCT/PS/TIS	1321	
MORENO Richard	DCT/PS/TIS	1321	
BAILLY Isabelle	DCT/PS/TIS	1321	
~~RAYSSIGUIER Michel~~	~~DCT/PS/TIS~~	~~1321~~	
~~RICHARD Pascal~~	~~DSI/EP/SL~~	~~3517~~	
MATHIEU Nathalie	EUROGICIEL	1415	
GOMEZ Marie-Hélène	DCT/PS/TIS	1321	
GLEYZES Jean-Pierre	DCT/PS/TIS	1321	

J. LOUSTAD } EUMETSAT
JACOBS

DIFFUSION EXTERNE

Nom	Sigle	Observations
AYER Patrick	THALES IS	
BOBIN Serge	THALES IS	
PASCAL Jean-Luc	THALES IS	

MODIFICATION

Ed.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
02	02	25/06/2004	- PASCAL Jean-Luc THALES IS Mise à jour suite aux tests de validation algorithmiqsue
02	01	17/09/2003	- PASCAL Jean-Luc THALES IS Mise à jour suite aux TU et aux tests de validation
02	00	22/08/2002	- PASCAL Jean-Luc THALES IS CABANE Philippe THALES IS Mise au format GDOC maj suite aux FEPS EUM-05, RM-09, PR-01 description des tests sur les librairies ESSL et metop
01	00	14/06/2002	- PASCAL Jean-Luc THALES IS CABANE Philippe THALES IS Création du document

SOMMAIRE

GLOSSAIRE ET LISTE DES PARAMETRES AC & AD

AVHRR	Advanced Very High Resolution Radiometer : radiomètre avancé à très haute résolution (visible et infrarouge) sur les satellites polaires
CCD	Corner Cube Direction : direction du coin de cube (miroir mobile de l'interféromètre)
CGS	Core Ground Segment : segment-sol développé par ALCATEL sous contrat d'EUMETSAT, et dans lequel l'OPS ira s'insérer
EUMETSAT	EUMETSAT est une organisation intergouvernementale regroupant 17 nations européenne, dont l'objectif est l'établissement, le maintien et l'exploitation des systèmes européens de satellites météorologiques opérationnels
FOV sondeur	Field Of View : champ de vue
IASI	Infrared Atmospheric Sounding Interferometer : interferomètre de sondage atmosphérique dans l'infrarouge.
IPSF	Instrument Point Spread Function : forme du pixel IASI
ISRFEM	Instrument Spectral Response Function Estimation Model
METOP	Série de satellites météorologiques opérationnels en orbite polaire. IASI est l'un des instruments de METOP.
NRT	Near Real Time
OPS	Logiciel Opérationnel (Operational Software) : correspond au IASI level 1 PPS dans les glossaires d'EUMETSAT. PPS=Product Processing Software
PPF	Product Processing Facility
TU	Test Unitaire

Liste des paramètres AC :

Liste des paramètres AD :

1.GENERALITES

1.1.DOCUMENTS DE REFERENCE ET APPLICABLES

La liste des documents constituant le référentiel du projet OPS-IASI est détaillée dans la « Liste Unique » du Logiciel OPS-IASI [DR100].

1.2.OBJECTIF

De part les algorithmes mis en œuvre et le volume de données à traiter, le sous-système OPS-IASI L1 est le sous-système le plus critique du CGS en terme de performances. EUMETSAT l'a ainsi clairement identifié comme le PPF (Product Processing Facility) critique dans ce domaine. L'OPS-IASI est ainsi le seul PPF du CGS à fonctionner en mode statique, mode dans lequel le sous-système est responsable de la parallélisation du traitement des données.

Ce document a pour objectif présenter la démarche et les tests réalisés pour évaluer et de vérifier que l'architecture informatique proposée permet d'atteindre les objectifs de performance.

Ce document est découpé de la façon suivante suivants:

le chapitre 1 est le chapitre introductif de ce document.

le chapitre 2 rappelle les exigences de performances que doit respecter l'OPS-IASI,

le chapitre 3 présente les hypothèses extraites des documents du CNES sur lesquelles nous nous sommes basés ; puis, les essais et les résultats de benchmarks obtenus par le CNES.

le chapitre 4 présente la démarche globale de suivi des performances que nous mettons en œuvre,

le chapitre 5 présente les résultats et leur analyse fine phase par phase. Ce suivi permet de vérifier à chaque étape que les performances de la chaîne sont atteintes.

Ce dossier est conçu pour être enrichi à chaque étape du développement, le chapitre 5 doit ainsi évoluer pour présenter les résultats obtenus.

2.EXIGENCES DE PERFORMANCES

2.1.PRESENTATION DU BESOIN

L'OPS est une chaîne opérationnelle qui doit fonctionner dans un contexte NRT (Near Real Time). Ceci impose de devoir **traiter N secondes de données en moins de N secondes** (E_CTX2 du document de spécification technique de besoin [DA2]). Ainsi, au regard des traitements lourds réalisés, une attention particulière doit être portée sur les aspects performances de la chaîne de traitement. Le respect de cette exigence permet d'assurer que la chaîne OPS s'intégrera correctement dans le concept de «pipeline» du CGS qui impose que « les produits doivent être disponibles 2h15 après acquisition instrument ».

2.2.OPTIMISATIONS REQUISES

Les optimisations présentées dans le document [DA2] sont considérées comme une base dans notre étude. Certaines sont directement intégrées dans la description des algorithmes, les autres peuvent être considérées comme des règles de codage à implémenter au cours de la réalisation. Ces optimisations sont les suivantes :

- l'optimisation de l'ordre de rangement des tableaux,
- les FFT complexes -> réels et réels -> complexes à utiliser dans les algorithmes 22_SOS 23_SSD (A1345, A1332),
- l'optimisation du processus de corrélation,
- la factorisation des calculs dans les algorithmes XX_INIT et YY_CONF,
- le pré-calcul des coefficients pour les transformées de Fourier de taille fixe (utilisation de la librairie ESSL),
- l'optimisation de l'interpolation par spline dans l'algorithme 35_S1B [pour la recherche des points (points initiaux quasi-réguliers) et les pré calculs],
- l'optimisation de l'interpolation par spline dans l'algorithme 22_SOS [pour la recherche des points (points initiaux réguliers) et les pré calculs (points initiaux réguliers)],
- l'utilisation de multiplications à la place de divisions par des constantes,
- le traitement 43_ISF à exécuter en début de la chaîne produit,
- l'optimisation du stockage de la banque spectrale en mémoire,
- l'optimisation des interpolations dans la banque spectrale,

3.HYPOTHESES ET PREMIERE ANALYSE

3.1.INTRODUCTION

La chaîne OPS IASI a fait l'objet d'une première analyse et de benchmarks poussés afin de valider la faisabilité du système. Ces études réalisées par le CNES sont décrites dans les documents [DA3] et [DR4]. Elles se basent sur une analyse à la fois théorique et pratique des performances, ce qui a permis à travers des recoupements d'affiner les résultats obtenus.

L'objet de ce chapitre est de résumer brièvement ces résultats ainsi que les méthodes choisies. Pour plus de précisions, on se reportera aux documents [DA3] et [DR4].

3.2.HYPOTHESE DE BASES

Le tableau ci-après liste l'ensemble des paramètres qui ont une influence sur les performances du logiciel opérationnel. Les valeurs de ces paramètres données dans le tableau ci-dessous seront utilisées dans tous les tests. Elles sont extraites du document [DA3].

VARIABLE	DESCRIPTION	VALEUR
LN	Nombre de lignes dans un granule	22
SN	Nombre de sous-cycles visées dans une ligne	30
Col_Img	Nombre de colonnes dans l'image IASI	64
Ligne_Img	Nombre de lignes dans l'image IASI	64
Torbite	Durée d'une orbite METOP (s)	6095 (102 mn)
Nb_Ech_Fen	Nb échantillons de la fenêtre spectrale ISRFEM	150
Surech_Fen	Facteur de suréchantillonnage fenêtre spectrale ISRFEM	8
PN	Nombre de pixels	4
CCD	Nombre de directions de coin de cube	2
Nb_Ech	Nombre d'échantillons du spectre	8500
Ech_Fcs	Nombre d'échantillons des fonctions de calibration spectrales qui sont sous-échantillonnées	141
Ech_Af_BD	Nombre d'échantillons dans les fonctions d'apodisation dans la banque spectrale	300
Ech_Af	Nombre d'échantillons dans les fonctions d'apodisation appliquées dans 37_S1C	1024

VARIABLE	DESCRIPTION	VALEUR
Nb_Af	Nombre de fonctions d'apodisation	141
Surech_spect1	Facteur de suréchantillonnage du spectre par 22_PRODUIT	5
L_AVHRR	Nombre de lignes AVHRR de part et d'autre de la ligne IASI pour pouvoir traiter une ligne IASI	80
nbligAVHRR	Nombre de lignes AVHRR nécessaire au traitement d'un granule	600
C_AVHRR	Nombre de colonnes AVHRR	2048
NbLinAvhrr	Nombre moyen de lignes AVHRR pour couvrir l'image IIS	60
NbColAvhrr	Nombre moyen de colonnes AVHRR pour couvrir l'image IIS	80
Dlin_Avhrr	Plage de variation en ligne de l'image IASI dans l'image AVHRR	10
Dcol_Avhrr	Plage de variation en colonne de l'image IASI dans l'image AVHRR	10
Nb_Bandes	Nombre de bandes	3
Nb_Seg_Cplx	Nombre de segments pour les moyennes de spectre cplx enlevés	25
Taille_code	Nombre de Mo occupés par 1 échantillon du spectre codé	1,0375E-06 Mo
Taille_amort	Taille des tableaux d'amortissement	~ 100 par tableau
NbPts_Ipsf	Nombre de points pour décrire les IPSF dans le repère imageur	100x100
NbPts_imag_filter	Nombre de points pour décrire le filtre imageur	800
NbRrs	Nombre de surfaces radiatives vues par le corps noir chaud : IDefBBNbRrs	10
IdefSdbGridNbLin IdefSdbGridNbCol	Nombre de lignes et de colonnes de la grille décrivant les variations possibles de l'axe interférométrique	11*11
Nitera	Nombre d'itération pour calculer décalage spectral : IdefSsdNitera	5
NbEchBand3	Nombre maximal d'échantillons en bande 3	3600
FLS	Longueur de l'historique de la position de l'axe interférométrique : 1 orbite	800
IdefCcsNbIterMax	Nombre maximum d'itérations pour l'algorithme 41_CCS	10

Tableau 1 : Paramètres et Valeurs des algorithmes de l'OPS

De plus, nous considérerons que les optimisations spécifiées au §2.2 sont mises en œuvre de façon systématique dans la chaîne de traitement.

3.3.CONFIGURATION MATERIELLE

La machine utilisée pour les tests est très proche de la configuration finale du CGS : nœud quadri-processeur IBM power 3 cadencé à 375 Mhz avec 4 Go de mémoire RAM. Dans le cas du CGS, la mémoire est de 2 Go, mais au vue du volume de données à stocker en mémoire, ce facteur ne semble pas déterminant pour les résultats.

Ce processeur fonctionne de manière native en 64 bits et ses caractéristiques générales sont récapitulées dans le tableau ci-dessous :

Caracteristiques	Valeurs
nombre d'unités de calculs	**Flottants : 2** **entiers : 3** **chargement/déchargement : 2**
cache L1	**Instructions :32 Ko** **Données : 64 ko**
cache L2	**8 Mo**
caractéristiques de calcul	**8 instructions max par cycle**
Temps d'une opération de base	**3 cycles**

Tableau 1 : Caractéristiques du Processeur Power3

Grâce à son débit mémoire particulièrement important, ce processeur permet de maintenir une puissance proche de la puissance de crête sur les calculs flottants.

De plus, la taille importante du cache L2 ainsi que le nombre d'unités de calcul permettent d'atteindre des puissances de calcul supérieures à celles des processeurs ayant un cadencement plus élevé.

3.4.ESSAIS REALISES

3.4.1.Benchmark

Différents tests on étés menés par le CNES pour évaluer la rapidité de la configuration matérielle envisagée :

bench sur des opérations simples (cf [DA3]),

bench sur des opérations complexes dans une contexte différent (FFT, spline) (cf [DA3]),

simulation globale de l'algorithmie réelle de la chaîne (cf [DR4])..

L'ensemble de ces tests a permis d'approximer chaque opérateur par un nombre d'opérations de base qui est utilisé par le CNES dans l'analyse théorique.

Naturellement, les tests globaux sont les plus représentatifs, ils ont permis d'affiner la vitesse du processeur dans un environnement représentatif et pour les calculs les plus dimensionnants (FFT). On constate ainsi que la vitesse obtenue est de l'ordre de **340 Mflop/s avec la librairie ESSL**. Ces tests permettent également d'évaluer globalement la vitesse de calcul réelle de l'algorithmie IASI (80 s de données sont traitées en 81 s). Ceci correspond approximativement à une vitesse de l'ordre de **430 Mflop/s.**

3.4.2.Analyse Théorique

L'analyse des algorithmes décrits dans le document [DA7] permet d'évaluer grâce aux tests précédents le nombre d'opérations «élémentaires» de chaque étape de la chaîne de niveau 1.

3.4.2.1.Dimensionnement CPU

Les tableaux ci-dessous récapitulent les performances des algorithmes à implémenter dans des chaînes IASI niveau1.

Chaîne ISFRFEM

Algorithme	Dimensionnement (Mflop)	
	%traitement complet	1 s de données
20_DOC: « décodage de la fenêtre spectrale »	<0.1%	0,13
22_SOS : « suréchantillonnage de la fenêtre spectrale »	<0.1%	0,56
23_SSD : « détermination des décalages spectraux »	**2%**	9,48
21_SSS : « sélection des déterminations des décalages spectraux »	<0.1%	0,34
24_IAX : « détermination de la position de l'axe interférométrique instantané »	<0.1%	0,00
25_FAX : « détermination de la position de l'axe interférométrique filtré »	<0.1%	0,00
43_ISF : « interpolation des fonctions spectrales »	<1%	3,46
TOTAL chaîne ISRFEM	**3%**	**13,97**

Tableau 2 : Dimensionnement de la chaîne ISRFEM

La performance de la chaîne ISRFEM dépend essentiellement de :

l'algorithme de détermination des décalages spectraux (algorithme à base de FFT et d'interpolation spline),

l'algorithme d'interpolation des fonctions spectrales (algorithme à base d'interpolation spline).

Chaîne Produit

Algorithme	Dimensionnement (Mflop)	
	% traitement complet	1 s de données
YY_CONF	<0.1%	N/A
XX_INIT : pré-calculs pour optimisation	<1%	2,51
20_DOC : « décodage des spectres »	**1%**	6,38
09_PLK : « calcul de la fonction de Planck calculée à bord »	<0.1%	0,04
33_SME : « estimation de la température de balayage »	<0.1%	0,00
31_SCR : « calibration spectrale du coefficient de calibration radiométrique »	<1%	3,44
32_HEC : « correction d'émissivité du BB	<1%	0,89
34_SMC : « correction angulaire de réflectivité et de polarisation du miroir de balayage »	<1%	1,91
44_GEO : « Géolocalisation des produits IASI	<0.1%	0,00
45_QIS : « Calcul des indices et flags de qualité	<0.1%	0,00
22_SOS : « suréchantillonnage du spectre »	**8%**	34,79
35_S1B : « re-échantillonnage du spectre	**23%**	101,11
37_S1C : « apodisation du spectre »	**30%**	132,11
40_IAC « co-registration imageur/AVHRR »	**8%**	33,21
41_CCS « analyse des radiances dans les FOV sondeur »	**20%**	88,47
TOTAL chaîne PRODUIT	**93%**	**404**

Tableau 3 : dimensionnement de la chaîne PRODUIT

La performance de la chaîne Produit dépend des algorithmes :

décodage des spectres,

sur échantillonnage du spectre (FFT),

re-échantillonnage du spectre (spline cubique),

apodisation du spectre (convolution par FFT),

co-registration imageur/AVHRR (corrélation),

analyse des radiances dans les FOV sondeur (classification/analyse).

La consommation CPU des chaînes Image et Monitoring est, quant à elle, négligeable hormis l'opération de codage des spectres (12 Mflop).

Ces résultats tiennent compte des optimisations spécifiées et ils montrent que les performances sont en grande partie influencées par deux traitements mathématiques coûteux en temps de calcul : **la transformée de Fourier** et **l'interpolation par spline cubique**.

3.4.2.2.Dimensionnement RAM

L'utilisation de la RAM repose sur un des choix d'implémentation qui sont issus de le phase de conception préliminaire (voir le § 4.2.3.3.5 du [DA108]) :

- le premier paramètre prépondérant qui influence le résultat est le choix de charger en mémoire globalement la banque spectrale.
- Par ailleurs, la décision de charger l'ensemble des données utilisées en entrée/sortie au début de chaque granule est également un facteur dimensionnement au niveau de l'occupation mémoire.

Ces choix nous semblent judicieux et permettent d'aboutir à un dimensionnement de l'ordre de **800 Mo** qui est détaillé dans le tableau ci-dessous en prenant en compte une marge de 30 % pour les données dynamiques. Ceci permet même d'envisager de stocker la banque spectrale en réels double précision.

	taille unitaire	codage	nombre	taille globale
Banque spectrale	330 522 062	1	1	330 522 062
Fichier de configuration stable	3 343 786	1	1	3 343 786
Fichier de configuration autre	120 270	1	1	120 270
contexte	0	1	1	0
				0
				0
données de niveau 0				0
paquets spectres				0
entete	160	2	2640	844 800
spectre codé	8 960	1	2640	23 654 400
spectre décodé	8 500	8	480	32 640 000
paquets images				0
entete	25	2	748	37 400
images codées	6 202	1	748	4 639 096
images décodées	1 920	8	480	7 372 800
paquets de verification				0
MDR	27 000	2	110	5 940 000
paquets auxiliaires				0
MDR	390	2	22	17 160
				0
données internes				22 543 697
				0
données de niveau 1				0
Ligne N1C				0
données toutes les 10 lignes	625 178	1	2,2	1 375 392
entete	25 753	1	22	566 566
spectre 1A decodé	1 020 000	8	4	32 640 000
spectre 1A codé	1 020 000	2	22	44 880 000
spectre 1B décodé	1 020 000	8	4	32 640 000
spectre 1B codé	1 020 000	2	22	44 880 000
spectre 1C décodé	1 020 000	8	4	32 640 000
spectre 1C codé	1 020 000	2	22	44 880 000
Ligne N1 technologiques	6 795	8	22	1 195 920
Ligne N1 vérification	27 000	2	110	5 940 000
données internes (images calibrées ...)				72 491 363
				0
données AVHRR de niveau 2				0
Lignes	10 240	2	1216	24 903 680
données internes				7 471 104
			volume total=	778 179 496

Figure 4 : taille RAM estimée (cas 0-1C)

Cette solution semble optimum pour minimiser les accès aux données en cours de traitement et La taille de la RAM disponible (2 Go) est suffisant sans utiliser le mécanisme de swapping néfaste aux performances. En outre, dans le cas de la banque spectrale, si les données sont correctement rangées en mémoire en stockant de manière contiguë les informations pour un point de la grille 11*11 (2,7 Mo de donnée), les déplacements dans ce grand tableau seront optimisés et utiliseront les performances de la mémoire cache.

3.4.2.3.Dimensionnement I/O

Concernant les entrées/sortie de la chaîne de traitement, l'estimation faite dans le document [DA3] suppose une lecture unique de la banque spectrale en début de dump. Les points les plus importants qui ressortent de cette étude sont :

les volumes liés a la lecture des produits en entrée (1,5+1,3 Mbit par seconde soit 0.35 Mo par seconde de donnée),

les volumes liés a la lecture des de la banque spectrale (666 Mo par dump (6095 s) soit 0.11 Mo par seconde de donnée),

les volumes liés a l'écriture des produits en sortie (3.6 Mbits par seconde soit 0.475 Mo par seconde de donnée).

Le débit est donc de l'ordre de **1 Mo** par seconde, réparti de manière non uniforme sur la durée du traitement. Ceci correspond à un besoin et doit etre comparé avec les performances actuelles d'accès aux disques (de l'ordre de 10 à 20 Mo/s).

4.DEMARCHE PROPOSEE

4.1.STRATEGIE CHOISIE

La démarche d'optimisation et de suivi des performances proposée repose sur les principes suivants :

les performances doivent être prises en compte dans la globalité du cycle de développement du logiciel,

des règles de bases sont définies à chaque étape et les résultats sont quantifiés,

l'utilisation maximale des produits constructeurs qui sont optimisés pour la configuration matérielle cible (compilateur, librairies).

L'architecture proposée par THALES IS met en œuvre ces principes de la manière suivante :

dès la phase de ***Spécifications***, un prototype est conçu et développé afin de rendre indépendant le composant traitement du reste du logiciel IASI. Ceci est possible grâce au haut niveau de réutilisation prévu (logiciel SSALTO),

en phase de ***Conception Préliminaire***, ce prototype, mettant en œuvre les principes de multi-threading est testé grâce à des muets simulant la consommation des ressources par les algorithmes. On peut ainsi évaluer l'impact de l'architecture parallèle sur les temps de traitement (ce prototype est intégré à la version V0). Des choix majeurs d'architecture sont également faits afin d'optimiser les accès disques et mémoire (banque spectrale, AVHRR, lecture/écriture produits ...),

en phase de ***Conception Détaillée***, des tests sont menés sur la configuration cible (noeud quadriprocesseur) afin de valider les performances de certains algorithme de traitement :

FFT complexe / complexe avec ESSL,

FFT réel / complexe et complexe / réel avec ESSL,

spline avec optimisation avec ESSL (point régulièrement espacés, points classés par ordre croissant), On vérifiera ici en outre que dans le cas ou la banque spectrale est stockée en simple précision, la conversion nécessaire en entrée de 43_ISF n'entraîne pas de dégradation majeure de performances.

dans chaque cas, les performances de la bibliothèque sont validées et comparées avec les résultats CNES. En particulier pour les interpolations par spline, on vérifie que les performances sont au moins aussi bonnes que celles évaluées (avec optimisation). Dans le cas contraire le codage d'une fonction spécifique pourra être envisagée. Durant cette phase, les algorithmes critiques sont identifiés pour anticiper un suivi particulier en TU,

en phase de ***Codage*** des algorithmes, les performances sont évaluées pour les fonctions critiques au moment des tests afin de vérifier leur cohérence avec les résultats attendus (en terme de vitesse). Si de gros écarts sont constatés sans possibilité d'amélioration, l'impact est si besoin reporté sur le prototype en modifiant les caractéristiques des muets. On peut ainsi vérifier que

l'objectif global de performances est tenu. Notons également que chaque algorithme au sens du document [DA7] est confié à un même développeur qui a en charge sa mise en œuvre et qui est responsable des performances,

la phase d'***Intégration / Validation*** permet d'assembler progressivement les chaînes de traitement en remplaçant les muets par les algorithmes. Le prototype, une fois complet, permet de valider numériquement les résultats et les performances globales.

4.2.OUTILS UTILISES

Les outils utilisés dans le domaine de l'optimisation sont de deux types :

- Les outils de profiling qui permettent d'analyser le temps et les ressources utilisés par le logiciel (commandes unix (ps,monitor) et utilitaires xprofiler gprof et timex). Pour avoir des résultats plus contextuels on peut également inclure dans le code des fonctions mesurant les ressources (par exemple dans le mode débug).
- Les outils d'optimisation. Dans notre cas, ceci se base sur une bonne connaissance du compilateur C et de ses options.

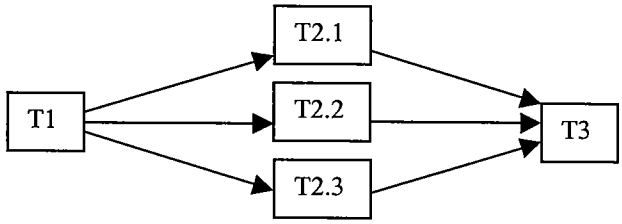
5.LE SUIVI DES PERFORMANCES PHASE PAR PHASE

5.1.PHASE DE SPECIFICATION / CONCEPTION PRELIMINAIRE

5.1.1.Présentation du Prototype

L'architecture du prototype réutilise l'architecture du processus multithreadé SD de SSALTO.

L'architecture multi-threadée mise en place est basée sur la notion de rendez-vous avec un nombre de thread de calcul paramétrables en plus de 2 threads dédiés à la gestion : un thread de communication avec l'extérieur, et un thread de monitoring. La figure suivante présente le type de diagramme de traitements pouvant être exécuté dans cette architecture.



T1 T2 T3 représentent ici des taches de nature différentes

T2.1 T2.2 T2.3 représentent des taches identiques exécutées en parallèle sur des données différentes.

C'est ce que l'on appelle le parallélisme de données.

Le nombre de threads est un paramètre configurable qui sert à l'initialisation au lancement du prototype. Durant la phase d'initialisation, le prototype charge la configuration (diagramme de traitements), met en place les threads, exécute des traitements généraux (Simulation du chargement de la banque spectrale,) puis se met en attente. Via un mécanisme de boite aux lettres, le traitement est activé en précisant la chaîne à simuler (0-1C) et le nombre de lignes d'un granule.

Le diagramme des traitements de la chaîne en terme de tâches est décrit dans un fichier de configuration chargé au lancement. A l'aide d'une syntaxe particulière, il est possible d'indiquer les tâches de type rendez-vous (RDV) et les taches parallélisables (LIGNE). Un exemple de diagramme de tâches est présenté ci-dessous.

Ex :

```
[0-1C]
NOMBRE TACHES=6
TACHE1=LECTURE_DONNEES
```

```
TYPE TACHE1=RDV
TACHE2=TACHE_ALGO_1
TYPE TACHE2=RDV
TACHE3=TACHE_ALGO_2
TYPE TACHE3=LIGNE
TACHE4=TACHE_ALGO_3
TYPE TACHE4=RDV
TACHE5=TACHE_ALGO_8
TYPE TACHE5=LIGNE
TACHE6=ECRITURE_DONNEES
TYPE TACHE6=RDV
```

La définition de chaque tache est disponible dans un autre fichier de configuration dans lequel le contexte associé à chaque tache est décrit (numéro de ligne, le paramètre de simulation, ...).

Un exemple de définition des tâches est présenté ci-dessous.

Cette approche ainsi que l'utilisation de fichiers de configuration amène une grande souplesse à l'architecture proposée qui peut être reconfigurer de manière très rapide.

Ex :

```
[LECTURE_DONNEES]
OCCURRENCE=Aucune
SIMULATION=-1
NOMBRE ACTIONS=1
ACTION1=LectureDonnees

[ECRITURE_DONNEES]
OCCURRENCE=Aucune
SIMULATION=-1
NOMBRE ACTIONS=1
ACTION1=EcritureDonnees

[TACHE_ALGO_1]
OCCURRENCE=Aucune
SIMULATION=-2
NOMBRE ACTIONS=1
ACTION1=Algo_00_INI

[TACHE_ALGO_2]
OCCURRENCE=Ligne
```

SIMULATION=-12
NOMBRE ACTIONS=8
ACTION1=Algo_38_ICC
ACTION2=Algo_110_DPT
ACTION3=Algo_39_IRC
ACTION4=Algo_20_DOC
ACTION5=Algo_22_SOS
ACTION6=Algo_23_SSD
ACTION7=Algo_21_SSS
ACTION8=Algo_24_IAX

[TACHE_ALGO_3]
OCCURRENCE=Aucune
SIMULATION=0
NOMBRE ACTIONS=1
ACTION1=Algo_25_FAX

[TACHE_ALGO_8]
OCCURRENCE=Ligne
SIMULATION=3
NOMBRE ACTIONS=16
ACTION1=Algo_43_ISF
ACTION2=Algo_09_PLK
ACTION3=Algo_33_SME
ACTION4=Algo_20_DOC
ACTION5=Algo_100_EXS
ACTION6=Algo_111_MCX
ACTION7=Algo_31_SCR
ACTION8=Algo_32_HEC
ACTION9=Algo_34_SMC
ACTION10=Algo_22_SOS
ACTION11=Algo_35_S1B
ACTION12=Algo_37_S1C
ACTION13=Algo_40_IAC
ACTION14=Algo_41_CCS
ACTION15=Algo_44_GEO
ACTION16=Algo_45_QIS

5.1.2.Hypothèses des tests réalisés avec le prototype

Les hypothèses retenues pour effectuer les tests avec le prototype sont les suivantes :

un granule de 22 lignes (176 s de données),

traitement complet (0-1C),

parallélisation les boucles 1 (ISRFEM) et 7(produit) de l'annexe 2 du document [DA2].

Le prototype réalise la simulation de la chaîne nominale (0-1C).

Cette chaîne se décompose ainsi :

Numéro tache	Parallelisation	Contenu	Synchronisation
1	Non	lecture données granules	
2	Non	algorithme 00_INI	attente fin 0
3	par ligne	algorithmes 38_ICC,110_DPT, 39_IRC, 20_DOC, 22,SOS, 23_SSD, 21_SSS, 24_IAX	attente fin 1
4	Non	algorithme 25_FAX	attente fin 2
5	par ligne	algorithmes 43_ISF, 09_PLK, 33_SME, 100_EXS, 111_MCX, 31_SCR, 32_HEC, 34_SMC, 22_SOS, 35_S1B, 37_S1C, 49_IAC, 41_CCS, 44_GEO, 45_QIS	attente fin 3
6	Non	écriture données granules	attente fin 4

En plus des traitements algorithmiques, les phases de lecture/écriture des produits sont simulées avec une taille mémoire allouée correspondant approximativement à la banque spectrale plus celle des produits :

Type	Nature	Volume en Mo
I/O	Lecture données	40
	Ecriture données	86
RAM 900Mo au total	Banque spectrale	683
	Simulation tâche	235

La RAM est allouée au lancement du prototype ; la mémoire « Simulation tâche » est accédée et mise à jour au niveau des taches algorithmiques. Ceci permet de simuler des accès mémoire réalistes afin d'éviter de rester dans la mémoire cache du processeur.

5.1.3.Configuration matérielle utilisée

La configuration matérielle de simulation est la suivante :

machine hôte : IBM quadri-processeur 4*375 Mhz ; RAM 4 Go

machine réservée pour l'application OPS .

options de compilation du prototype : **-O –DPROTO -D_REENTRANT**

option d'édition de liens : **–bmaxdata :1800000000**

5.1.4.Démarche

Notre démarche a pour but de nous mettre dans des conditions d'exécution proche de celles constatées pour la maquette de l'OPS (cf [DR4]). Apres nous être donc étalonné par rapport à cette référence, nous avons évalué l'effet d'une parallélisation par ligne sur la chaîne OPS en utilisant l'architecture multithreadé envisagée. Les étapes de ce processus sont les suivantes :

1/ Analyse d'un benchmark simple (avec compilation optimisée (-O))

La vitesse de la machine utilisée est estimée à partir d'un benchmark de corrélation entre 2 images volumineuses :
Nb op : 815 Mop
Temps mesuré (ellapsed) : 2,51 s
Vitesse estimée = 324 Mflop/s

2/ Comparaison avec la vitesse réelle de la maquette OPS (cf [DA3]):

- Maquette OPS 435*80 Mflop en 81 s Vr = 425 Mflop/s
- Benchmark Vb= 321 Mflop/s

3/ Extrapolation des nombres d'opérations : on applique à chaque algorithme dimensionné dans le document [DA3] le facteur **Vb/Vr** (0,756) pour avoir un temps d'exécution comparable à la maquette OPS.

On obtient ainsi un nombre d'opération équivalent simulant avec notre benchmark les temps d'exécution de l'OPS. Ces nombres d'opération sont directement utilisés comme paramètres du prototype :

Tâche	Plate-forme CNES (Mflop)	Plate-forme TIS (Mflop)
TACHE_ALGO_2/granule	441,32	334
TACHE_ALGO_3/ligne	84,32	64
TACHE_ALGO_4/granule	0,16	0.12
TACHE_ALGO_5/ligne	3205	2424
TOTAL	**72799**	**55070**

Tableau 5 : Nombre d'opérations équivalent bench OPS

5.1.5.Résultats

La stratégie précédente permet alors d'analyser le comportement du prototype en fonction du degré de parallélisation. Les résultats ci-dessous (en secondes) prennent également en compte les E/S qui ne sont pas parallélisées et qui représentent 5s.

Configuration	Elapsed time	User time	System time
1 thread de calcul 22 lignes	181.25	175.20	1.43
2 threads de calcul 22 lignes	93.49 (t1/1,94)	175.62	1.20
3 threads de calcul 22 lignes	69.54 (t1/2.6)	175.79	1.40
4 threads de calcul 22 lignes	54.55 (t1/3,35)	176.26	1.28
5 threads de calcul 22 lignes	52.60 (t1/3,48)	175.95	1.39
6 threads de calcul 22 lignes	52.5	175.61	1.24
7 threads de calcul 22 lignes	51.11	175.84	1.31
8 threads de calcul 22 lignes	50.11	174.72	1.31
9 threads de calcul 22 lignes	50.91	174.33	1.33
10 threads de calcul 22 lignes	53.02	174.22	1.31

Configuration	Elapsed time	User time	System time
11 threads de calcul 22 lignes	50.06	173.77	1.40
12 threads de calcul 22 lignes	49.87	173.67	1.35

Tableau 6 : Résultats des Simulations en Phase de Spécifications

Ces premiers résultats montrent que :

Pour un seul thread de calcul, les résultats sont comparables aux résultats obtenus par le CNES (178s pour 22 lignes), ce qui valide notre démarche et montre que l'architecture multithreadé induit très peu de pertes en terme de performances,

le gain de performances lié à la parallélisation des calculs est élevé : le ratio est de 3,4 pour 4 threads de calcul.

Ces premiers résultats semblent démontrer que l'architecture proposée permettra d'atteindre les exigences de performance.

5.2.PHASE DE CONCEPTION DETAILLEE

Au cours de cette phase, nous nous sommes attaché à vérifier les performances des librairies utilisées sur une configuration proche de la configuration cible. Nous avons travaillé ici dans un contexte mono-thread afin de mettre à jour l'approche théorique proposée dans le document [DA3]. Les tests sont decomposés en deux parties principales :

- Test de la librairies ESSL au travers des fonctions dimensionnantes que sont :
    - L 'interpolation spline cubique,
    - La transformée de Fourier directe (réels vers complexe) et inverse (complexe vers réels).
- Test des librairies Metop (metop_lib, metop_pointing, metop_orbit). On s'attache ici à estimer les temps de calcul pour vérifier les risques de blocage dus au caractère non thread-safe de ces librairies. On vérifie en particulier :
    - La fonction de conversion des coordonnées Metop vers les coordonnées géodésiques,
    - La fonction de conversion des coordonnées géodésiques vers les coordonnées Metop.

Dans chacun des cas, les tests on été codés de manière réaliste en implémentant les fonctions concernées (SD_BAS_LibESSL et SD_BAS_OperateurLoc) dans le logiciel.

Notre approche dans ce qui suit consiste à définir et à analyser dans chacun des tests les points suivants:
1. Les fonctions utilisées,
2. Les algorithmes OPS utilisateurs,
3. Les entrées/sorties choisies,
4. Les conditions réelles d'activation ,
5. la séquence d'activation ,
6. la fréquence d'activation ,
7. Les temps de calcul.

Des comparaisons sont ensuite effectuées avec les résultats théorique afin de valider les performances

5.2.1.Librairie ESSL

Les fonctions à tester dans la librairie ESSL sont issues du document [DA3] :
- Transformée de fourrier directe (réel vers complexe),
- Transformée de fourrier Inverse (complexe vers réel),
- Interpolation par spline cubique.

5.2.1.1.interpolation spline cubique

5.2.1.1.1.Cas d'utilisation 1

Fonction ESSL	*DCSINT*
Algorithmes utilisateurs	SSD
Entrées utilisées	1400 pts quelconques
Condition d'activation réelles	5 itérations sans possibilité d'initialisation des coefficients
Sorties	1400 pts
Séquence d'activation	5 appels
Fréquence d'activation	Par ligne : PN SN
Temps à mesurer	Temps de 120 appels
Nb operations théoriques (cf DA3)	5 *88200*120= 52,9 Mflop
Temps CPU mesuré (s)	0.19
Vitesse évaluée	278

5.2.1.1.2.Cas d'utilisation 2

Fonction ESSL	*DCSINT*
Algorithmes utilisateurs	S1B
Entrées utilisées	42500 pts quelconques
Condition d'activation réelles	possibilité d'initialisation des coefficients par ligne
Sorties	8500 pts
Séquence d'activation	1 appels
Fréquence d'activation	Par ligne : PN SN
Temps à mesurer	Temps sur 4*30 appels
Nb operations théoriques (cf DA3)	6740500*120= 808 Mflop
Temps CPU mesuré (s)	0.16
Vitesse évaluée	Non significative du fait des optimisations (5000)

5.2.1.1.3.Conclusion

Les résultats obtenus montrent un bon niveau de performances de la librairie ESSL qui se rapproche de la puissance optimum visée [DA3] (420Mflop). On utilise ici si possible les optimisations proposées afin de précalculer les coefficients d'interpolation et donc de dépasser largement ces performances.

5.2.1.2.transformée de fourrier directe réels vers complexe et inverse complexe vers réels

5.2.1.2.1.Cas d'utilisation 1

Fonction ESSL	*DRCFT*
Algorithmes utilisateurs	SSD
Entrées utilisées	1400 pts quelconques complétées par des 0 jusqu'a 1408 ou 2048 (zero padding)
Condition d'activation réelles	5 itérations avec possibilité d'initialisation des coefficients
Sorties	1400 pts
Séquence d'activation	5 appels
Fréquence d'activation	Par ligne : PN SN
Temps à mesurer	Temps de 120 appels
Nb operations théoriques (cf DA3)	5 *33800*120= 20,3 Mflop
Temps CPU mesuré (s)	0.02
Vitesse évaluée (Mflops)	Non significative du fait des optimisations (1515)

5.2.1.2.2.Cas d'utilisation 2

Fonction ESSL	*DRCFT + DCRFT*
Algorithmes utilisateurs	SOS (produit)
Entrées utilisées	16384 pts (16 k)
Condition d'activation réelles	6 appels avec possibilité d'initialisation des coefficients
Sorties	16384 pts (16 k)
Séquence d'activation	6 appels (1 direct + 5 inverse)
Fréquence d'activation	Par ligne : PN SN
Temps à mesurer	Temps de 120 appels
Nb operations théoriques (cf DA3)	6*344000*120= 247 Mflop
Temps CPU mesuré (s)	0.65
Vitesse évaluée (Mflops)	380

5.2.1.2.3.Cas d'utilisation 3

Fonction ESSL	*DRCFT + DCRFT*
Algorithmes utilisateurs	S1C (produit)
Entrées utilisées	1024 pts
Condition d'activation réelles	141 appels (FFT + FFT^{-1}) avec possibilité d'initialisation des coefficients
Sorties	1024 pts
Séquence d'activation	141 appels (FFT + FFT^{-1})
Fréquence d'activation	Par ligne : PN SN
Temps à mesurer	Temps de 120 appels
Nb operations théoriques (cf DA3)	2*26672*141*120= 902 Mflop
Temps CPU mesuré (s)	1.24
Vitesse évaluée (Mflops)	Non significative du fait des optimisations (720)

5.2.1.2.4.Conclusion

Les résultats obtenus sont cohérents par rapport aux vitesses constatées dans le document [DA3] (340Mflop) dans un cas comparable (cas 2) . Dans les autres cas, les précalculs de coefficients permettent de dépasser ces objectifs.

5.2.2.Librairies Metop

Les fonctions à tester dans les librairies Metop sont contenus dans la classe SD_BAS_OperateurLocalisation.

Ces fonctions sont testées à priori non pas pour leur coût important en terme de calcul mais à cause du fait qu'elles ne sont pas réentrantes. Il faut donc envisager une solution de verrou rendant leur utilisation exclusive. Dans ce contexte nous allons évaluer pour chacune :

- La séquence d'appel nécessaire,
- La fréquence d'appel de cette séquence au sein de la chaîne parallélisée (/ligne),
- Les temps calculs par PN/ligne/globale. On pourra alors estimer le temps maximum de blocage (avec 4 thread) par la formule : tglob – (tglob/4),

5.2.2.1.Conversion Metop vers Geod

5.2.2.1.1.Cas d'utilisation 1

Fonction Metop	*Mp_target*
Algorithmes utilisateurs	41_CCS et 40_IAC
Entrées utilisées	Issus du code de test metop_lib avec les options choisies par le CNES
Condition d'activation réelles	metop_orbit (2 fois) + mp_target (n points)
Sorties	
Séquence d'activation	1 appel pour 4 points ou 1 appel pour 16 points
Fréquence d'activation	Par ligne : SN
Temps mesuré par SN (s)	négligeable
Temps mesuré par thread (ligne)	0.05 0.11
Temps global (s)	1.1(0.82)
(temps de blocage maximum 4 thread)	2.42 (1.81)

5.2.2.1.2.Cas d'utilisation 2

Fonction Metop	*Mp_target*
Algorithmes utilisateurs	44_GEO
Entrées utilisées	Issus du code de test metoplib avec les options choisies par le CNES
Condition d'activation réelles	Metop_orbit (2 fois) + mp_target (n points)
Sorties	
Séquence d'activation	1 appel pour (5*5+4) points
Fréquence d'activation	Par ligne : SN
Temps mesuré par SN (s)	négligeable
Temps mesuré par thread (ligne) (s)	0.17
Temps global (s)	3.74 (2.8)
(temps de blocage maximum 4 thread)	

5.2.2.1.3.Conclusion

Le blocage potentiel engendré par ces fonctions ne nécessite pas une optimisation de l'algorithmie.

5.2.2.2.Conversion Metop vers Geod

5.2.2.2.1.Cas d'utilisation 1

Fonction Metop	*Mp_stavis*
Algorithmes utilisateurs	41_CCS
Entrées utilisées	Issus du code de test metoplib avec les options choisies par le CNES. 4*20*37 points (cas extrême) 4*12*12 points (cas minimum)
Condition d'activation réelles	metop_orbit (2 fois) + mp_stavis (n points)
Sorties	
Séquence d'activation	1 appel pour 4*20*37 points 1 appel pour 4*12*12 points
Fréquence d'activation	Par ligne : SN
Temps mesuré par SN (s)	0,27 0.05
Temps mesuré par thread (ligne) (s)	8.2 1.6
Temps global (s) **(temps de blocage maximum 4 threads)**	180 (135) 35(26)

5.2.2.2.2.Cas d'utilisation 2

Fonction Metop	*Mp_stavis*
Algorithmes utilisateurs	40_IAC
Entrées utilisées	Issus du code de test metoplib avec les options choisies par le CNES. 87*154 points (cas extrême) 50*50 points (cas minimum)
Condition d'activation réelles	metop_orbit (2 fois) + mp_stavis (n points)
Sorties	
Séquence d'activation	1 appel pour 87*154 points 1 appel pour 50*50 points
Fréquence d'activation	Par ligne : PN SN
Temps mesuré par SN (s)	1.2 0.22
Temps mesuré par thread (ligne) (s)	36 6.8
Temps global (s) **(temps de blocage maximum 4 threads)**	792 (594) 150 (112)

5.2.2.2.3.Conclusion

On constate ici des temps de calcul importants (entre 3 et 17 mn) et et donc un risque majeur de ralentissement de la chaîne (entre 2 et 12 mn) si des blocages surviennent du fait du mécanisme de verrou.

Le blocage potentiel engendré par ces fonctions nécessite une optimisation particulière de l'algorithmie. Le problème vient en fait du temps de calcul intrinsèque qui est incompatible avec les objectifs de performances (22 lignes doivent être traitées en moins de 3 mn !). Plusieurs éventualités sont possibles :

1. Sous échantillonnage dans les algorithmes IAC et CCS.,

2. Modification des parametres d'utilisation des librairies. Actuellement, les paramètres choisis sont les suivants :

```
iatt=3;
aocs_g[0]= -0.1660;
aocs_g[1]= 0.0508;
aocs_g[2]= 3.9570;
att[0] = MphrN0->pitch_error;
att[1] = MphrN0->roll_error;
att[2] = MphrN0->yaw_error;
datt[0] = 0.0;          /* MetOp-1 SRAR mispointing rates [deg/s] */
datt[1] = 0.0;
datt[2] = 0.0;
 sta[0] = lon[i];        /* Ground station geocentric longitude [deg] */
 sta[1] = lat[i];        /* Ground station geodetic latitude [deg] */
 sta[2] = 0.0;           /* Ground station geodetic altitude [m] */
 sta[3] = -90.;          /* Ground station minimum elevation link [deg] */
```

3. Recodage des librairies metop.

5.3.PHASE DE CODAGE

Lors de la phase de codage, l'OPS a été soumis à un certains nombre de tests ayant pour but de valider les performances élémentaires des algorithmes. Ces tests on été menés sur la machine de développement décrite ci-dessous.

5.3.1.Configuration matérielle utilisée :

>IBM Power 3 bi-processeur 440 Mhz,

>1 Go de RAM,

>Optimisations standards (O2)

5.3.2.Hypothèses :

>Les fonctions testées ont été appelées au travers des fonctions réelles du paquetage SD ALG / SD_BAS en utilisant les lanceurs de TU.

>L'utilisation prend en compte la mise en œuvre réelle sur l'OPS **sans appel aux librairies Metop**. De plus du fait de la spécification tardive de l'algorithme 41_CCS, celui-ci n'a pas fait l'objet de cette analyse.

5.3.3.Démarche utilisée :

La démarche utilisée dans le cadre de l'analyse des performances est la suivante :

1 Codage des TU des algorithmes

2 Tests unitaires informatiques .3 Tests unitaires de performance spécifiques pour les fonctions en mesurant les performances pour plusieurs appels consécutifs dans des conditions réalistes.

4 Calcul du nombre d'opérations théorique en se basant sur le document [DA3]

5 Estimation d'une performance réelle de chaque algorithme et analyse des écarts par rapport à la vitesse recherchée (340 Mflops en moyenne) et à la vitesse de la machine.

5.3.4.Résultats :

ALGORITHMES CRITIQUES	NOMBRE D'OPERATIONS THEORIQUES	NOMBRE D'ITERATIONS	TEMPS CPU (S)	VITESSE ESTIMEE (MFLOPS)
22_SOS (ISRFEM)	37240 (CNES) 65800 (THALES)	1000	0,461	142
23_SSD	620000	1000	1,583	391
43_ISF	13800000	1000	40,65	339
40_IAC (HORS METOP LIB)	8856174	1000	18,75	472
35_S1B	206107	1000	1,57	130
37_S1C	9114958	1000	39,4	231

5.3.5.Analyse

Les points suivants peuvent être mis en évidence en fonction des résultats précédents :

- Pour 22_SOS, des calculs négligés dans le document [DA3]s'avèrent déterminants,
- Des optimisations sont en cours dans le cas de la chaîne produit, les résultats de ces optimisations seront évalués en validation.

Les performances visées (~ 400 Mflops) ne sont pas toujours atteintes mais la configuration cible ayant de bien meilleures performances que prévue, cela relativise le risque.

L'utilisation des ressources par les fonctions Metop (temps non parallélisable) reste ici un risque majeur qui doit être surveillé.

5.4.PHASE DE VALIDATION

Lors de la phase de validation, des tests de validation ayant pour but de valider les performances globales de la chaîne ont été mis en oeuvre. Ces tests on été menés sur la machine de validation décrite ci-dessous.

5.4.1.Configuration matérielle utilisée :

>IBM Power 4 quadri-processeur 1 Ghz,

>2 Go de RAM,

>Optimisations standards (O2)

5.4.2.Hypothèses :

- Le test de validation utilisé pour ces mesures est le test GRAN_OPS_N_16 qui met en œuvre le jeu de donnée J0 composé de 41 granules de 3 minutes de données (total 120 minutes),
- Le nombre de thread de calcul est 4,
- Les temps relevés sont des temps écoulés après mise à disposition des données,
- Le fichier de configuration utilisé spécifie une fenêtre de recherche de +-20 pixels,
- L'algorithme 41_CCS du fait des données de localisation non correctes n'a pas été pris en compte dans le chaîne (sa tolérance aux erreurs de localisation semblant faible).

5.4.3.Démarche utilisée :

L'utilisation prend en compte la mise en œuvre réelle sur l'OPS en partant d'une initialisation **avec appel aux librairies Metop**. De plus du fait de nombreux problèmes liés à la localisation des données, les algorithmes de la chaîne AVHRR sont en partie non pris en compte dans les résultats de ce test.

Les temps relevés sont des temps écoulés qui peuvent différer d'un granule à l'autre. Le premier granule est utilisé dans l'OPS pour l'initialisation des données statiques, son temps n'est donc pas représentatif.

5.4.4.Résultats :

Numero du granule	Temps écoulé pour le traitement (s)
Granule 1	198
Granule 2	199
Granule 3	203
Granule 2	199
Granule 23	200
Granule 24	202
Granule 39	206
Granule 40	213
Granule 41	200

5.4.5.Analyse

On constate ici un temps identique au niveau du granule numéro 1 ce qui est étonnant au vue de la phase d'initialisation menée . Chaque granule contient en moyenne 22 lignes (176s) , ce qui met en évidence le dépassement des performances visées (176s) qui est pour nous du à :

- La corrélation (40_IAC) qui utilise une taille de fenêtre de recherche 2 fois plus importante que prévue)
- Les fonction métop qui sont utilisées en mode non parallèle et qui même avec un sous échantillonnage ne semble pas négligeables en temps d'exécution.

Une analyse plus poussée ne nous semble pas à ce stade nécessaire tant que des données réalistes en terme de localisation ne seront pas utilisées. En effet ceci est indispensable pour avoir des résultats réalistes. Cependant une exécution utilisant un outil de profiling pourrait être menée pour analyser les temps passés dans les différents algorithmes. Il reste en outre à évaluer 41_CCS sur des données réalistes

Remarque : une amélioration notable des performances peut être obtenue en utilisant une parallélisation sur 5 threads.

5.4.6.Résultats en mettant en œuvre la chaine AVHRR avant validation

scientifique (V2.0):

Les test précédents on donc été repris après mise au point des algorithmes de la chaîne AVHRR dans l'optique de la validation scientifique. Les librairies Metop sont donc ici utilisées de manière nominale dans le contexte du jeu J0.

Les premiers tests on montrés des temps de calcul très supérieurs aux objectifs initiaux ce qui a entraîné une démarche d'optimisation sur l'ensemble de la chaîne. Cette démarche s'est appuyée sur l'utilisation de l'outil de profiling gprof qui permet avec des options de compilation optimisées d'estimer le temps passé dans les différentes fonctions. Nous présentons ci-dessous les résultats obtenus au départ en se basant sur une exécution sur 2 granules (granules 23 et 24). On visualise ainsi les fonctions les plus coûteuses en temps dans la chaîne OPS. Une analyse plus fine a alors été menée

1. sortie de gprof avant optimisation

% time	cumulative seconds	self seconds	self calls	total ms/call	ms/call	name
20.7	264.71	264.71	5280	50.13	50.32	SD_ALG_TransformIpsf_CCS
7.3	357.71	93.00	15840	5.87	5.87 .	SD_BAS_GeodAVHRR [13]
6.1	435.67	77.96	4023980	0.02	0.02 .	SD_ALG_Moyenne [15]
5.9	510.74	75.07	10560	7.11	7.68 .	SD_ALG_22_SOS [14]
3.3	552.44	41.70	1320	31.59	39.95 .	SD_ALG_40_IAC [16]
3.2	593.10	40.66				mcount [19]
3.0	632.16	39.06	352	110.97	114.69	.SD_ALG_TransformIpsf [20]
2.5	664.74	32.58				.dwdn$ [21]
2.4	695.61	30.87				dftp2$ [23]
2.0	721.31	25.70				df8di$ [24]
1.9	746.19	24.88	10560	2.36	2.36	.SD_ALG_20_DOC [25]
1.9	770.97	24.78				dbsrch [26]
1.8	794.03	23.06	70897	0.33	0.33 .	SD_ALG_DistPixProto [27]
1.8	816.88	22.85	73167	0.31	1.38 .	SD_ALG_StatClas [10]

Cette approche nous a permis de nous focaliser sur les points majeurs suivants :

- utilisation d'option de compilation plus efficaces (-O3 -qstrict -qarch=pwr3 -qtune=pwr3)
- transform ipsf (optimisation du code dans 41_CCS et SSD, optimisation des paramètres de configuration (OSFFactor)
- optimisation des options de compilation
- optimisation de la fonction d'initialisation des FFT (une initialisation par ligne)
- optimisation de la fonction SD_ALG_Moyenne (41_CCS) par minimisation du nombre d'appels
- optimisation de la recherche dans le bandeau AVHRR (SD_BAS_GeodAVHRR) et optimisation

des paramètres de conf associés . (marges en ligne et en colonnes)

Le résultat obtenu après optimisation est le suivant :

2. sortie de gprof après optimisation

% time	cumulative seconds	self seconds	self calls	total ms/call	ms/call	name
12.7	122.76	122.76	5280	23.25	23.41	.SD_ALG_TransformIpsf_CCS [7]
8.9	209.40	86.64	10560	8.20	8.83	.SD_ALG_22_SOS [8]
4.7	255.00	45.60				_mcount [15]
4.5	299.04	44.04	1320	33.36	36.94	.SD_ALG_40_IAC [14]
3.8	336.20	37.16				dftp2$ [16]
3.3	367.78	31.58				df8di$ [17]
3.0	397.05	29.27	1474908	0.02	0.02	SD_ALG_Moyenne [18]
2.7	423.39	26.34				dbsrch [19]
2.7	449.18	25.79	834171	0.03	0.03	SD_BAS_PreparationFFTC [20]
2.6	474.35	25.17				d8fw$ [21]
2.6	499.27	24.92	73164	0.34	0.74	.SD_ALG_StatClas [12]
2.6	523.98	24.71	10560	2.34	2.34	.SD_ALG_20_DOC [22]
2.5	548.20	24.22				dfcp1$ [23]
2.5	572.11	23.91	70897	0.34	0.34	.SD_ALG_DistPixProto [24]
2.4	595.52	23.41				df16i$ [25]
2.4	618.48	22.96				d32l$ [27]
2.2	639.57	21.09				dcsts$ [28]
1.9	658.28	18.71				.memset [30]
1.7	674.66	16.38				dptf [32]
1.6	690.14	15.48	118211660	0.00	0.00	exp [33]

Dans ce contexte les postes majeurs de la chaîne restent les fonctions .SD_ALG_TransformIpsf_CCS , SD_ALG_22_SOS et SD_ALG_40_IAC qui semblent dans l'état actuel correctement optimisées

Les résultats globaux après optimisation sont les suivants :

Numero du granule	Temps écoulé pour le traitement (s)
Granule 1	171
Granule 2	182
Granule 3	185
Granule 4	181
Granule 23	185

Granule 24	184
Granule 39	174
Granule 40	181
Granule 41	186

En tenant compte du fait que la zone de recherche dans l'algorithme de corrélation (IAC) est de 20 à la place de 10 en nominal (perte estimées à 2 % du temps de la chaine (4 s)) , les temps obtenus satisfont aux exigences de performance (176 s en moyenne pour un granule de 22 lignes) .

5.4.7.Résultats en mettant en œuvre la chaîne AVHRR après validation scientifique (V3.0):

Les tests précédents ont été repris après la phase de validation algorithmique. Il est à noter que certaines modifications concernant notamment les aspects géolocalisation ont été notablement modifiés avec un impact sur les performances.

Des optimisations complémentaires ont donc du être menées pour compenser les pertes constatées

Parmi les modifications faites, on trouve :

- Optimisation du multi-threading lors de l'appel aux librairies Metop,
- Optimisation des paramètres de configuration (OSFFactor est mis actuellement à la valeur 4)

Nous présentons ci-dessous les résultats obtenus après optimisation en se basant sur une exécution sur 2 granules (granules 23 et 24). On visualise ainsi les fonctions les plus coûteuses en temps dans la chaîne OPS.

1. sorties de l'outil gprof après optimisation

granularity: each sample hit covers 4 byte(s) Total time: 902.14 seconds

% time	cumulative seconds	self seconds	self calls	self ms/call	total ms/call	name
11.5	103.80	103.80				.SD_ALG_22_SOS [1]
4.9	148.02	44.22				.dftp2$ [2]
4.4	187.52	39.50				.__mcount [3]
4.1	224.47	36.95				.df8di$ [4]
3.7	257.40	32.93				.SD_ALG_Moyenne [5]
3.4	287.63	30.23				.SD_ALG_TransformIpsf_CCS [6]

3.3	316.97	29.34			.SD_BAS_PreparationFFTC [7]
3.2	345.95	28.98			.SD_ALG_StatClas [9]
3.1	374.24	28.29			.dfcp1$ [10]
3.0	401.61	27.37			.d8fw$ [11]
3.0	428.26	26.65			.dbsrch [12]
2.7	452.78	24.52			.df16i$ [13]
2.7	476.77	23.99			.dcsts$ [14]
2.7	500.68	23.91			.SD_ALG_20_DOC [15]
2.7	524.59	23.91			.SD_ALG_DistPixProto [16]
2.6	547.84	23.25			.d32l$ [17]
2.2	567.45	19.61			.memset [18]
2.0	585.52	18.07	158479599	0.00 0.00	.exp [19]
1.9	602.74	17.22			.SD_BAS_PreparationFFTR [21]
1.7	618.13	15.39			.dptf [22]
1.7	633.34	15.21			.SD_ALG_37_S1C [23]

Dans ce contexte, les postes majeurs de la chaîne sont, outre les fonctions de base (fft…) les fonctions .SD_ALG_TransformIpsf_CCS , SD_ALG_22_SOS et .SD_ALG_Moyenne (41_CCS) qui semblent dans l'état actuel correctement optimisées. Si des améliorations devaient être recherchées, il faudrait analyser plus finement le code de 22_SOS.

Les résultats globaux après optimisation sont les suivants :

Numero du granule	Temps écoulé pour le traitement (s)
Granule 1	197
Granule 2	171
Granule 3	175
Granule 4	170
Granule 23	173
Granule 24	169
Granule 39	170
Granule 40	170
Granule 41	169

Les temps obtenus sur la machine avec 2Go de mémoire satisfont aux exigences de performance (176 s en moyenne pour un granule de 22 lignes) hormis pour le premier granule qui intègre le temps de chargement des données de configuration (en particulier la banque spectrale).