

Ubuntu Linux での気象庁 Nhm のメイクと六甲おろしのシミュレーション

余田 史絵⁽¹⁾・*金子 晃⁽²⁾ (お茶の水女子大学)

1 はじめに

気象庁数値予報課から非静力学モデルのソースの貸与を受け、Ubuntu Linux 上でメイクし、冬の六高山における風速分布の計算を行ってみた結果を報告する。

2 使用した計算機環境

貸与を受けて1年ほどはメイクに手間取っていたが、2010年5月にリリースされた日本語マニュアル [1], [2] が大変分かりやすく、気象庁数値予報課の成田氏、気象研究所の林氏のご教示のお蔭もあって、ようやくメイクに成功し、使ってみることができた。以下メイクの記録をやや詳細に報告する。Ubuntu Linux がよく普及していて使いやすいことと、数万円程度の自作パソコンで十分並列計算が可能なことから、うちのような小規模研究室でも十分 Nhm が動かせるという経験が参考になるかもしれないと思ったからである。

本研究で用いたマシンは以下の通りである。

diamond: Opteron 2431 2.4GHz 6 コア × 2, 主記憶 32 GB. CentOS 5.3 (3 研究室共用, 60 万円)

goophy: Intel Core i7 860 2.8GHz 4 コア (Intel hyperthread 技術で 8 コアに見える), 主記憶 8 GB. Ubuntu 10.04 (x64 版) (自作, 約 8 万円)

snufkin: AMD Phenom II X6 3.2GHz 6 コア, 主記憶 8 GB. OS は同上 (自作, 約 6 万円)

sapphire: Intel Core 2 Duo 2.53GHz 2 コア, 主記憶 4GB. OS は同上 (ノートパソコン, 約 9 万円)

Ubuntu のバージョンについては、その後今回の予稿を書くのに際して、Ver.10.10 でも追試した。Ver.11.04 は (猛暑による?) 原因不明のマシントラブルのため、まだできていない。

並列計算には openmpi を用いた。その主な理由は、diamond で openmpi の方が若干速かったのと、Ubuntu で普通に apt-get できる mpich ではマルチコアを使ってくれず、ネット上の他のマシンを探しに行ってしまうようだったからである。

参考までに、これらの機械の並列計算の速度比較を挙げておく。用いたソフトは東大情報基盤センターで 2010 年 3 月に HA8000 の一部のノードを用いて行われた MPI セミナーで使われた、2048 次行列同士の乗算の C プログラム mat-mat.c である。並列数の欄は、コア数 × ノード数、の形に表示している。計算の所要時間の単位は秒である。数値は実行の度変わるので、概

数を示してある。Intel Core i7 が CPU クロックに比して驚異的に高速だが、コンパイラの並列最適化が AMD CPU に合っていないためかもしれない。

機種	CPU 速度	並列数	所要時間	MFLOPS
HA8000	2.3GHz	16 × 4	9.46	1816
goophy	2.8GHz	4 (×2)	28.17	610
diamond	2.4GHz	6 × 2	69.36	248
snufkin	3.2 GHz	6	106.2	162
sapphire	2.53GHz	2	156.2	110

3 Ubuntu Linux での Nhm のメイク

使用した Nhm のバージョンは CVS で取得した 2010 年 5 月版のソースで、使用されるデータ類は、それ以前に ftp で取得したものを用了。いずれも最近確かめた限りではその後の変更は無いようである。2010 年 5 月にリリースされた日本語マニュアル [1], [2] を参考にメイクを行ったが、ほとんどは [1] の記述通りで済んだ。以下、変更あるいは自分で補った点のみを記す。

1. メイクに必要なソフトの準備。Ubuntu の場合、OS の標準インストールでは開発環境は gcc 以外ほとんど入っていないので、apt-get でコンパイラ gfortran, g++, 並列計算ツール openmpi-bin, libopenmpi-dev, (numactl) の他, gawk, bison, flex, m4, automake, libtool などの基本的な開発ツールを追加する。libjasper-dev, grads (libhdf5 も依存関係で入る) も追加しておく。ソースの修正に emacs も入れた方が便利である。

2. NuSDaS 用ソフトのメイク。ここは [1] の通りにやると、g22n.h が無いというエラーで止まる。これは取り敢えず無くてもよいものらしいので、最も簡単な対処法はこの部分をスキップすることである。すると、後に make nhmcst_n01 の実行時に、依存関係により NuSDaS 用の必要最小限のメイクも自動的に行われる。

3. タイムカードツールのメイク。[1] の通りでよいが、tcredit.c をコンパイルするところで、exit 関数の型が合わない等のエラーが出るので、#include<stdlib.h> をソースに追加して対処する。

4. config ファイルの修正。Module/Mk/config/pc に行き、config.g95_openmpi をコピーして

```
config.gfortran_openmpi
```

を作り、これに以下のような修正を施す。g95 を gfortran に書き直す箇所は実は存在しないが、openmpi の参照先を次のように書き直す必要がある。

```
MPIHOME=/usr/lib/openmpi
INC_MPI=$MPIHOME/include
LIB_MPI="-L$MPIHOME/lib -lmpi"
```

(1) 現在日産自動車。(2) 現在定年退職

更に、FFLAGS="-fendian=big" は消しておかないと、コンパイル時に unrecognized command line option のエラーが出る。これだけでメイクはでき、本体は恐らく little endian で正しく動作するのだろうが、実データ実験で付属ソフトが正しく動かず、mftopo の標高データなど一部の内容におかしな数値がセットされ、計算結果もおかしくなる。ここは、config.gfortran に記述されている FFLAGS="-fconvert=big-endian" を書いておくのが良いようである。これで mftopo データが big endian で作成され、計算もうまく行く。ついでに、

```
JASPERLIB=/usr/lib
JASPERINC=/usr/include/jasper
```

も追加しておくといよい。(-ljasper は後に config スクリプトにより自動的に追加されるので不要。)

5. メイク。以上の準備のもとに Module/Mk に戻って

```
sh configure -c
config/pc/config.gfortran_openmpi
```

(実際には1行)を実行する。この後、nhmcst_n01 から始まる一連のメイクを [1] の通りに実行する。これらのメイクスクリプトは開発ツールに足りないものが有っても config 段階でエラー終了はせず、次に進めてしまうので、ログを見て自分で確認する必要がある。

6. その他。Tools/grib2nus の下で sh configure -c ../../Module/Mk/config/pc/config.gfortran_openmpi

の後 make を実行すると、ncepgrib22nus.o が無いというエラーで止まるが、必要なものはできており無視してよいようである。nus2grads のメイクは [1] になぜかここだけ ./configure と記述されているが、配布ファイルの属性上 sh configure とすべきで、それ以外は問題ない。pandah のメイクは、Tools/pandah/Mk の下で config.g95 を config.gfortran にコピーし、その中の g95 を gfortran に書き直して

```
sh configure -c config.gfortran
make pandah
```

を実行する。このとき、pandah/Src/stl2.f のコンパイル時に 99 行の DATA ZFF /Z'FFFFFFFF' が Arithmetic overflow を起こすので、エラーメッセージの指示に従い、config.gfortran の OPT_F="-O" の記述を

```
OPT_F="-O -fno-range-check"
```

に変えてからやり直した。更に、nsymb1.f の 1323 行の ";" SKIP=0." という行に対して、セミコロンの前に指令が無い、というエラーが出るので、セミコロンを削除して再実行した。これらのエラーは gfortran の古いバージョンでは起きない。最後の ../Sh では

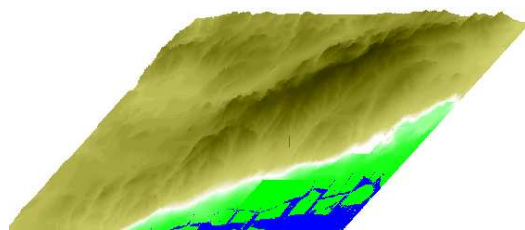
```
sh configure -c config.g95
```

を実行するだけでよい。この config ファイルには、g95 の記述は無いので、そのまま使え、pandah.sh ができる。この後の ../Parm/map での地図輪郭の準備は [1]

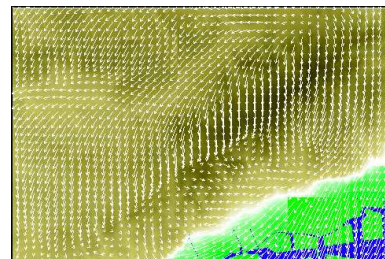
の通りである。Tools/webpandah での作業には、自分のホームディレクトリに public_html というディレクトリを予め作成しておく必要がある。gtopo.c には #include<string.h> を追加した方がよいだろう。

この後の定数データの準備は [1] の通りで概ね問題無いが、一部は ftp サイトの /pub/nhmconst から取らねばならない。以上の後、スクリプトが供給されているモデル実験を試してみた。理想実験の Lmwv では、grads 版はすんなり行くが、nusdas 版はスクリプトが欠落しており、気象研の林氏にメールで送ってもらった。また実行すると Tools/pandah/Parm/trjct_table.txt が無いというエラーに会う。実際には不要ようだが、これも林氏にメールで送ってもらった。この他のトラブルについての詳細は [3] に書いた。実データ実験の RF20km test は、[1] には、Intel Core 2 duo 2.3GHz で2時間半程度と書かれているが、私達の追試では、snufkin (6 並列), goophy (8 並列) とともに約 37 分であった。

4 六甲おろしのシミュレーション



GTPO30 は 1 km のメッシュしかないので、六甲山系を覆うには荒すぎる。国土地理院から 50 m メッシュのデジタル標高データが比較的安価で販売されているが、これは後で知ったので、ネットで国土地理院の『うおつちず』の画面を取り込み、画像処理と手作業で 50 m 間隔の等高線を抽出し、それらから補間により、ほぼ同じサイズのメッシュデータを作成して用いた。mftopo 仕様のファイルはこれから自分で作った。その表示ツール等も自作した(上図)。スクリプトは RF20km test を書き直して用いた、下図は 2011 年 1 月 3 日 12:00 の地表面における風速場を、その 12 時間前の実データを初期値として計算したものである。これを見ると、冬期の六甲山系は季節風を遮っている感じにも見える。



参考文献

- [1] 気象庁編『NHM ユーザーズガイド』, 2010 年 5 月.
- [2] 同『NHM デベロッパーズガイド』, 2010 年 5 月.
- [3] 余田史絵『局所気象現象の数値的研究』, お茶の水女子大学修士論文, 2011 年 1 月.