

Portage sur Prefix et dernières optimisations

Portage sur Prefix (nouveau supercalculateur)

- Compilateur **ifort**: portage presque immédiat (compilation interactive comme sur PC)
- Runs parallèles **MPI**: portage immédiat
- Runs parallèles **open-mp**: portage délicat. Réécriture d'une partie du code OFFLINE open-mp.
 - ⇒Préférer le MPI pour les runs parallèles
 - ⇒À terme, possible recodage local du open-mp avec ciblage des boucles lentes
- **Scripts** d'exécution: disponibles pour PGD, PREP, OFFLINE

Comparaison des performances TORI / PREFIX

Configuration SIM: 9892 points, ISBA seul + LTOWN_TO_ROCK, CISBA='DIF', 12 patchs, PHOTO=NON, CSNOW=3-L.

- *Temps passé sur les deux machines, en secondes, en fonction de nombre de procs MPI :*

procs	1	8	16	32	48	64	80	96
TORI	18380	3905 (/4.7)						
PREFIX	89632 (*3.2)	8139 (/7.33)	4061 (/14.68)	2110 (/28.27)	1518 (/39.29)	1184 (/50.37)	1013 (/58.86)	872 (/38.36)

- *Mémoire max par nœud utilisée en fonction du nombre de procs MPI :*

procs	1	8	16	32	48	64	80	96
TORI	1491	2739						
PREFIX		12335	9375	6294	4352	3918	3422	3070

Optimisation du PREP (1)

- Tayfun Dalkilic et Daan Degrauwe, de la communauté ALADIN, ont proposé deux voies d'optimisation du PREP:
 - Réduire le nombre d'appel à la routine d'interpolation horizontale + open-mp
 - Réduire les tailles du tableau XCOVER et de la boucle dans AV_PATCH_PGD_1D + open-mp

⇒ Choix basé sur un profiling du temps passé dans chaque routine durant le PREP
- Modifications intégrées dans la branche NEW_PREP

Optimisation du PREP (2)

- Profiling du PREP avant / après sur un cas OFFLINE:

Memory usage : 468 MBytes (heap), **460** MBytes (rss), 0 MBytes (stack), 0 (paging)

Wall-time is **28.78** sec on proc#1 (1 procs, 4 threads)

#	% Time (self)	Cumul (sec)	Self (sec)	Total (sec)	# of calls	Self ms/call	Total ms/call	Routine@<thread-id>
1	59.21	17.040	17.040	17.040	19	896.86	896.86	MODI_AV_PGD:AV_PATCH_PGD_1D_3@
2	26.23	24.589	7.548	7.549	685	11.02	11.02	BILIN_2@1
3	2.24	25.234	0.646	0.646	12	53.80	53.80	MODI_AV_PGD:AV_PGD_1D@1
4	2.21	25.871	0.637	0.637	685	0.93	0.93	BILIN_1@1
5	2.11	26.478	0.607	0.607	30	20.23	20.23	COEF_VER_INTERP_LIN_SURF@1
6	1.20	26.825	0.347	0.348	26449	0.01	0.01	LFILDO_MT@1
7	0.94	27.094	0.269	0.269	109	2.47	2.47	MODE_GRIDTYPE_CONF_PROJ:XY_CONF_1

Memory usage : 249 MBytes (heap), **217** MBytes (rss), 0 MBytes (stack), 0 (paging)

Wall-time is **2.74** sec on proc#1 (1 procs, 4 threads)

#	% Time (self)	Cumul (sec)	Self (sec)	Total (sec)	# of calls	Self ms/call	Total ms/call	Routine@<thread-id>
1	26.09	0.714	0.714	0.714	223	3.20	3.20	BILIN_2@1
2	12.47	1.055	0.341	0.343	26373	0.01	0.01	LFILDO_MT@1
3	10.16	1.333	0.278	0.278	19	14.63	14.63	MODI_AV_PGD:AV_PATCH_PGD_1D_3@1
4	7.62	1.541	0.209	0.209	223	0.93	0.94	BILIN_1@1
5	5.90	1.703	0.161	0.161	65	2.48	2.48	MODE_GRIDTYPE_CONF_PROJ:XY_CONF_PI

Optimisations du PGD (1)

- Les grilles optimisées sont **GAUSS**, **IGN** et **LONLATVAL**.
- Pour les autres grilles, au-delà d'une certaine résolution, la mémoire utilisée est limitante et non plus le temps de calcul.
- L'optimisation du PGD effectuée porte sur:
 - La détermination des indices de la grille de sortie pour chaque point des grilles physiographiques d'entrée (réduction du nombre de calculs effectués)
 - Le nombre d'appels à la routine qui détermine les indices
 - La parallélisation de certaines boucles avec OPEN-MP

Optimisations du PGD (2)

- Comparaisons avant / après sur la grille de GAUSS:

Avant sur Tori: **T539C24**: 38h58min / 43474MB

Troncature plus fine

Après sur PC, 4 threads: **T1798c24**: 424.81s / 4886MB

Troncature plus fine

Après sur Prefix, 16 threads: **T1198C22**: 337.34s / 10891MB

Perspective sur l'optimisation

Parallélisation du PGD et du PREP avec MPI pour
les grilles à haute résolution