

Rapport de stage MF/CNRM/GMAP

Optimisation dans le code de rayonnement SRTM

Période du 20 Septembre 2010 au 30 Octobre 2010

Par : M. HACHELAF Rabah

Office National de la Météorologie /Algérie

Encadrement : M. Y.Bouteloup

1-Introduction :

Lors du branchement du code SRTM dans ARPEGE, pendant le stage de l'année passée (Branchement du code de rayonnement SRTM dans ARPEGE)

On a constaté que le coût de calcul du modèle ARPEGE avec ce code est devenu beaucoup plus cher, et pour remédier à ce problème il faut chercher quelles sont les routines qui utilisent beaucoup plus de temps de calcul et essayer de les optimiser en les vectorisant.

Pour cela il faut utiliser des packs du modèle ARPEGE en FTRACE (flow trace), ce FTRACE est un profileur permettant d'obtenir pour chaque routine sa fréquence d'appel, le temps de calcul (CPU) moyen passé dans la routine, les MFLOPS, la longueur moyenne des vecteurs lors du calcul et le ratio de vectorisation.

2-Procédure de travail :

Avant d'entamer des modifications pour l'optimisation, il faut cibler les routines à optimiser, par un run du modèle ARPEGE avec SRTM en FTRACE. (Expérience)

Le résultat qui nous intéresse à partir de cette expérience sont les profils FTRACE.

Du moment qu'on a exécuté notre expérience sur 2 processeurs, on va trouver 2 fichiers FTRACE (un pour chaque processeur).

Les profils FTRACE en sorties montre que le modèle ARPEGE avec SRTM passe plus de 32% de temps de calcul totale sur la routine « srtm_reftra.F90 », sur le 1^{er} processeur et 26% sur le second.

```
*****
FTRACE ANALYSIS LIST
*****

Execution Date : Fri Sep 24 09:27:00 2010
Total CPU Time : 0:14'51"856 (891.856 sec.)

PROC.NAME  FREQUENCY  EXCLUSIVE  AVER.TIME  MOPS  MFLOPS  V.OP  AVER.  VECTOR  I-CACHE  O-CACHE  BANK  CONFLICT
           TIME[sec]( % )  [msec]
srtm_reftra  44352  289.879( 32.5)  6.536  13097.1  4750.4  99.04  70.6  280.546  0.064  0.438  9.653  153.411
srtm_vrtqdr  44352  75.710( 8.5)  1.707  16149.1  4773.1  99.89  255.1  75.561  0.034  0.040  8.848  51.358
fpcincapc  8612  53.891( 6.0)  6.258  52068.6  28305.8  99.71  255.5  53.713  0.036  0.058  0.241  7.002
mpl_barrier_mod.mpl_barrier  3683  41.272( 4.6)  11.206  150.6  0.0  7.45  6.6  8.374  0.030  0.037  0.010  7.173
srtm_srtm_224gp  198  35.345( 4.0)  178.508  1819.2  591.3  95.27  17.2  35.329  0.004  0.012  0.018  34.452
srtm_spcvrt  198  33.546( 3.8)  169.423  38971.5  16476.7  99.61  255.1  33.346  0.044  0.088  0.969  10.482
rrtm_rtrnla_140gp  198  30.468( 3.4)  153.877  56673.5  19232.9  99.68  253.3  30.378  0.006  0.014  0.460  13.596
mf_phys  1078  18.627( 2.1)  17.279  14972.5  2468.2  98.60  94.6  12.100  0.096  4.622  0.198  1.388
laitri  11858  17.737( 2.0)  1.496  52949.5  15853.4  99.96  255.1  17.727  0.001  0.001  3.965  6.888
lascaw  3234  14.579( 1.6)  4.508  12646.6  1750.8  99.83  254.6  14.535  0.011  0.013  0.947  10.894
```

Profils FTRACE de l'exécution du modèle ARPEGE+SRTM sur le 1^{er} processeur.

 FTRACE ANALYSIS LIST

Execution Date : Fri Sep 24 09:27:00 2010
 Total CPU Time : 0:18'13"459 (1093.459 sec.)

PROC.NAME	FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	I-CACHE MISS	O-CACHE MISS	BANK CPU PORT	CONFLICT NETWORK
srtm_reftra	44352	288.956(26.4)	6.515	13098.8	4759.5	99.05	70.6	279.572	0.082	0.472	9.575	152.637
fpincscape	27237	177.590(16.2)	6.520	52628.4	28611.8	99.71	255.8	177.142	0.028	0.066	0.795	21.570
srtm_vrtqdr	44352	75.585(6.9)	1.704	16175.8	4781.0	99.89	255.1	75.431	0.039	0.043	8.848	51.234
ppsta	22573	39.671(3.6)	1.757	25333.8	8796.3	99.79	255.7	39.585	0.025	0.037	2.739	21.144
srtm_srtm_224gp	198	37.000(3.4)	186.868	1737.8	564.9	95.27	17.2	36.982	0.005	0.016	0.018	36.105
srtm_spcvrt	198	33.532(3.1)	169.352	38988.0	16483.7	99.61	255.1	33.327	0.048	0.091	0.969	10.453
rrtm_rtrnla_140gp	198	30.626(2.8)	154.678	56396.3	19143.0	99.68	253.3	30.534	0.007	0.016	0.482	13.773
fpint12	9856	27.582(2.5)	2.799	16952.6	3782.0	99.60	240.8	26.064	0.006	0.011	2.194	17.785
ppintp	67518	20.821(1.9)	0.308	15669.3	2666.5	99.78	255.7	20.770	0.002	0.006	1.818	16.561
mf_phys	1078	19.087(1.7)	17.706	15059.5	2451.8	98.61	93.1	12.532	0.097	4.627	0.201	1.393
aval	4158	18.086(1.7)	4.350	39074.1	19380.7	99.65	255.8	17.570	0.074	0.144	0.343	5.293
laitri	11858	17.743(1.6)	1.496	52934.2	15848.8	99.96	255.1	17.731	0.003	0.004	4.009	6.891
lascaw	3234	14.632(1.3)	4.524	12601.1	1744.5	99.83	254.6	14.584	0.014	0.017	0.964	10.937

Profils FTRACE de l'exécution du modèle ARPEGE+SRTM sur le second processeur.

On est amené à faire des modifications sur la routine «srtm_reftra.F90», pour optimiser son temps d'utilisation du temps de calcul (CPU) par la vectorisation du moment ou on travaille sur un calculateur vectoriel.

Les modifications apportée sur la routine :

La routine modifiée se trouve sur yuki dans le chemin suivant :

~mrpa671/pack/cy35t2-op1_rad_op/src/local/arp/phys_radi

Cette routine permet le calcul de la transmissivité et de la réflectivité dans un ciel clair ou nuageux en utilisant des différentes approximations.

1- On a ôté les calculs indépendants, directement ou indirectement, de l'indice de la boucle.

```
.....
DO JK=1, KLEV
  !DO IC=1, ICOUNT
  ! JL=INDEX( IC )

  DO JL=KIDIA, KFDIA
```

.....
 On a remplacé le calcul de l'indice de boucle JL, par un indice déterminé (JL=KIDIA,KFDIA).

2- On a remplacé les calculs récurrents des variables temporaires par des vecteurs (tableaux) par ce que le calculateur vectoriel accomplit plus rapidement un bloc d'opérations identiques sur un tableau, qu'un calculateur scalaire.

3- Remplacer les boucles de test, par des expressions numériques en utilisant les fonctions max et min.

Exemple :

```
!      ZZZ= PTAU(JL,JK) / PRMUZ(JL)
!      IF(ZZZ <= 500._JPRB) THEN
!          ZE2 = EXP ( - ZZZ )
!      ELSE
!          ZE2 = ZEXPM500
!      ENDIF

      ZE1 = MIN ( PTAU(JL,JK) / PRMUZ(JL) , 500._JPRB)
      ZE2 = EXP ( - ZE1 )
```

4- Remplacer le test logique de la ligne par un masque :

```
!      IF (ZW*ZZZ >= ZWCRT*(ZZZ - (1._JPRB - ZW)*(ZG **2))) THEN

!!-- conservative scattering

ZMASK = (PW(JL,JK)*ZZZ)-ZWCRT*(ZZZ - (1._JPRB - PW(JL,JK))*(PGG(JL,JK) **2))
ZIMASK = MAX (0.0_JPRB, SIGN(1._JPRB,ZMASK))
```

Et le calcul final devient :

```
PREF(JL,JK)= ZIMASK * ZPREF1(JL,JK) + (1._JPRB -ZIMASK) * ZPREF2(JL,JK)
PTRA(JL,JK)= ZIMASK * ZPTRA1(JL,JK) + (1._JPRB -ZIMASK) * ZPTRA2(JL,JK)
PREFD(JL,JK)= ZIMASK * ZPREFD1(JL,JK) + (1._JPRB -ZIMASK) * ZPREFD2(JL,JK)
PTRAD(JL,JK)= ZIMASK * ZPTRAD1(JL,JK) + (1._JPRB -ZIMASK) * ZPTRAD2(JL,JK)
```

Pour plus de détail il faut consulter la routine «srtm_reftra.F90» au chemin cité au début de ce rapport.

Pour confirmer la réussite de la vectorisation, il faut faire un nouveau run en FTRACE, pour connaître les nouveaux temps de calcul passés dans chaque routine.

Les sorties FTRACE révèlent qu'on a diminué le temps de calcul passé dans la routine à 6% et 8% environs du temps de calcul globale du modèle, ce qui confirme la réussite de notre intervention sur la routine.

Une autre routine qui utilise pas mal de temps de calcul apparaît sur le profile, c'est la routine «srtm_vrtqdr.F90» avec une utilisation de 9% et 8% du temps de calcul.

La prochaine étape sera d'essayer d'optimiser cette dernière routine.

 FTRACE ANALYSIS LIST

Execution Date : Fri Oct 15 15:46:30 2010
 Total CPU Time : 0:14'33"542 (873.542 sec.)

PROC.NAME	FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	I-CACHE MISS	O-CACHE MISS	BANK CPU PORT	CONFLICT NETWORK
fpcincape	27270	176.335(20.2)	6.466	53043.3	28837.4	99.71	255.8	175.898	0.028	0.063	0.797	20.243
srtm_vrtqdr	44352	78.932(9.0)	1.780	15489.9	4578.3	99.89	255.1	78.792	0.032	0.035	8.848	54.524
srtm_reftra	44352	57.916(6.6)	1.306	47545.1	22665.1	99.71	255.1	56.537	0.013	0.107	0.075	13.781
ppsta	22573	39.979(4.6)	1.771	25138.2	8728.3	99.79	255.7	39.893	0.025	0.038	2.753	21.439
srtm_srtm_224gp	198	37.737(4.3)	190.589	1703.9	553.9	95.27	17.2	37.722	0.003	0.011	0.019	36.845

 FTRACE ANALYSIS LIST

Execution Date : Fri Oct 15 15:46:30 2010
 Total CPU Time : 0:11'16"711 (676.711 sec.)

PROC.NAME	FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	I-CACHE MISS	O-CACHE MISS	BANK CPU PORT	CONFLICT NETWORK
srtm_vrtqdr	44352	79.308(11.7)	1.788	15416.5	4556.6	99.89	255.1	79.166	0.033	0.036	8.848	54.717
srtm_reftra	44352	57.817(8.5)	1.304	47626.9	22704.1	99.71	255.1	56.437	0.015	0.109	0.077	13.687
fpcincape	8616	52.814(7.8)	6.130	53140.5	28888.4	99.71	255.5	52.674	0.011	0.024	0.242	5.970
mpl_barrier_mod.mpl_barrier	3683	45.884(6.8)	12.458	150.2	0.0	7.45	6.6	9.499	0.027	0.031	0.012	8.165
srtm_srtm_224gp	198	37.742(5.6)	190.614	1703.6	553.8	95.27	17.2	37.727	0.003	0.010	0.018	36.851
srtm_spcvrt	198	34.061(5.0)	172.027	38381.6	16227.3	99.61	255.1	33.889	0.061	0.041	0.942	11.082
rrtm_rtrnla_140gp	198	31.225(4.6)	157.702	55347.2	18795.1	99.68	253.4	31.136	0.005	0.013	0.488	14.289

Profils FTRACE de l'exécution du modèle ARPEGE+SRTM après optimisation de la routine
 «srtm_reftra.F90 » sur les deux processeurs.

Dans la routine « srtm_vrtqdr.F90 », on été amené à remplacer juste le calcul de l'indice de boucle par des variables déterminées :

```

.....
! DO IC=1, ICOUNT
!   JL=INDEX( IC )
DO JK=1, KLEV+1
.....

```

Le profile FTRACE de la version optimisée de ARPEGE+SRTM se présente comme suite :

```

*-----*
* FTRACE ANALYSIS LIST
*-----*

Execution Date : Sun Oct 24 11:31:49 2010
Total CPU Time : 0:11'06"790 (666.790 sec.)

PROC.NAME  FREQUENCY  EXCLUSIVE  AVER.TIME  MOPS  MFLOPS  V.OP  AVER.  VECTOR  I-CACHE  O-CACHE  BANK  CONFLICT
           TIME[sec]( % )  [msec]
           TIME  MISS  MISS  CPU PORT  NETWORK

srtm_reftra
44352      62.104( 9.3)   1.400  44339.6  21137.0  99.71  255.1  60.724  0.039  0.100  0.075  17.946
fpcincape  8616      57.590( 8.6)   6.684  48734.0  26493.0  99.71  255.5  57.458  0.007  0.018  0.241  10.726
mpl_barrier_mod.mpl_barrier
3683      40.233( 6.0)  10.924   147.5     0.0  7.46   6.6    8.984  0.040  0.050  0.010  7.834
srtm_srtm_224gp
198       39.462( 5.9)  199.302  1629.4    529.7  95.27  17.2   39.437  0.003  0.021  0.018  38.560
srtm_spcvrt
198       38.748( 5.8)  195.695  33739.6  14264.7  99.61  255.1  38.579  0.047  0.041  0.942  15.775
rrtm_rtrnla_140gp
198       35.317( 5.3)  178.367  48935.0  16617.6  99.68  253.4  35.228  0.005  0.013  0.488  18.382

```

Avec une utilisation finale de 9 % du CPU pour la routine «srtm_reftra.F90 » et de 4% pour la routine « srtm_vrtqdr.F90 ».

Les nouveaux temps de calcul CPU time se présentent comme suite :

Version non optimisée : 2039 s
Version optimisée : 1455 s
Version optimisée en utilisant MCICA : 9034 s

L'utilisation du recouvrement nuageux avec l'approximation MCICA, reste quand même coûteuse.

Conclusion et perspectives :

L'optimisation du code SRTM reste insuffisante pour la validation finale de ce code, Plusieurs simulations en version haute résolution sont nécessaire, ainsi que des comparaisons entre les normes spectrales d'une version opérationnelle de ARPEGE et notre version avec le code SRTM pour vérifier la fiabilité des sorties du modèle après notre intervention dans le code.