

dc pam を用いた並列計算  
- NIES SX-6 での実行性能評価

# はじめに

- 前回, OpenMP を用いて並列化されている ispack とそれに対応した spml を用いて dcpam の並列計算を試みた.
- dcpam の並列化は成功し, eva05 を用いた性能評価実験結果を報告した.
- 今回は, NIES SX-6 での性能評価実験結果について報告する.

# 使用プログラムと計算環境

- プログラム
  - dcpam3-20060118
    - 並列化のために修正が必要(付録を参照のこと).
  - spml-0.4.0-2
  - gt4f90io-20050720
    - 並列化のために修正が必要(付録を参照のこと).
  - ispack-0.71
- 計算環境
  - NIES SX-6

# 実験条件

- Held and Suarez (1994) dynamical core test
  - T42L20
  - 等温静止の初期条件から 10 日間積分.
  - 計算結果は 1 日毎に出力(データ出力は膨大ではない).
  - 力学過程は, OpenMP によって並列化された ispack とによって並列実行される.
  - 物理過程(実体はレイリー摩擦とニュートン冷却)は並列実行されない.

# 性能評価実験結果

CPU 数	Real Time (s)	速度比
1	1887	1.00
2	1266	1.49
4	1230	1.53
6	-	-
8	-	-

- なぜか 4 CPUs までしか上手く動かない。
- 2, 4 CPUs 使用時の速度は単一 CPU 使用時の約 1.5 倍。

# Flow Trace Analysis List

- 1 CPU

EXCLUSIVE TIME[sec]( % )	AVER.TIME [msec]	V.OP RATIO	AVER. V.LEN	PROG.UNIT
925.399( 51.0)	0.473	96.29	78.9	w_base_module.w_xy
381.314( 21.0)	3.923	98.81	189.2	wa_base_module.wa_xya
300.370( 16.6)	4.172	98.81	190.9	wa_base_module.xya_wa
56.256( 3.1)	15.627	96.87	134.0	dynamics_mod.dynamics_leapfrog
51.068( 2.8)	14.186	99.09	129.7	hs94forcing_mod.hs94forcing
30.688( 1.7)	0.341	0.47	179.3	varinfo_mod.varinfo_inquire
15.530( 0.9)	15529.677	36.11	63.5	dycore_in_mod.dycore_in_vars

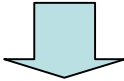
- 4 CPUs

EXCLUSIVE TIME[sec]( % )	AVER.TIME [msec]	V.OP RATIO	AVER. V.LEN	PROG.UNIT
541.733( 44.2)	0.277	93.61	68.1	w_base_module.w_xy
260.410( 21.3)	2.679	95.47	111.0	wa_base_module.wa_xya
203.462( 16.6)	2.826	94.90	112.8	wa_base_module.xya_wa
63.153( 5.2)	17.543	96.48	134.0	dynamics_mod.dynamics_leapfrog
48.433( 4.0)	13.454	99.09	129.7	hs94forcing_mod.hs94forcing
40.742( 3.3)	0.453	1.08	213.4	varinfo_mod.varinfo_inquire
13.139( 1.1)	13138.520	36.04	63.5	dycore_in_mod.dycore_in_vars

# まとめ

- NIES SX-6 でも dcpam は複数の CPU を用いて実行可能である.
- ただし, 現状では 4 CPUs までしか同時に使えない. 原因は不明である.
- 並列実行性能はあまり高くなく, 4 CPUs を用いた場合の計算速度は, 単一 CPU を用いた時の約 1.5 倍程度であった. 現状の NIES SX-6 の使用状況から考えると, この程度の高速化はあまり意味がないかもしれない.

# 付録: gt4f90io に関する特記事項 1

- gt4f90io-20050720 の修正
  - src/an\_file.f90 において 183 行目を修正.  
mode = NF\_NOCLOBBER  
  
mode = NF\_CLOBBER
  - 最新版ではこの部分は変更されているため, 上記の修正は必要ないはず.
  - この修正に関する詳細は次ページを参照のこと.

## 付録: gt4f90io に関する特記事項 2

- この修正は, (NIES) SX-6 において, OpenMP を使ったプログラムがなぜかファイルを新規作成できないために必要となる.
- 問題の回避方法に関する報告は森川君による次のページ [http://www.gfd-dennou.org/library/gtool4/gt4tutorial/2006-06-11\\_morikawa\\_sx6\\_install/#label-7](http://www.gfd-dennou.org/library/gtool4/gt4tutorial/2006-06-11_morikawa_sx6_install/#label-7) を参照のこと.
- 上記ページの報告に基づき, ここでは結果のファイルを予め作っておき, それを上書きすることにする. 使用した gt4f90io-20050720 はデフォルトで上書きを許さないため, 上書きを許すように変更した.

# 付録: dcpam に関する特記事項

- 並列化に対応した spml を dcpam から使うためのコードの修正に関しては前回の資料を参照.
- コンパイルオプションに `-Popenmp` を追加.
- `/opt/SX/opt/sxf90/rev335/include/module/mo_omp.f90` をコピーして一緒にコンパイルする必要がある(? 少なくともそうすると動く).
- 中身はなくても良いので, 結果のファイルを作っておく必要がある(理由については, `gt4f90io` の修正に関する付録を参照のこと).

# 付録: ジョブの実行に関する特記事項

- 使用する CPU 数が 8 以外の時には, ジョブの投入時に CPU 数を指定する必要がある ( ? 指定しない場合に動かないかどうかは確認していない).
  - 例:

```
% qsub -c 4 -q vector_ps -lT 7200 model_sx.sh
```
- ジョブスクリプト内で, 環境変数 OMP\_NUM\_THREADS に使用する CPU 数を指定.
  - スクリプトの例:

```
#!/bin/csh
setenv F_PROGINF DETAIL
setenv F_SYSLEN 327670
setenv F_FMTBUF 327670
setenv F_FTRACE FMT1
setenv OMP_NUM_THREADS 4
cd /home1/cger/ggec101/yot/dcpam/dcpam3-20060118-parallel-4/practice
#../bin/init >& dcpam.msg
../bin/dcpam >& dcpam.msg
```

# 付録: AFES 火星版の並列化性能

CPU 数	Real Time (s)	速度比
1	2457	1.00
4	862	2.85
8	732	3.36

- 思っていたよりも遅い
  - こんなものか？ (もっと速いと思っていたのは個人的な幻想？)
  - 火星版にしたから遅い？
    - セミラグの実装が悪い？
    - 物理過程の実装が悪い？

# 謝辞

- 性能評価実験には NIES SX-6 を使わせていただきました。

# 参考文献

- Held and Suarez (1994), BAMS, 75, 1825