

# PCクラスタを用いた 風況予測シミュレータRIAM-COMPACTの開発 —その1. 種々の計算機におけるCPU時間の比較—

内田 孝紀\*, 大屋 裕二\*

(2003年7月31日受理)

## Development of the RIAM-COMPACT using a PC cluster —Part.1 Comparison of the CPU time in various computers—

Takanori UCHIDA and Yuji OHYA

E-mail of corresponding author: [takanori@riam.kyushu-u.ac.jp](mailto:takanori@riam.kyushu-u.ac.jp)

### Abstract

We are developing the numerical model called the RIAM-COMPACT (Research Institute for Applied Mechanics, Kyushu University, Computational Prediction of Airflow over Complex Terrain). The object domain of this numerical model is from several m to several km, and can predict the flow of the wind over complex terrain with high precision. Since we put this numerical model in practical use, we are considering introduction of a PC cluster. This research shows comparison of the CPU time in various computers as the first phase. Consequently, it became clear that the newest computer environment has about the same performance as a supercomputer. The difference in a compiler option is also discussed.

**Key words :** RIAM-COMPACT, Various computers, CPU time, Compiler option

## 1. 緒言

日本国内の地勢は欧米とは著しく異なり、平坦な地形は少なく、多様性に富む複雑地形がほとんどである。こうした状況において、風力タービン設置のための風況精査(適地選定)や、大型ウインドファーム建設後の局所風況場のリアルタイムシミュレーション(日々の発電量予測)を高精度に実施するためには、流れの衝突、剥離、再付着、逆流などの風に対する地形効果を再現することが極めて重要である。

我々は数百m～数(十)km程度の局所域スケールの問題を絞り、RIAM-COMPACT(Research Institute for Applied Mechanics, Kyushu University, Computational Prediction of Airflow over Complex Terrain)と称する風況予測シミュレータを開発している<sup>1)</sup>。この数値モデルは、(有限)差分法(FDM(Finite-Difference Method))に基づいたFORTRAN(FORMula TRANslater)プログラムである。これ以降はFortranと記述する。乱流モデルには非定常流体シミュレーションが可能なLES(Large-Eddy Simulation)を採用している。既にRIAM-COMPACTの予測精度の検証、実地形上の風況場解析などを行い、風洞実験結果と比較してその有効性を確認している。

現在はRIAM-COMPACTを実用化するため、PCクラスタ(cluster)によるスカラー並列計算を検討している。

PCクラスタとは、PC(Personal Computer)を構成要素(ノード)とし、各ノードを高速ネットワークで複数台相互に接続した分散メモリ(distributed memory)型の並列計算機(parallel computer)である。詳細は後述する。本研究では、その第一段階として最近の計算機性能を調べた。つまり、種々の計算機を用い、同一条件の下でRIAM-COMPACTによる風況予測シミュレーションを実施した。計算の対象は急峻な孤立峰を過ぎる流れ場(中立成層流)である。特にFortranのコンパイル(翻訳)のオプションとCPU時間に注目し、それらの比較結果について報告する。

## 2. 検討した種々の計算機環境

ここでは、本研究で検討した種々の計算機環境、すなわち、それぞれの計算機の仕様(スペック)を示す。

表1には、我々の研究室で所有している計算機環境を示す。少々古い機種も含まれるが、比較検討のために使用することとした。

表2には、応用力学研究所所有の計算機環境を示す。分散メモリ方式のベクトル並列型スーパーコンピュータVPP5000/2(富士通株式会社)と、SMP(Symmetric Multi Processor)クラスタES40(米コンパック・コンピュータ:現在、米ヒューレット・パッカー)がある。

\*九州大学応用力学研究所

| Machine name | Model               | CPU                    | Memory  | OS                |
|--------------|---------------------|------------------------|---------|-------------------|
|              | Vendor              | Clock                  |         |                   |
| M-01         | NetVista A22p(6823) | Intel Pentium 4        | 1.5GB   | WinXP SP1         |
|              | IBM                 | 2.20GHz                |         |                   |
| M-02         | Handmade            | Intel Pentium 4        | 1,024MB | Win2000 SP3       |
|              | —                   | 2.26GHz                |         |                   |
| M-03         | Handmade            | AMD Athlon XP 2200+    | 1,024MB | Win2000 SP3       |
|              | —                   | 1.8GHz                 |         |                   |
| M-04         | ThinkPad X30 (4XJ)  | Mobile Intel Pentium 3 | 1,024MB | WinXP SP1         |
|              | IBM                 | 1.20GHz                |         |                   |
| M-05         | FMV-6500DX4         | Intel Pentium 3        | 512MB   | Win98 2nd Edition |
|              | Fujitsu             | 500MHz                 |         |                   |
| M-06         | Handmade            | Intel Pentium 3        | 1,024MB | Red Hat Linux 7.2 |
|              | —                   | 800MHz                 |         |                   |
| M-07         | VT-Alpha6           | Alpha 21264            | 512MB   | WinNT v.4.0 SP3   |
|              | Visual Technology   | 666MHz                 |         |                   |
| M-08         | VT-Alpha6 500DP     | Alpha 21264            | 1,024MB | WinNT v.4.0 SP3   |
|              | Visual Technology   | 500MHz                 |         |                   |
| M-09         | VT-Alpha6 600       | Alpha 21164            | 512MB   | WinNT v.4.0 SP1   |
|              | Visual Technology   | 600MHz                 |         |                   |
| M-10         | Alpha               | Alpha 21164            | 256MB   | WinNT v.4.0 SP6   |
|              | iCFD                | 300MHz                 |         |                   |

Table.1 Computer environment of our laboratory

| Machine name | Model             | CPU                             | Memory                | OS         |
|--------------|-------------------|---------------------------------|-----------------------|------------|
|              | Vendor            | Clock                           |                       |            |
| M-11         | VPP5000/2         | PE×2                            | 1.5GB B0 and B1 queue | UXP/V V20  |
|              | Fujitsu           | —                               |                       |            |
| M-12         | Alpha Server ES40 | Alpha 21264 A (codename : EV67) | 1GB Serial E queue    | Tru64 UNIX |
|              | Compaq            | 667MHz×12 (4CPU×3) (1.33GFLOPS) | 2GB SMP S queue       |            |

Table.2 Computer environment of the computer room of RIAM

| Machine name | Model              | CPU                             | Memory                        | OS         |
|--------------|--------------------|---------------------------------|-------------------------------|------------|
|              | Vendor             | Clock                           |                               |            |
| M-13         | VPP5000/64         | PE×64                           | 7GB/2GB s queue               | UXP/V V20  |
|              | Fujitsu            | —                               | 15GB/10GB x16 queue           |            |
|              |                    |                                 | 7GB/2GB p32 queue             |            |
| M-14         | Alpha Server GS320 | Alpha 21264 B (codename : EV68) | 64GB                          | Tru64 UNIX |
|              | Compaq             | 731MHz×32 (4CPU×8)              | 16GB for one user interactive |            |
| M-15         | GP7000F Model 900  | SPARC64-GP                      | 64GB                          | Solaris7   |
|              | Fujitsu            | 300MHz×64                       | 32GB for one user sc32 queue  |            |

Table.3 Computer environment of the computing and communications center of Kyushu University

| Machine name | Specification   |
|--------------|---|
| M-16         | CPU : Opteron 1.6GHz×2 (Dual Processor)<br>Mem : 2GB (512KB×4)<br>OS : Turbo Linux 8 for AMD 64,<br>KERNEL 2.4.20-1 |
| M-17         | CPU : Opteron 1.8GHz×2 (Dual Processor)<br>Mem : 2GB (512KB×4)  |

Table.4 Computer environment of Visual Technology

| Machine name | Specification   |
|--------------|---|
| M-18         | Hostname : p4103<br>CPU : Pentium 4 2.4GHz (512KB cache/FSB400)<br>Mem : 1GB RDRAM (PC800-45nsec)   |
| M-19         | Hostname : p4104<br>CPU : Pentium 4 3.06GHz (512KB cache/FSB533)<br>Mem : 1GB RDRAM (PC1066)  |
| M-20         | Hostname : p4110<br>CPU : Pentium 4 3.06GHz (512KB cache/FSB533)<br>Mem : 4GB DDR266 SDRAM  |
| M-21         | Hostname : p4111<br>CPU : Pentium 4 3.0GHz (512KB cache/FSB800)<br>M/B : i875PE chipset<br>Mem : DDR400 512MB×4 = 2,048MB<br>HDD : Maxtor 6Y080L0 (IDE 80GB 7200rpm)<br>OS : SuSE Linux 7.3 Kernel 2.4.20 |
| M-22         | Hostname : iax01<br>CPU : Xeon 2.8GHz Dual (512KB cache/FSB400)<br>Mem : 8GB DDR200 SDRAM   |
| M-23         | Hostname : iax02<br>CPU : Xeon 2.8GHz Dual (512KB cache/FSB400)<br>Mem : 4GB DDR200 SDRAM   |
| M-24         | Hostname : iax06<br>CPU : Xeon 2.8GHz Dual (512KB cache/FSB533)<br>Mem : 3GB DDR266 SDRAM   |
| M-25         | Hostname : iax06<br>CPU : Xeon 3.06GHz Dual (512KB cache/FSB533)<br>Mem : 2GB DDR266 SDRAM  |
| M-26         | CPU : Xeon 3.06 GHz×2 (Dual Processor)<br>Memory : 2,048MB<br>Chipset : E7501   |
| M-27         | Hostname : ita1<br>CPU : Itanium 2 1.0GHz 3MB cache Quad SMP<br>Mem : 16GB DDR266 SDRAM<br>HDD : SCSI 36GB<br>OS : Red Hat Advanced Workstation 2.1   |
| M-28         | Hostname : ita1<br>CPU : Itanium 2 900MHz 1.5MB cache Quad SMP<br>Mem : 32GB DDR200 SDRAM<br>HDD : SCSI 36GB<br>OS : Red Hat Advanced Server 2.1AS  |
| M-29         | CPU : Itanium 2 1.5GHz 6MB cache Dual SMP   |

Table.5 Computer environment of the Open-SCC of HIT

分散メモリ方式では、OS(Operating System)とメモリ(memory)はそれぞれのCPU(Central Processing Unit),あるいは、PE(Precessing Element)に文字通り分散している。各CPU(PE)は完全に独立しており、高速ネットワークでお互いに接続されている。VPP5000の場合には、各PE間はクロスバーネットワーク(crossbar network)により接続されている。通信速度は1.6GB/sである。並列計算を行う際には、各CPU(PE)間を通信するためのメッセージ・パッシング・ライブラリMPL(Message Passing Library)

と呼ばれる予め用意されたライブラリが必要になる。MPLはFortranなどのプログラミング言語ではない。メッセージ・パッシングについて簡単に説明する。分散メモリ方式の並列計算機においては、データや仕事をCPU(PE)の数で分割するので、各プロセス間でデータの交換を行う必要が生じる。ここで、あるプロセスのメモリ空間から別のプロセスのメモリ空間へのデータ転送のことをメッセージ・パッシングと言う。つまり、メッセージ・パッシングとは通信手段の一つである。

一方、SMPクラスタとは分散メモリ方式と、後述する共有メモリ(shared memory)方式をミックスしたような形式である。各ノード(node)の内部では数台のCPU、あるいは、スレッド(thread)がメモリを共有するSMPを構成し、ノードをまたぐ場合には、分散メモリ方式のように高速ネットワークで接続される。すなわち、各ノード(SMP)がクラスタを構成する。共有メモリ方式(SMP)内では、CPU(スレッド)がメモリを共有しているため、メッセージ・パッシング(通信手段)は必要なく、エンドユーザーはあたかも一台のCPUを使用するイメージでSMP環境を利用できる。さらに、ほとんどの場合においてFortranコンパイラは自動並列化(automatic parallelization)機能を備えている。本研究では、これを利用しその有効性を検討する。なお、ノードをまたぎ、複数のCPU(スレッド)を用いて並列計算を行う際には、当然ながら何らかのメッセージ・パッシング(通信手段)が必要となる。

さて本報では、VPP5000(便宜上、M-11と称する)の1PEを用いたベクトル逐次計算の結果(Table.7, No.28)をCPU時間の比較基準とした。すなわち、他の機種別のCPU時間については、M-11の計算結果との比(=各計算機のCPU時間/M-11のベクトル逐次計算のCPU時間(Table.7, No.28))として表示した。ここで、逐次処理(serial processing)とは、1PEでの計算を示す。ES40については、スカラー逐次計算および自動並列化機能を利用した4CPU(4スレッド)までのスカラー並列計算を行った。

表3には、九州大学情報基盤センターの計算機環境を示す。ここには、応用力学研究所と同様、分散メモリ方式のベクトル並列型スーパーコンピュータVPP5000/64がある。VPP5000/64では、ベクトル逐次計算および1~32PEまでのベクトル並列計算を行った。スカラー並列サーバGS320(米コンパック・コンピュータ:現在、米ヒューレット・パカード)は、応用力学研究所にあるES40の上位機種である。GS320の基本ユニットは、QBB(Quad Building Block)と呼ばれ、4CPUおよび4メモリ・モジュールが搭載されている。QBB内は共有メモリ・プロセッシングである。ES40と大きく異なる点は、QBBをまたいだ並列処理(parallel processing)も共有メモリ・プロセッシングであるという点である。すなわち、GS320は

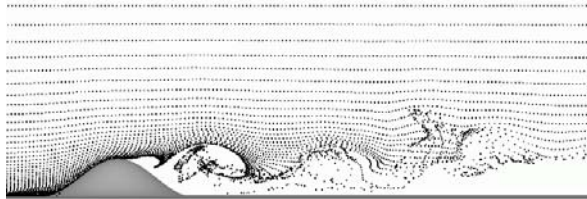
共有メモリ方式のスカラー並列計算機である。但し、QBB内とそれをまたぐ場合ではメモリ・アクセス速度が異なる。これについては後述する。ここでは、スカラー逐次計算および自動並列化機能を利用した8CPU(8スレッド)までのスカラー並列計算を行った。汎用UNIXサーバのGP7000Fモデル900は、共有メモリ方式のスカラー並列計算機である。ここでは、自動並列化機能を利用し、1~32CPUまでのスカラー並列計算を行った。

表4には、ビジュアルテクノロジー株式会社の計算機環境を示す。ここでは、AMD社の新しい64Bitマイクロプロセッサ(microprocessor)であるOpteronを搭載した機種において、スカラー逐次計算および自動並列化機能を利用したDual CPU(2スレッド)のスカラー並列計算を評価していただいた。

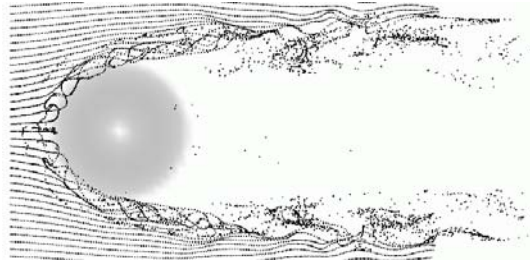
表5には、株式会社エッチ・アイ・ティー所有のOpen-SCC(Open Super Computing Center)の環境を示す。ここには、大型計算機センター並みのHPC(High Performance Computing)環境が整備されている。ユーザー登録を行えば、それらの計算機(host)をネットワーク経由で自由に利用できる。具体的には、Table.2およびTable.3に示す計算機環境と同様、SSH(Secure SHell)やTELNETによりTCP/IP(Transmission Control Protocol/Internet Protocol)接続された計算機にリモート・ログイン(remote login)し、その計算機を遠隔操作できる。これらのSSH、TELNETサーバ機能を利用することで、ユーザー(クライアント)は離れた所にある計算機をまるで自分の前にあるかのように操作できる。また同時に、FTP(File Transfer Protocol)によるファイル転送(ファイルの送受信)も可能である。ここでは、32Bitおよび64BitのIntel社のマイクロプロセッサであるPentium 4(IA-32, Intel Architecture 32), Xeon(IA-32), Itanium 2(IA-64)などの最新のCPU性能が評価できる。詳しくは後述するが、各種Fortranコンパイラのテストや最適化オプションの比較が可能である。本報では、これらの結果を示す。

### 3. 計算対象とパラメータ設定

ここでは、RIAM-COMPACTによる数値シミュレーションの概要について示す。一般曲線座標系のコロケート格子に基づいたRIAM-COMPACTを用い、急峻な孤立峰を過ぎる流れ場の数値シミュレーションを行う。数値計算法などの詳細については、文献<sup>1)</sup>を参照していただきたい。パッシブ粒子追跡法によって視覚化した孤立峰周辺の流れパターンをFig.1に示す。ここで、パラメータなどの条件設定は後述する。Fig.1(a)では孤立峰下流で渦放出が、Fig.1(b)では孤立峰を取り囲むように形成された馬蹄渦が明確に観察される。このように、孤立峰



(a) Side view at the central plane  
in the y-direction ( $y=0$ )



(b) Top view near the ground

**Fig.1 Flow pattern around a steep isolated-hill  
visualized by the passive particle tracking method**

周辺は複雑乱流場を呈していることが見て取れる。

パラメータなどの条件設定は以下の通りである。孤立峰の主流方向にx軸を、主流直交方向(スパン方向)にy軸を、鉛直方向にz軸を設定する。計算領域はx, y, z方向に $23h \times 16h \times 5h$ の空間領域を有する。ここで、 $h$ は孤立峰の高さである。孤立峰は流入境界面から $8h$ 下流のスパン中央地面上に設置する。その形状は余弦の二乗であり、関数 $z(x, y) = h \times \cos^2(\pi(x^2+y^2)^{1/2}/2L)$ で記述される。ここで、地形形状パラメータは $L=2h$ である。格子点数はx, y, z方向に $81 \times 61 \times 51$ 点(約25万点)である。x, y方向の格子解像度は同じで不等間隔に $(0.2 \sim 0.55)h$ 、z方向の格子解像度は不等間隔に $(0.003 \sim 0.5)h$ である。速度の境界条件に関して、流入境界面は一様流入条件、側方境界面と上部境界面は滑り条件、地面は粘着条件、流出境界面は対流型流出条件とする。レイノルズ数は孤立峰の高さ $h$ と一様流入風速 $U$ に基づいて $Re(=Uh/\nu) = 10^4$ とした。時刻刻みは $\Delta t = 2 \times 10^{-3}h/U$ とした。同一条件の下でCPU時間を比較するため、孤立峰周辺の流れ場が十分に発達した無次元時間 $t=100$ の計算結果を入力データとし、 $t=100 \sim 110$ における計算(5,000ステップの時間積分)を各計算機で実施した。ここで、CPU時間について説明する。一般には、

経過時間(elapsed time)=

CPU時間(=ユーザーCPU時間+システムCPU時間)  
+I/O(Input/Output)時間

(1)

という関係が成り立つ。本研究では、システムCPU時間とI/O時間は全ての計算においてわずかであったので無

視し、

CPU時間 $\approx$ 経過時間

(2)

とした。ここで、経過時間(elapsed time)とは計算の開始から終了までに要する実時間である。これは他の計算が実行されている場合には、メモリ帯域が各プログラムに分配されるので異なる結果を示す。本研究における全ての結果は、他の計算が実行されていない状態で計測された値である。ユーザーCPU時間とは、ユーザプログラム自体が消費したCPU時間である。システムCPU時間とは、OSのオーバーヘッド(overhead)に消費された時間である。オーバーヘッドとは、プログラムの計算以外で処理に遅延が生じることを意味する。システムCPU時間はI/O処理が多い場合には増加する。なお、並列計算においては、各CPU(スレッド、PE)におけるCPU時間の合計がユーザーCPU時間に表示されるので、経過時間をそのまま使用した。CPU時間および経過時間の測定には、CALL CPU\_TIME(time)などの組込みサブルーチン、あるいは、timeおよびtimexコマンドを利用した。詳細は後述する。

## 4. 結果と考察

### 4.1 研究室所有の計算機環境について

ここでは、我々の研究室で所有する計算機環境(Table.1)で得られた結果(Table.6)について考察する。これらの計算は全て対話型処理で行った。対話型処理とは、コマンドライン(command line)上でコンパイルや実行などをインタラクティブ(interactive)に行うことである。

M-01(IBM NetVista)の結果に注目する。No.1~No.4では、Intel Fortranコンパイラのバージョン(7.0, 7.1)に関して有意な違いは見られない。No.3とNo.5を比較すると、Intel Pentium 4に特化したオプションを含め、最適化オプション(コンパイルオプション)の効果は顕著に現れている。最適化オプションが有る場合のNo.5では、それが無い場合のNo.3に比べて約1.13倍(=No.3/No.5)の計算速度の向上率が得られている。すなわち、No.3の計算に要したCPU時間の約11%( $= (1 - \text{No.5}/\text{No.3}) \times 100$ )が短縮されたことになる。No.6のnewとは、RIAM-COMPACTをz\*座標系に変更した場合の結果である。これ以降についても、全て同じ意味である。No.5とNo.6を比較すると、約1.15倍(=No.5/No.6)の速度向上率が得られている。これはz\*座標系に変更したことで、解くべき座標変換後の支配方程式が大幅に簡略化され、これに伴い計算負荷が小さくなったためである。結果として、No.3からNo.6への計算速度の向上率は約1.3倍(=No.3/No.6)である。

| Machine name      | Compiler                                  | Compiler option                                  | CPU time (s)                       | CPU time ratio |
|-------------------|---|--|------------------------------------|----------------|
| <b>M-01 No.1</b>  | ifl<br>Intel Fortran<br>7.0               | nothing  | 2667.844<br>(interactive)          | 8.39           |
| <b>M-01 No.2</b>  | ifl<br>Intel Fortran<br>7.0               | /O3  | 2642.625<br>(interactive)          | 8.31           |
| <b>M-01 No.3</b>  | ifl<br>Intel Fortran<br>7.1               | nothing  | 2692.063<br>(interactive)          | 8.46           |
| <b>M-01 No.4</b>  | ifl<br>Intel Fortran<br>7.1               | /O3  | 2696.875<br>(interactive)          | 8.48           |
| <b>M-01 No.5</b>  | ifl<br>Intel Fortran<br>7.1               | /O3<br>/G7<br>/QaxW<br>/QxW                      | 2384.094<br>(interactive)          | 7.50           |
| <b>M-01 No.6</b>  | ifl<br>Intel Fortran<br>7.1               | /O3<br>/G7<br>/QaxW<br>/QxW                      | 2070.438<br>(interactive)<br>(new) | 6.51           |
| <b>M-01 No.7</b>  | f90<br>DIGITAL<br>Visual Fortran<br>5.0   | nothing  | 3394.859<br>(interactive)          | 10.67          |
| <b>M-01 No.8</b>  | f90<br>Compaq<br>Visual Fortran<br>6.1    | nothing  | 3338.969<br>(interactive)          | 10.50          |
| <b>M-01 No.9</b>  | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 2818.641<br>(interactive)          | 8.86           |
| <b>M-01 No.10</b> | f90<br>Compaq<br>Visual Fortran<br>6.6    | nothing  | 3299.641<br>(interactive)          | 10.37          |
| <b>M-01 No.11</b> | f90<br>Compaq<br>Visual Fortran<br>6.6    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 2798.156<br>(interactive)          | 8.80           |
| <b>M-01 No.12</b> | f90<br>Compaq<br>Visual Fortran<br>6.6    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 2397.969<br>(interactive)<br>(new) | 7.54           |
| <b>M-01 No.13</b> | f77<br>Compaq<br>Visual<br>Fortran<br>6.6 | nothing  | 3309.703<br>(interactive)<br>)     | 10.41          |
| <b>M-02 No.14</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 2375.609<br>(interactive)          | 7.47           |
| <b>M-03 No.15</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 3026.609<br>(interactive)          | 9.52           |
| <b>M-04 No.16</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 5606.622<br>(interactive)          | 17.63          |
| <b>M-05 No.17</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 9162.061<br>(interactive)          | 28.81          |
| <b>M-06 No.18</b> | pgf90<br>PGI Fortran<br>3.2-3             | nothing  | 8400.0<br>(interactive)            | 26.41          |
| <b>M-06 No.19</b> | pgf90<br>PGI Fortran<br>3.2-3             | -fast  | 7860.0<br>(interactive)            | 24.71          |
| <b>M-06 No.20</b> | pgf90<br>PGI Fortran<br>3.2-3             | -fast<br>-Mcache_align<br>-Mvect=sse             | 7140.0<br>(interactive)            | 22.45          |
| <b>M-07 No.21</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | nothing  | 3908.422<br>(interactive)          | 12.29          |
| <b>M-07 No.22</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 3343.906<br>(interactive)          | 10.51          |
| <b>M-08 No.23</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 3817.328<br>(interactive)          | 12.00          |
| <b>M-09 No.24</b> | f90<br>Compaq<br>Visual Fortran<br>6.1    | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 7140.0<br>(interactive)            | 22.45          |
| <b>M-10 No.25</b> | f90<br>Compaq<br>Visual<br>Fortran<br>6.1 | /arch:host<br>/tune:host<br>/fast<br>/optimize:4 | 9543.695<br>(interactive)<br>)     | 30.01          |

Table.6 Numerical results of our laboratory

M-01で、Compaq Visual Fortranコンパイラを用いた結果(No.7~No.13)に注目する。No.7(v.5.0)→No.8(v.6.1)→No.10(v.6.6)とそのバージョンが上がるにつれてCPU時間が徐々に短縮されている。最適化オプションが有る場合のNo.11(v.6.6)では、それが無い場合のNo.10(v.6.6)に比べて約1.18倍(=No.10/No.11)の速度向上率が得られている。さらに、z\*座標系に変更することで約1.17(=No.10/No.11)の速度向上率が得られている。この値はIntel Fortranコンパイラとほぼ同じである。結果として、No.10(v.6.6)からNo.12(v.6.6)への計算速度の向上率は約1.38倍(=No.10/No.12)であった。この値もIntel Fortranコンパイラとほぼ同じである。Visual

Fortranコンパイラのv.6.6では、f90(No.10)とf77(No.13)の違いはほとんど見られなかった。ここで特筆すべきは、Fortranコンパイラの違いによるCPU時間の差異である。Intel Pentium 4搭載の機種においては、当然と思われるがIntel Fortranコンパイラの方がCompaq Visual Fortranコンパイラよりも計算速度が速いということが示された。Intel Fortranコンパイラを用いたNo.6と、Compaq Visual Fortranコンパイラを用いたNo.12を比較すると、その速度比は約1.16(=No.12/No.6)であった。すなわち、No.12の計算に要したCPU時間を基準にすると、その約14%(=(1-No.6/No.12)×100)が短縮されたことになる。

M-08(VT社のAlphaマシン)では、Table.6には示さな

| Machine name                | Compiler                     | Compiler option                          | CPU time (s)                                 | CPU time ratio | Machine name                | Compiler                      | Compiler option  | CPU time (s)                                      | CPU time ratio |
|-----------------------------|------------------------------|--|--|----------------|-----------------------------|-------------------------------|--|---|----------------|
| <b>M-11</b><br><b>No.26</b> | f90 or frt90<br>Fortran95/VP | nothing                                  | 323.15<br>(B0 or B1 queue)                   | 1.02           | <b>M-12</b><br><b>No.40</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-tune ev6   | 1960.0<br>(S queue)<br>(2CPU)<br>(198%)           | 6.16           |
| <b>M-11</b><br><b>No.27</b> | f90 or frt90<br>Fortran95/VP | -Oe                                      | 321.01<br>(B0 or B1 queue)                   | 1.01           | <b>M-12</b><br><b>No.41</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-tune ev6   | 1603.0<br>(S queue)<br>(2CPU)<br>(198%)<br>(new)  | 5.04           |
| <b>M-11</b><br><b>No.28</b> | f90 or frt90<br>Fortran95/VP | -Kfast                                   | 318.07<br>(B0 or B1 queue)                   | 1<br>standard  | <b>M-12</b><br><b>No.42</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-arch host<br>-tune host<br>-fast -O5<br>(new)    | 1620.02<br>(S queue)<br>(2CPU)<br>(195%)<br>(new) | 5.09           |
| <b>M-11</b><br><b>No.29</b> | f90 or frt90<br>Fortran95/VP | -Kfast                                   | 293.32<br>(B0 or B1 queue)<br>(new)          | 0.92           | <b>M-12</b><br><b>No.43</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-tune ev6   | 1200.05<br>(S queue)<br>(4CPU)<br>(394%)          | 3.77           |
| <b>M-11</b><br><b>No.30</b> | f90 or frt90<br>Fortran95/VP | -O5<br>-KA32                             | 318.37<br>(B0 or B1 queue)                   | 1.00           | <b>M-12</b><br><b>No.44</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'  | 1011.0<br>(S queue)<br>(4CPU)<br>(392%)<br>(new)  | 3.18           |
| <b>M-11</b><br><b>No.31</b> | f90 or frt90<br>Fortran95/VP | -Kfast                                   | 587.56<br>(B0 or B1 queue)<br>(Scalar SOR)   | 1.85           | <b>M-12</b><br><b>No.45</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-tune ev6   | 1009.0<br>(S queue)<br>(4CPU)<br>(393%)<br>(new)  | 3.17           |
| <b>M-11</b><br><b>No.32</b> | f90 or frt90<br>Fortran95/VP | -Wv,-sc                                  | 7509.17<br>(B0 or B1 queue)<br>(Scalar mode) | 23.61          | <b>M-12</b><br><b>No.46</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-arch ev6<br>-tune ev6<br>-O5<br>(new)            | 1006.0<br>(S queue)<br>(4CPU)<br>(393%)<br>(new)  | 3.16           |
| <b>M-12</b><br><b>No.33</b> | f90<br>Compaq<br>Fortran     | nothing                                  | 3492.686<br>(E queue)<br>(99%)               | 10.98          | <b>M-12</b><br><b>No.47</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-arch ev6<br>-tune ev6<br>-fast<br>-O<br>(new)    | 1019.0<br>(S queue)<br>(4CPU)<br>(393%)<br>(new)  | 3.20           |
| <b>M-12</b><br><b>No.34</b> | f90<br>Compaq<br>Fortran     | -fast                                    | 3174.305<br>(E queue)<br>(99%)               | 9.98           | <b>M-12</b><br><b>No.48</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-arch host<br>-tune host<br>-fast<br>-O<br>(new)  | 1016.0<br>(S queue)<br>(4CPU)<br>(393%)<br>(new)  | 3.19           |
| <b>M-12</b><br><b>No.35</b> | f90<br>Compaq<br>Fortran     | -arch ev6<br>-tune ev6<br>-fast<br>-O    | 3163.536<br>(E queue)<br>(99%)               | 9.95           | <b>M-12</b><br><b>No.49</b> | kf90<br>Compaq KAP<br>Fortran | -fkapargs=<br>'-concurrent'<br>-arch host<br>-tune host<br>-fast<br>-O5<br>(new) | 1020.0<br>(S queue)<br>(4CPU)<br>(392%)<br>(new)  | 3.21           |
| <b>M-12</b><br><b>No.36</b> | f90<br>Compaq<br>Fortran     | -arch ev6<br>-tune ev6<br>-fast<br>-O    | 2584.0<br>(E queue)<br>(99%)<br>(new)        | 8.12           |                             |                               |  |   |                |
| <b>M-12</b><br><b>No.37</b> | f90<br>Compaq<br>Fortran     | -arch ev6<br>-tune ev6<br>-O5            | 2938.0<br>(E queue)<br>(99%)<br>(new)        | 9.24           |                             |                               |  |   |                |
| <b>M-12</b><br><b>No.38</b> | f90<br>Compaq<br>Fortran     | -arch ev6<br>-tune ev6<br>-fast<br>-O5   | 2639.0<br>(E queue)<br>(99%)<br>(new)        | 8.30           |                             |                               |  |   |                |
| <b>M-12</b><br><b>No.39</b> | f90<br>Compaq<br>Fortran     | -arch host<br>-tune host<br>-fast<br>-O5 | 2592.0<br>(E queue)<br>(99%)<br>(new)        | 8.15           |                             |                               |  |   |                |

Table.7 Numerical results of the computer room of RIAM

かったが、計算の進行状態を示す数値をディスプレイ、あるいは、ファイルに出力することが全体の計算時間にどの程度影響するかを検討した。通常、時間積分の数値シミュレーションでは、時間、ステップ数、レイノルズ数、圧力のポアソン方程式のr.m.s値など、計算の進行状況を示す数値をディスプレイ、あるいは、ファイルに出力する。5,000ステップの時間積分で100回おきに計算の進行状況をディスプレイ、あるいは、ファイルに出力した場合は、その差はほとんど見られなかった。ファイルに出力する場合、その間隔を100回とした場合と、5,000回(最後に1回出力)とした場合においても両者の違いはなかった。ディスプレイに出力する場合、その間隔を100回とした場合と、1回にした場合では後者の方が前者よりも20(s)程度CPU時間が延びた。

Table.6に示す計算結果を概観すると、ここ数年間で

PCの計算機性能は大幅に向上しているのが見て取れる。しかしながら、スーパーコンピュータ1PE(Table.7に示すM-11, No.28)との差は依然として大きい。

## 4.2 応力研所有の計算機環境について

ここでは、応用力学研究所所有の計算機環境(Table.2)で得られた結果(Table.7)について考察する。これらの計算は全てバッチ型処理(batch processing)で行った。バッチ型処理とは、対話的に行う一連の処理(コマンド)の流れをバッチ・リクエスト(batch request)と呼ばれるシェル・スクリプト(shell script)にエディタ(editor)を使って記述し、そのファイル(シェル・スクリプト)をバッチ・キュー(batch queue)に投入して計算を行うことである。ジョブ(job)はその種類やサイズなどに応じてバッチ・キュー

一名(投入先)を指定する。VPP5000(M-11)およびES40(M-12)ともに、ジョブの管理はNQS(Network Queueing System)と呼ばれるバッチ制御システムで行われている。なお、バッチ・リクエストの依頼はqsubコマンド、処理状況の確認はqstatコマンド、ジョブのキャンセルはqdelコマンドで行う。M-11とM-12における計算は、両者ともにPCからアプリケーション・サーバ(application server)にTELNETでログインし、そこからバッチ・リクエストを投入して行った。

M-11において、実行可能オブジェクト(executable object)は以下の二つの方法で生成した。一つは、アプリケーション・サーバ上でクロスコンパイラコマンドfirtpxを用いて対話的に生成した。もう一つは、firtコマンドを用いて記述したCシェル・スクリプトをアプリケーション・サーバからコンパイル専用のLキューに投入して生成した。M-11のNo.26~No.30に示す逐次計算では、自動ベクトル化機能を有するFortran95/VP(Vector Processor)を用いた。ベクトル処理(vector processing)については後述する。それらの結果に注目すると、最適化オプションの効果はそれほど顕著に現れていない。これは、Fortran95/VPが富士通製品を対象に開発されたFortranコンパイラ(1PE用)であるので、最適化オプションを付けない場合においてもかなりの高速化が実現されていることを意味している。M-11のNo.31は、圧力のポアソン方程式の緩和計算をベクトル計算機用の修正SOR法(No.26~No.30)ではなく、回帰演算を含む通常のSOR法のままで計算した結果である。つまり、圧力のポアソン方程式を緩和計算するDOループはベクトル化されていない。No.28と比較すると1.85倍の計算時間を要している。しかしながら、この値はVPP5000以前に応用力学研究所に導入されていた高速演算サーバVX(富士通株式会社)と比較すると、大幅に短縮されたように感じる。

以下では、ベクトル計算の目安となるベクトル化率 $\alpha$ とベクトル加速率 $\beta$ について考察する。この目的に対し、コンパイル時のオプションとして-Wv,-scを付け、スカラーモードでの計算を実行した(Table.7, No.32)。これにより全てのDOループのベクトル化が抑制される。Table.7に示すNo.28とNo.32のtimex情報をそれぞれTable.8およびTable.9に示し、これを用いて $\alpha$ および $\beta$ を評価する。Table.8およびTable.9において、realは経過時間(elapsed time)、userはユーザーのCPU使用時間(=ベクトルユニットとスカラーユニットの合計時間)、sysはシステムのCPU使用時間(=ベクトルユニットとスカラーユニットの合計時間)、vu-userはユーザーCPU時間の中でベクトルユニットが動作した時間、vu-sysはシステムCPU時間の中でベクトルユニットが動作した時間である。便宜上、Table.8およびTable.9の各時間の名称をそれぞれのTable中に記入する。

|                      |                      |
|----------------------|----------------------|
| <b>real (Ta)</b>     | 5:23.15 (323.15 (s)) |
| <b>user (Ta1)</b>    | 5:13.75 (313.75 (s)) |
| <b>sys (Ta2)</b>     | 0.29 (0.29 (s))      |
| <b>vu-user (Ta3)</b> | 4:52.03 (292.03 (s)) |
| <b>vu-sys (Ta4)</b>  | 0.00 (0.00 (s))      |

**Table.8 Output result using the timex command for the case of No.28, Vector mode**

|               |                          |
|---------------|--------------------------|
| real (Tb)     | 2:05:09.17 (7509.17 (s)) |
| user (Tb1)    | 2:03:13.48 (7393.48 (s)) |
| sys (Tb2)     | 3.92 (3.92 (s))          |
| vu-user (Tb3) | 0.05 (0.05 (s))          |
| vu-sys (Tb4)  | 0.00 (0.00 (s))          |

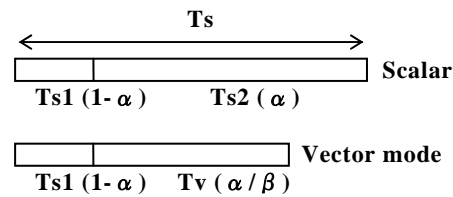
**Table.9 Output result using the timex command for the case of No.32, Scalar mode**

$\alpha$  および  $\beta$  はFig.2を参照して、以下に示す(3)式および(4)式で定義される。但し、(3)式および(4)式ではI/O時間は考慮していない。実際に計算すると、以下に示す値を得る。

$$\alpha = Ts2 \div Ts = [(Tb1+Tb2) - \{(Ta1+Ta2) - (Ta3+Ta4)\}] / (Tb1+Tb2) \quad (3)$$

$$\beta = Ts2 \div Tv = [(Tb1+Tb2) - \{(Ta1+Ta2) - (Ta3+Ta4)\}] / (Ta3+Ta4) \quad (4)$$

$$\alpha \approx 0.997, \quad \beta \approx 25.26 \quad (5)$$



$Ts = Tb1+Tb2$   
: All calculation time in scalar mode

$Ts1 = (Ta1+Ta2) - (Ta3+Ta4)$   
: Scalar calculation time of the portion which cannot be vectorized

$Ts2 = Ts - Ts1$   
 $= (Tb1+Tb2) - \{(Ta1+Ta2) - (Ta3+Ta4)\}$   
: Scalar calculation time of the portion which can be vectorized

$Tv = Ta3+Ta4$   
: Vector calculation time of the portion which can be vectorized

**Fig.2 Schematic view of the calculation time**

ベクトル化率  $\alpha$  を簡単に概算するには、ベクトルモードで実行した場合 (Table.8) において、全CPU時間 ( $=Ta1+Ta2$ ) に対するベクトルユニットの占有時間 ( $=Ta3+Ta4$ ) の比

$$(Ta3+Ta4)/(Ta1+Ta2) \quad (6)$$

を求める方法もある。これをVU率と言う。VU率は

$$(Ta3+Ta4)/(Ta1+Ta2) \approx 0.93 \quad (7)$$

となる。(5)式および(7)式の値から、RIAM-COMPACTはベクトル計算にも優れた性能を有することが示された。

ここで、ベクトル計算(処理)について簡単に説明する<sup>2)</sup>。まず、スカラーデータとは変数や配列の要素などの一つのデータのことである。一方、ベクトルデータとは配列全体や配列の部分など複数のスカラーデータから構成されているものである。ベクトル処理とは、DOループの演算に対し、配列をベクトルデータとして処理する方法である。具体的には、1.データをベクトルとして連続的にメモリから取り出し、2.同一の演算をベクトルデータに対して連続的に実行し、3.結果を再度連続的にメモリに書き込むというものである。実際にどの程度高速化できるかというのは、ベクトル化率  $\alpha$ 、ベクトル加速率  $\beta$ 、実行性能向上比  $E=(\text{スカラーモードでの実行時間}(Tb1+Tb2)/\text{ベクトルモードでの実行時間}(Ta1+Ta2))$ 、RIAM-COMPACTの場合には  $E \approx 23.56$  の関係を表したアムダールの法則(Amdahl's Law)で公式化されている。これによると、1 PEで十分なベクトルチューニングを行うためには、ベクトル化率  $\alpha$  をできるだけ1に近づけ、言い換えると、ベクトル化可能な部分をできる限り増やし、その後、ベクトル

加速率  $\beta$  を向上させる必要がある。また、ベクトル化の効率を上げるためには、1.大規模な配列を用いてベクトルデータの要素数(DOループの回転数NX, IMAXなど)を増やす、すなわち、ベクトル長を伸ばすこと、2.連続的にメモリにアクセスするように列方向にデータを参照することなどが挙げられる。前者では、ベクトルデータの要素数が多いほど、一括して処理できる配列データの数が増加し、処理効率が向上する。これと併せてDOループによる演算が多い場合も処理効率が上がる。なお、ベクトル長が極端に短い場合にはスカラーモードでの実行の方が速いということがある。後者の場合では、具体的に以下のようにコーディングする。例えば、Fig.3に示す2次元配列の添え字を二重のDOループで動かす場合には、内側のDOループで第一の添え字が動き、外側のDOループで第二の添え字が動くようにする。3次元配列の場合でも同様である。

再度、Table.7に注目する。M-12のスカラー逐次計算の結果をNo.33~No.39に、自動並列化機能を利用したDual CPU(2スレッド)のスカラー並列計算の結果をNo.40~No.42に、同様に4CPU(4スレッド)のスカラー並列計算の結果をNo.43~No.49に示す。スカラー逐次計算(No.33~No.39)は、Compaq Fortranでコンパイルを行い、Eキューで実行した。その結果をVPP5000(M-11)のベクトル逐次計算の結果(Table.7, No.28)と比較する。最適化オプションなどの効果は明確に認められるものの、約8~11倍の計算時間を要しており、その差はかなり大きい。No.40~No.49のスカラー並列計算は、Compaq KAP Fortranを用い、Sキューで実行した。コンパイルオプションの-fkapargs='-concurrent'がKAP Fortranで解釈されるスイッチとなる。これに伴い、プログラム中の並列化可能なDOループが自動検出され、C\$OMP...などのOpenMP指示行(directive)が挿入される。OpenMPとは、共有メモリ型マルチプロセッサMP(Multi Processor)向けの並列プログラミングの標準API(Application Program Interface)である。その結果として、SMPシステム上で動作する実行可能オブジェクト(実行バイナリ)が生成される。他のオプションは逐次計算と同様、f90に引き渡される。-fkapargs='-concurrent -scaleropt=0'とすれば、自動並列化のみを施したプログラム(拡張子cmp)が出力される。これを足掛かりにさらなる改良も可能である。併せて、並列最適化レポートファイル(拡張子out)も出力される。自動並列計算において、計算に参加するスレッド数は環境変数OMP\_NUM\_THREADSにより指定する。例えば、スレッド数を4に指定する際には以下のようになる。

For the B shell (/bin/sh, bash) :

```
export OMP_NUM_THREADS=4
```

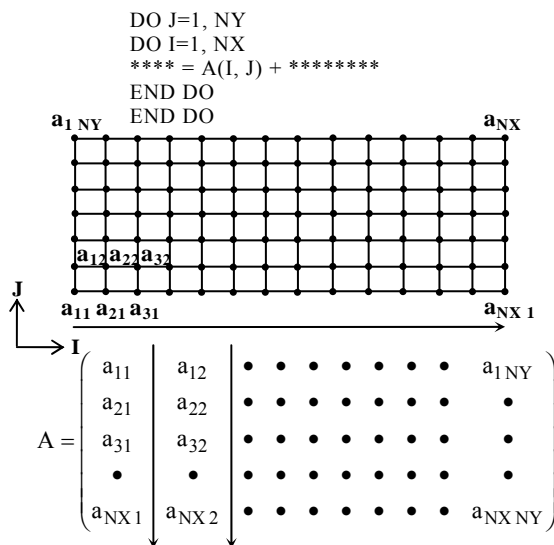


Fig.3 Conceptual figure of the order of calculation



For the C shell (/bin/csh, tcsh) :

```
setenv OMP_NUM_THREADS 4
```

(8)

さて, No.39(1CPU), No.42(2CPU), No.49(4CPU)の結果に注目する. CPU時間の括弧の中の数字はプログラムの実行に要した実時間に対して, 実際にCPUを使用した時間の比率(%)を示している. 1CPU分につき100%なので, 2CPU(2スレッド)がフル稼働した場合には200%, 4CPU(4スレッド)がフル稼働した場合には400%になる. これについては後述する. 各ケースともにCPUをほぼ完全に使用している. スケーラビリティ(scalability)は,

$$\begin{aligned} \text{No.39(1CPU, 1スレッド)/No.42(2CPU, 2スレッド)} &\approx 1.60 \\ \text{No.39(1CPU, 1スレッド)/No.49(4CPU, 4スレッド)} &\approx 2.54 \end{aligned}$$

(9)

となり, RIAM-COMPACTに対する自動並列化機能の有効性が確認された. スケーラビリティとは, 1CPU(1スレッド)で実行した場合に比べ, マルチプロセッサで実行した際に性能がどれくらい向上するかを示す指標である. この値が大きいほど優れた並列効率を有することになる. 結果として, 4CPU(4スレッド)のスカラ並列計算では, スーパーコンピュータ1PEの約3倍に迫る結果を得た.

### 4.3 情報基盤センターの計算機環境について

ここでは, 情報基盤センター所有の計算機環境(Table.3)で得られた結果(Table.10)について考察する.

M-13のベクトル並列計算機(VPP5000/64)の結果(No.50~No.56)に注目する. No.50~No.56は全てNQSによるバッチ処理で行った. No.50はFortran95/VPによる逐次計算の結果である. No.51~No.56はFortran95/VPP(Vector Parallel Processor or Vector Pipeline Processor)を用いたベクトル並列計算の結果である. Fortran95/VPPは富士通のVPPシリーズでサポートされている複数PE用のベクトル並列Fortranコンパイラである. 現在では, 単にVPP Fortranと呼ばれている. VPP Fortranは逐次のFortranプログラムに!XOCL...を挿入して並列プログラムを作成する. 拡張最適化制御行 XOCL(eXtended Optimization Control Line)はFortran95/VPP独自の用語である. コンパイルオプションとして-Wxを指定することでVPP Fortranコンパイラが起動し, 逐次プログラムに挿入した!XOCL...が有効になる. -Kfastオプションは並列, 非並列に関わらず使用可能で, これは各PEで実行される部分, つまり, 通信を行わない部分を高速化する. RIAM-COMPACTでは, I/O処理やメトリック(metric)の計算は1台のPEを使ってプログラムを実行する逐次処理(serial processing)ではなく, 全てのPEで同一処理を実行する冗長処理

(redundant processing)とした. その他の部分は複数のPEで処理を分担し, 並列処理(parallel processing)とした. No.50に示すベクトル逐次計算の結果では, 情報基盤センター所有のVPP5000(Table.10, No.50)の方が, 応力研所有のVPP5000(Table.7, No.28)よりも若干良い結果を示すことが分かった. ベクトル並列計算におけるスケーラビリティは以下の通りである.

$$\begin{aligned} \text{No.51(1PE)/No.52(2PE)} &\approx 1.68 \\ \text{No.51(1PE)/No.53(4PE)} &\approx 2.45 \\ \text{No.51(1PE)/No.54(8PE)} &\approx 3.33 \\ \text{No.51(1PE)/No.55(16PE)} &\approx 3.81 \\ \text{No.51(1PE)/No.56(32PE)} &\approx 4.02 \end{aligned}$$

(10)

M-14のスカラ並列サーバGS320による計算結果(No.57~No.62)に注目する. これらの計算は全てGS320にTELNETでログインした後, 対話的に実行した. スカラ逐次計算(No.57)は, Compaq Fortranでコンパイルを行い実行した. Dual CPU(2スレッド)のスカラ並列計算(No.58), 4CPU(4スレッド)のスカラ並列計算(No.59~No.61), 8CPU(8スレッド)のスカラ並列計算(No.62)は, Compaq KAP Fortranの自動並列化機能を利用して実行した. スカラ並列計算において, それぞれのスケーラビリティは以下の通りである.

$$\begin{aligned} \text{No.57(1CPU, 1スレッド)/No.58(2CPU, 2スレッド)} &\approx 1.81 \\ \text{No.57(1CPU, 1スレッド)/No.61(4CPU, 4スレッド)} &\approx 2.72 \\ \text{No.57(1CPU, 1スレッド)/No.62(8CPU, 8スレッド)} &\approx 2.22 \end{aligned}$$

(11)

No.58(2CPU, 2スレッド)およびNo.61(4CPU, 4スレッド)に示すスカラ並列計算では, 並列化の効果は明確に現れている. しかしながら, No.62(8CPU, 8スレッド)の並列計算においては, No.61よりも悪い結果を示している. これはQBB内とそれをまたぐ場合では, メモリ・アクセス速度が異なることに起因している. QBB内にあるメモリにアクセスするときの速度(ローカル速度)は6GB/sを超える. しかしながら, QBBをまたいで別のQBB内のメモリにアクセスする速度(リモート速度)は約1.6GB/sである. すなわち, リモート速度はローカル速度の1/3弱である. このメモリアクセスの速度差が大きな性能低下の原因であると予想される. また, ここで特筆すべきはM-12のES40との比較である. GS320はES40の上位機種であるにも関わらず, 計算速度はGS320よりもES40の方が速いという結果を示した. この原因は以下のことが考えられる. GS320はCPU/クロック速度はAlpha 21264 B(EV68)/731MHzで2次キャッシュに4MBを搭載している. 一方, ES40はCPU/クロック速度はAlpha 21264 A(EV67)/667MHzで2次キャッシュに8MBを搭載している. GS320はES40に対して

| Machine name      | Compiler                              | Compiler option  | CPU time (s)   | CPU time ratio |
|-------------------|---------------------------------------|--|--|----------------|
| <b>M-13 No.50</b> | f90<br>Fortran95/VP                   | -Kfast   | 311.70<br>(s queue)<br>(1PE)                         | 0.98           |
| <b>M-13 No.51</b> | f90<br>VPP Fortran<br>(Fortran95/VPP) | -Wx<br>-Kfast  | 335.93<br>(x16 queue)<br>(1PE)                       | 1.06           |
| <b>M-13 No.52</b> | f90<br>VPP Fortran<br>(Fortran95/VPP) | -Wx<br>-Kfast  | 199.77<br>(x16 queue)<br>(2PE)                       | 0.63           |
| <b>M-13 No.53</b> | f90<br>VPP Fortran<br>(Fortran95/VPP) | -Wx<br>-Kfast  | 136.95<br>(x16 queue)<br>(4PE)                       | 0.43           |
| <b>M-13 No.54</b> | f90<br>VPP Fortran<br>(Fortran95/VPP) | -Wx<br>-Kfast  | 101.02<br>(x16 queue)<br>(8PE)                       | 0.32           |
| <b>M-13 No.55</b> | f90<br>VPP Fortran<br>(Fortran95/VPP) | -Wx<br>-Kfast  | 88.09<br>(x16 queue)<br>(16PE)                       | 0.28           |
| <b>M-13 No.56</b> | f90<br>VPP Fortran<br>(Fortran95/VPP) | -Wx<br>-Kfast  | 83.53<br>(p32 queue)<br>(32PE)                       | 0.26           |
| <b>M-14 No.57</b> | f90<br>Compaq Fortran                 | -arch ev6<br>-tune ev6<br>-O5  | 3990.0<br>(interactive)<br>(1CPU)<br>(99%)<br>(new)  | 12.54          |
| <b>M-14 No.58</b> | kf90<br>Compaq KAP<br>Fortran         | -fkapargs=<br>'-concurrent'<br>-arch ev6<br>-tune ev6<br>-O5 -fast   | 2200.0<br>(interactive)<br>(2CPU)<br>(194%)<br>(new) | 6.92           |
| <b>M-14 No.59</b> | kf90<br>Compaq KAP<br>Fortran         | -fkapargs=<br>'-concurrent'<br>-arch ev6<br>-tune ev6<br>-O5         | 1495.0<br>(interactive)<br>(4CPU)<br>(380%)<br>(new) | 4.70           |
| <b>M-14 No.60</b> | kf90<br>Compaq KAP<br>Fortran         | -fkapargs=<br>'-concurrent'<br>-arch ev6<br>-tune ev6<br>-O5 -fast   | 1465.0<br>(interactive)<br>(4CPU)<br>(382%)<br>(new) | 4.61           |
| <b>M-14 No.61</b> | kf90<br>Compaq KAP<br>Fortran         | -fkapargs=<br>'-concurrent'<br>-arch host<br>-tune host<br>-O5 -fast | 1480.0<br>(interactive)<br>(4CPU)<br>(379%)<br>(new) | 4.65           |
| <b>M-14 No.62</b> | kf90<br>Compaq KAP<br>Fortran         | -fkapargs=<br>'-concurrent'<br>-arch host<br>-tune host<br>-O5 -fast | 1795.0<br>(interactive)<br>(8CPU)<br>(722%)<br>(new) | 5.64           |
| <b>M-15 No.63</b> | f90<br>Fortran95                      | -Kparallel,<br>reduction,<br>instance=1,<br>fast_eval                | 9905.45<br>(sc32 queue)<br>(1CPU)<br>(new)           | 31.14          |
| <b>M-15 No.64</b> | f90<br>Fortran95                      | -Kparallel,<br>reduction,<br>instance=2,<br>fast_eval                | 5277.40<br>(sc32 queue)<br>(2CPU)<br>(new)           | 16.59          |
| <b>M-15 No.65</b> | f90<br>Fortran95                      | -Kparallel,<br>reduction,<br>instance=4,<br>fast_eval                | 3070.46<br>(sc32 queue)<br>(4CPU)<br>(new)           | 9.65           |
| <b>M-15 No.66</b> | f90<br>Fortran95                      | -Kparallel,<br>reduction,<br>instance=8,<br>fast_eval                | 2013.76<br>(sc32 queue)<br>(8CPU)<br>(new)           | 6.33           |
| <b>M-15 No.67</b> | f90<br>Fortran95                      | -Kparallel,<br>reduction,<br>instance=16,<br>fast_eval               | 1530.83<br>(sc32 queue)<br>(16CPU)<br>(new)          | 4.81           |
| <b>M-15 No.68</b> | f90<br>Fortran95                      | -Kparallel,<br>reduction,<br>instance=32,<br>fast_eval               | 1236.77<br>(sc32 queue)<br>(32CPU)<br>(new)          | 3.89           |

Table.10 Numerical results of the computing and communications center of Kyushu University

クロック速度の部分は高速であるが、2次キャッシュの部分に約2倍のハンディを負っている。これはスカラー型の計算機においてコンパイラによる最適化の目的の一つである2次キャッシュ領域の有効活用(キャッシュ・ヒット・ミスの低減)において大きなハンディになり得る。おそらく、RIAM-COMPACTはキャッシュに対する最適化がよい方向に進むタイプのアプリケーションであり、キャッシュのサイズが顕著に性能に反映されたものであると考えられる。

M-15の汎用UNIXサーバのGP7000Fモデル900による計算結果(No.63~No.68)に注目する。自動並列化機能を利用し、1~32CPUまでのスカラー並列計算を行った。コンパイルオプションの-Kparallelが自動並列化のスイッチとなり、その他のオプションはf90に引き渡される。なお、-Kで始まるオプションはカンマで区切って続けて指定できる。GP7000Fの自動並列化機能は、ソースプログラムを出力するタイプではなく、富士通専用のライブラリ(\_mpc\_initとか\_mpc\_pend)が呼び出される。使用するCPUの数は環境変数PARALLELで設定する。それぞれのスケラビリティは以下の通りである。

$$\begin{aligned} \text{No.63(1CPU)/No.64(2CPU)} &\div 1.88 \\ \text{No.63(1CPU)/No.65(4CPU)} &\div 3.23 \\ \text{No.63(1CPU)/No.66(8CPU)} &\div 4.92 \\ \text{No.63(1CPU)/No.67(16CPU)} &\div 6.47 \\ \text{No.63(1CPU)/No.68(32CPU)} &\div 8.01 \end{aligned}$$

(12)

この結果は、これまでの機種に比べて良い結果を示しており、自動並列化機能の有効性は確認された。しかしながら、スーパーコンピュータ1PE(Table.7のM-11, No.28)との差は依然として大きいと言える。

#### 4.4 V-T社の計算機環境について

ここでは、ビジュアルテクノロジー株式会社所有の計算機環境(Table.4)で得られた結果(Table.11)について考察する。AMD社の新しい64BitマイクロプロセッサであるOpteronを搭載した機種において、スカラー逐次計算および自動並列化機能を利用したDual CPU(2スレッド)のスカラー並列計算を評価していただいた。PGI Fortranコンパイラでは、f77(pgf77)とf90(pgf90)について、64Bit版と32Bit版のテストを行って頂いた。また、g77は64Bitモード(デフォルト)と32Bitモード(-m32)の双方についてテストを行って頂いた。

M-16の結果に注目する。Intel Fortranコンパイラ(No.69)が最も良い結果を与え、単体性能としてはスーパーコンピュータ1PEの約4倍に迫る結果を得た。以下、PGIの64Bit版(No.70およびNo.71)、PGIの32Bit版

(No.72およびNo.73), GNUの64Bit版(No.74), GNUの32Bit版(No.75)の順となった. Intel Fortranコンパイラの自動並列化機能を利用したDual CPU(2スレッド)のスカラー並列計算(No.76およびNo.77)では, 逐次実行よりもわずかに良い結果を与えているが, 並列化効率はそれほど良くない. ここでは, -parallelオプションが自動並列のスイッチになり, その他のオプションは性能向上に寄与するものである. 使用するCPU(スレッド)の数は, 環境変数OMP\_NUM\_THREADSで指定する.

M-17の結果に注目する. 1CPU(1スレッド)での結果(No.78およびNo.79)はクロック速度の増加(1.6GHz→

1.8GHz)に伴い, 若干向上している. しかしながら, Dual CPU(2スレッド)での実行時(-Mconcurオプション使用時)では, わずかではあるが1CPU(1スレッド)よりも計算速度が遅くなった. なお, 以下の点に注意が必要である. 環境変数OMP\_NUM\_THREADSが未設定時の場合, Intel Fortranコンパイラでは全てのCPU(スレッド)が稼働するが, PGI Fortranコンパイラでは単一CPU(スレッド)での動作となる.

#### 4.5 HIT社の計算機環境について

ここでは, 株式会社エッチ・アイ・ティー所有のOpen-SCCの計算機環境(Table.5)で得られた結果(Table.12)について考察する.

Open-SCCの運用形態は以下の通りである. まず, SSHによりゲートウェイマシン(gateway machine)にログインし, そこからTELNETでsccへ移る. sccがプログラムのコンパイルおよびジョブ投入ホストである. sccにIntel Fortranコンパイラ7.1およびPGI Fortranコンパイラ4.02が実装(implement)されている. M-18~M-20, M-23~M-25についてはsccでコンパイルを行い, その後, LSF(Load Sharing Facility)というバッチ処理ソフト(ジョブ管理ソフト)により各キューにジョブを投入して計算を行った. バッチ・リクエストの依頼はbsubコマンド, 処理状況の確認はbjobs, bqueues, qstatコマンド, ジョブのキャンセルはbkillコマンドで行う. それ以外のホストについては, バッチ運用ではなく, sccからその計算機に直接TELNETでログインし, 対話的に計算を実行した.

No.82~No.87(M-18~M-20)では, Intel Pentium 4に特化したオプションを含め, 最適化オプション(コンパイルオプション)の効果は顕著に現れている. スーパーコンピュータ1PE(Table.7に示すM-11, No.28)と比較すると, その速度差は4倍弱にまで短縮されており, PCの単体性能は著しい向上を示している. M-21の結果に注目する. これは, Pentium 4機の最新チップセット(i875PE)搭載マシンである. 性能上の大きな改良点として, FSB(Front Side Bus)のクロック周波数(clock frequency)が533MHzから800MHzに向上したこと, DDR400の2wayメモリになったことでメモリバンド幅がFSBクロックと同期し理論値6.4GB/sになったことが挙げられる. FSBクロック, メモリアクセス性能の向上はアプリケーション実行のパフォーマンスに大きく貢献する. このマシンは現時点のIA-32環境で最高のパフォーマンスを有するものである. No.89およびNo.90では, スーパーコンピュータ1PE(Table.7に示すM-11, No.28)の3倍弱に迫る結果を得た. また注目すべきは, Fortranコンパイラの違いである. Intel Fortranコンパイラ7.1とPGI Fortranコンパイラ4.02(No.91~No.96)では, やはり前者の方が後者よりも

| Machine name  | Compiler  | Compiler option  | CPU time (s)                                | CPU time ratio |
|---------------|---|--|---|----------------|
| M-16<br>No.69 | ifc<br>Intel<br>Fortran<br>7.1 (IA32)             | -static<br>-O2<br>-xKW                                 | 1268.94<br>(interactive)<br>(1CPU)<br>(new) | 3.99           |
| M-16<br>No.70 | pgf90<br>PGI<br>Fortran90<br>5.0-beta2<br>(64bit) | -O2<br>-Mvect<br>=cachesize:<br>1048576                | 1464.00<br>(interactive)<br>(1CPU)<br>(new) | 4.60           |
| M-16<br>No.71 | pgf77<br>PGI<br>Fortran77<br>5.0-beta2<br>(64bit) | -O2<br>-Mvect<br>=cachesize:<br>1048576                | 1469.75<br>(interactive)<br>(1CPU)<br>(new) | 4.62           |
| M-16<br>No.72 | pgf90<br>PGI<br>Fortran90<br>5.0-beta2<br>(32bit) | -O2<br>-Mvect<br>=cachesize:<br>1048576                | 1642.00<br>(interactive)<br>(1CPU)<br>(new) | 5.16           |
| M-16<br>No.73 | pgf77<br>PGI<br>Fortran77<br>5.0-beta2<br>(32bit) | -O2<br>-Mvect<br>=cachesize:<br>1048576                | 1647.73<br>(interactive)<br>(1CPU)<br>(new) | 5.18           |
| M-16<br>No.74 | g77<br>GNU<br>Fortran<br>(64bit)                  | -O2<br>-funroll<br>-loops                              | 1798.39<br>(interactive)<br>(1CPU)<br>(new) | 5.65           |
| M-16<br>No.75 | g77<br>GNU<br>Fortran<br>(32bit)                  | -m32<br>-O2<br>-funroll<br>-loops                      | 2291.15<br>(interactive)<br>(1CPU)<br>(new) | 7.20           |
| M-16<br>No.76 | ifc<br>Intel<br>Fortran<br>7.1 (IA32)             | -parallel<br>-static<br>-O2<br>-xKW                    | 1122.12<br>(interactive)<br>(Dual)<br>(new) | 3.53           |
| M-16<br>No.77 | ifc<br>Intel<br>Fortran<br>7.1 (IA32)             | -parallel<br>-static<br>-O2<br>-xKW<br>-par_threshold0 | 1120.66<br>(interactive)<br>(Dual)<br>(new) | 3.52           |
| M-17<br>No.78 | pgf90<br>PGI<br>Fortran90<br>(64bit)              | -O2<br>-Mvect  | 1362.00<br>(interactive)<br>(1CPU)<br>(new) | 4.28           |
| M-17<br>No.79 | pgf77<br>PGI<br>Fortran77<br>(64bit)              | -O2<br>-Mvect  | 1359.94<br>(interactive)<br>(1CPU)<br>(new) | 4.28           |
| M-17<br>No.80 | pgf90<br>PGI<br>Fortran90<br>(64bit)              | -Mconcur<br>-O2<br>-Mvect                              | 1444.00<br>(interactive)<br>(Dual)<br>(new) | 4.54           |
| M-17<br>No.81 | pgf77<br>PGI<br>Fortran77<br>(64bit)              | -Mconcur<br>-O2<br>-Mvect                              | 1457.50<br>(interactive)<br>(Dual)<br>(new) | 4.58           |

Table.11 Numerical results of Visual Technology

| Machine name                | Compiler                          | Compiler option   | CPU time (s)                          | CPU time ratio | Machine name                 | Compiler                       | Compiler option                   | CPU time (s)  | CPU time ratio |
|-----------------------------|-----------------------------------|---|---------------------------------------|----------------|------------------------------|--------------------------------|-----------------------------------|---|----------------|
| <b>M-18</b><br><b>No.82</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3   | 1582.29<br>(normal<br>queue)<br>(new) | 4.97           | <b>M-23</b><br><b>No.99</b>  | ifc<br>Intel<br>Fortran<br>7.1 | -O3                               | 1644.80<br>(normal<br>queue)<br>(new)                   | 5.17           |
| <b>M-18</b><br><b>No.83</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3<br>-tpp7<br>-xW   | 1407.28<br>(normal<br>queue)<br>(new) | 4.42           | <b>M-23</b><br><b>No.100</b> | ifc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp7<br>-xW               | 1469.64<br>(normal<br>queue)<br>(new)                   | 4.62           |
| <b>M-19</b><br><b>No.84</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3   | 1338.50<br>(normal<br>queue)<br>(new) | 4.21           | <b>M-24</b><br><b>No.101</b> | ifc<br>Intel<br>Fortran<br>7.1 | -O3                               | 1388.95<br>(normal<br>queue)<br>(new)                   | 4.37           |
| <b>M-19</b><br><b>No.85</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3<br>-tpp7<br>-xW   | 1218.0<br>(normal<br>queue)<br>(new)  | 3.83           | <b>M-24</b><br><b>No.102</b> | ifc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp7<br>-xW               | 1248.4<br>(normal<br>queue)<br>(new)                    | 3.92           |
| <b>M-20</b><br><b>No.86</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3   | 1390.20<br>(normal<br>queue)<br>(new) | 4.37           | <b>M-25</b><br><b>No.103</b> | ifc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp7<br>-xW               | 1186.03<br>(normal<br>queue)<br>(new)                   | 3.73           |
| <b>M-20</b><br><b>No.87</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3<br>-tpp7<br>-xW   | 1232.58<br>(normal<br>queue)<br>(new) | 3.88           | <b>M-25</b><br><b>No.104</b> | ifc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp7<br>-xW<br>-axW       | 1185.71<br>(normal<br>queue)<br>(new)                   | 3.73           |
| <b>M-21</b><br><b>No.88</b> | ifc<br>Intel<br>Fortran<br>7.1    | nothing   | 1091.20<br>(interactive)<br>(new)     | 3.43           | <b>M-26</b><br><b>No.105</b> | ifc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp7<br>-axW              | 1171.97<br>(interactive)<br>(1CPU)<br>(99.9%)<br>(new)  | 3.68           |
| <b>M-21</b><br><b>No.89</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3<br>-tpp7<br>-xW   | 935.47<br>(interactive)<br>(new)      | 2.94           | <b>M-26</b><br><b>No.106</b> | ifc<br>Intel<br>Fortran<br>7.1 | -parallel<br>-O3<br>-tpp7<br>-axW | 1083.21<br>(interactive)<br>(Dual)<br>(199.2%)<br>(new) | 3.41           |
| <b>M-21</b><br><b>No.90</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3<br>-tpp7<br>-xW<br>-axW   | 935.06<br>(interactive)<br>(new)      | 2.94           | <b>M-27</b><br><b>No.107</b> | efc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp2                      | 1164.71<br>(interactive)<br>(1CPU)<br>(99.3%)<br>(new)  | 3.66           |
| <b>M-21</b><br><b>No.91</b> | pgf90<br>PGI<br>Fortran90<br>4.02 | -Bstatic  | 1378.00<br>(interactive)<br>(new)     | 4.33           | <b>M-27</b><br><b>No.108</b> | efc<br>Intel<br>Fortran<br>7.1 | -parallel<br>-O3<br>-tpp2         | 699.65<br>(interactive)<br>(Dual)<br>(197.1%)<br>(new)  | 2.20           |
| <b>M-21</b><br><b>No.92</b> | pgf90<br>PGI<br>Fortran90<br>4.02 | -Bstatic<br>-fast   | 1238.00<br>(interactive)<br>(new)     | 3.89           | <b>M-27</b><br><b>No.109</b> | efc<br>Intel<br>Fortran<br>7.1 | -parallel<br>-O3<br>-tpp2         | 485.45<br>(interactive)<br>(4CPU)<br>(382.7%)<br>(new)  | 1.53           |
| <b>M-21</b><br><b>No.93</b> | pgf90<br>PGI<br>Fortran90<br>4.02 | -Bstatic<br>-fast<br>-Mcache_align<br>-Mvect=sse  | 1214.00<br>(interactive)<br>(new)     | 3.82           | <b>M-28</b><br><b>No.110</b> | efc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp2                      | 1315.18<br>(interactive)<br>(1CPU)<br>(99.3%)<br>(new)  | 4.13           |
| <b>M-21</b><br><b>No.94</b> | pgf90<br>PGI<br>Fortran90<br>4.02 | -Bstatic<br>-fast<br>-Mcache_align<br>-Mvect=sse<br>-tpp7                                     | 1216.00<br>(interactive)<br>(new)     | 3.82           | <b>M-28</b><br><b>No.111</b> | efc<br>Intel<br>Fortran<br>7.1 | -parallel<br>-O3<br>-tpp2         | 793.56<br>(interactive)<br>(Dual)<br>(197.4%)<br>(new)  | 2.49           |
| <b>M-21</b><br><b>No.95</b> | pgf90<br>PGI<br>Fortran90<br>4.02 | -Bstatic<br>-fast<br>-Mcache_align<br>-Mvect=sse,<br>cachesize:262144<br>-tpp7                | 1214.00<br>(interactive)<br>(new)     | 3.82           | <b>M-28</b><br><b>No.112</b> | efc<br>Intel<br>Fortran<br>7.1 | -parallel<br>-O3<br>-tpp2         | 558.41<br>(interactive)<br>(4CPU)<br>(383.7%)<br>(new)  | 1.76           |
| <b>M-21</b><br><b>No.96</b> | pgf90<br>PGI<br>Fortran90<br>4.02 | -Bstatic<br>-fast<br>-Mcache_align<br>-Mvect=assoc,<br>cachesize:262144,<br>prefetch<br>-tpp7 | 1288.00<br>(interactive)<br>(new)     | 4.05           | <b>M-29</b><br><b>No.113</b> | efc<br>Intel<br>Fortran<br>7.1 | -O3<br>-tpp2                      | 946.75<br>(interactive)<br>(1CPU)<br>(99.9%)<br>(new)   | 2.98           |
| <b>M-22</b><br><b>No.97</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3   | 1619.94<br>(interactive)<br>(new)     | 5.09           | <b>M-29</b><br><b>No.114</b> | efc<br>Intel<br>Fortran<br>7.1 | -parallel<br>-O3<br>-tpp2         | 566.31<br>(interactive)<br>(Dual)<br>(198.9%)<br>(new)  | 1.78           |
| <b>M-22</b><br><b>No.98</b> | ifc<br>Intel<br>Fortran<br>7.1    | -O3<br>-tpp7<br>-xW   | 1459.02<br>(interactive)<br>(new)     | 4.59           |                              |                                |                                   |   |                |

Table.12 Numerical results of the Open-SCC of HIT

実行速度が速いという結果を示した。以上から、Pentium 4機では、Intel FortranコンパイラがCompaq Visual FortranコンパイラおよびPGI Fortranコンパイラよりも高速に計算を実行することが明らかになった。

No.97~No.106(M-22~M-26)のXeon機の結果に注目する。クロック周波数などの性能向上に伴い、計算時間は徐々に短縮されている。また、最適化オプション(コンパイルオプション)の効果も顕著に現れている。しかしながら、逐次計算において最も良い結果を示したM-26のNo.105においても、Pentium 4機(M-21)とはまだ有意な差異が認められる。また、自動並列化機能を利用したDual CPU(2スレッド)の並列計算では、その効果はあまり顕著には現れなかった。以上から、現時点ではPentium 4機の方がXeon機よりも計算速度が速いという結論を得た。なお、Xeon機では4CPU(4スレッド)対応のXeon MPという型番がある。しかしながら、その最高クロック速度は現在2.0GHzであり、また対応するマザーボード(mother board)のチップセット(chip set)の性能も悪く、現状では技術開発はあまり進んでいないようである。

Intel社の64BitマイクロプロセッサであるItanium 2(IA-64)の結果(No.107~No.112)について考察を行う。特筆すべきは自動並列化機能を利用したスカラー並列計算の結果(No.108およびNo.109, No.111およびNo.112)である。4CPU(4スレッド)の計算(No.109およびNo.112)では、スーパーコンピュータ1PE(Table.7のM-11, No.28)の2倍弱に迫る結果を示しており、スーパーコンピュータ1PEとほぼ同程度の性能を有することが示された。最新機種(M-29)では、自動並列化機能を利用したDual CPU(2スレッド)のスカラー並列計算において、M-27およびM-28の4CPU(4スレッド)と同程度の結果を得た。今度ともItanium 2の技術開発は急速に進むことが期待され、自動並列化機能はRIAM-COMPACTの実用化へ向けて強力なツールになることが期待される。M-27~M-29において、自動並列化機能を利用したスカラー並列計算のスケラビリティは以下の通りである。

$$\begin{aligned} & \text{(M-27)} \\ & \text{No.107(1CPU)/No.108(2CPU)} \approx 1.66 \\ & \text{No.107(1CPU)/No.109(4CPU)} \approx 2.40 \end{aligned} \quad (13)$$

$$\begin{aligned} & \text{(M-28)} \\ & \text{No.110(1CPU)/No.111(2CPU)} \approx 1.66 \\ & \text{No.110(1CPU)/No.112(4CPU)} \approx 2.36 \end{aligned} \quad (14)$$

$$\begin{aligned} & \text{(M-29)} \\ & \text{No.113(1CPU)/No.114(2CPU)} \approx 1.67 \end{aligned} \quad (15)$$

自動並列化機能を利用したスカラー並列計算においては、CPU(スレッド)の使用率(%)は重要な指標となる。ここでは、M-28の対話型処理を例にとり、その算出方法について説明する。CPU(スレッド)の使用率(%)はCシェルにおけるtimeコマンドで出力される。以下には、BシェルとCシェルでtimeコマンドを用いた場合の出力結果の比較を示す。また併せてそれぞれのシェル・スクリプトとその実行方法も示す。両者の計算は、他の計算が走っていない状態で連続して実行した結果である。

```
[tuchida@ita1 ~/smp]$ cat parallel-bsh

export OMP_NUM_THREADS=2
export KMP_LIBRARY=turnaround

efc -parallel -O3 -tpp2 -o 3d.out riam-compact.f

time ./3d.out
[tuchida@ita1 ~/smp]$ sh parallel-bsh
real    13m18.163s
user    26m14.858s
sys     0m0.418s
```

Table.13 Case of the B shell, Dual CPU, Interactive processing

```
[tuchida@ita1 ~/smp]$ cat parallel-csh

setenv OMP_NUM_THREADS 2
setenv KMP_LIBRARY turnaround

efc -parallel -O3 -tpp2 -o 3d.out riam-compact.f

time ./3d.out
[tuchida@ita1 ~/smp]$ csh parallel-csh
1563.166u 0.417s 13:12.16 197.3%
0+0k 0+0io 146pf+0w
```

Table.14 Case of the C shell, Dual CPU, Interactive processing

Table.13およびTable.14において、userまたはuは計算の実行に要した時間(ユーザー時間)、sysまたはsはシステムのオーバーヘッドに費やされた時間(システム時間)、realは実行開始から実行終了までに経過した時間(実時間)を示す。ここで、ユーザー時間にはスレッドのCPU時間の合計(この場合にはDual CPUの合計)が表示されている。また、Table.14では左から三番目の13:12.16が実時間を示している。先に述べたように、CシェルではCPU(スレッド)の使用率(%)が表示されているが、Bシェルではそれは表示されていない。

スカラーおよびベクトル並列計算におけるCPU時間の変化をFig.4およびFig.5に示す。Fig.4では、M-13の結果のみを縦軸のスケールを変更して再度表示した。M-16およびM-17を除いては、CPU(PE, スレッド)を1台

から2台に増やした場合が最も並列化効率が良いことが分かる。この結果からも、RIAM-COMPACTに対してSMPにおける自動並列機能は非常に有効であることが見て取れる。

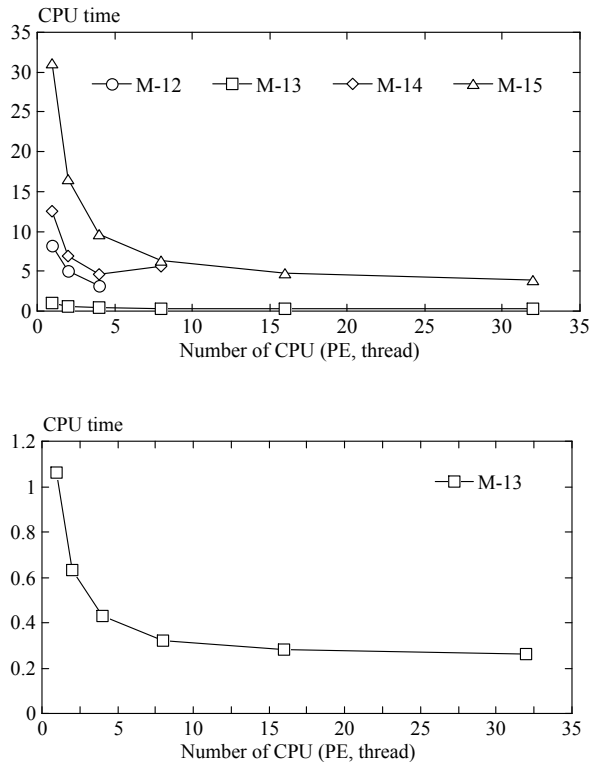


Fig.4 Variation in the CPU time in parallel calculation, M-12, M-13, M-14, M-15

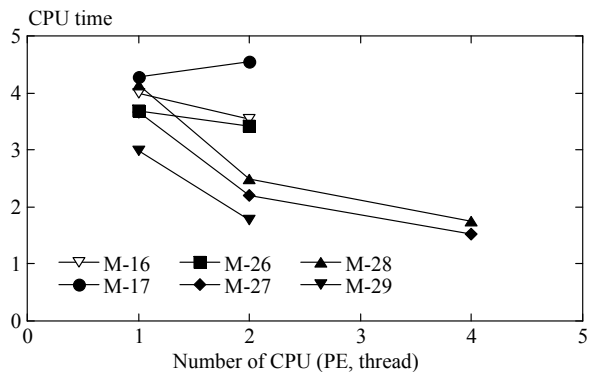


Fig.5 Variation in the CPU time in parallel calculation, M-16, M-17, M-26, M-27, M-28, M-29

## 5. 結言

RIAM-COMPACTの実用化へ向けた検討として、最新の計算機性能と、Fortranのコンパイルオプションに関する幾つかの知見を報告した。これまで、LESによる乱

流数値シミュレーションは、スーパーコンピュータか、あるいは、スーパーコンピュータと同レベルの性能を有する大規模なPCクラスタを用いなければ困難であった。しかしながら、同一条件におけるCPU時間の比較を行った結果、最近の計算機はスーパーコンピュータ1PE並みの性能を有することが明らかになった。特に逐次計算に関しては、Intel Fortranコンパイラ7.1を用い、Pentium 4機の最新チップセット(i875PE)を搭載したマシンがスーパーコンピュータ1PEの3倍弱に迫る結果を示した。一方、自動並列化機能を利用した並列計算では、同じくIntel Fortranコンパイラ7.1を用い、同社の64BitマイクロプロセッサであるItanium 2を搭載したSMPがスーパーコンピュータ1PEの2倍弱に迫る結果を示した。

今後はIntel Pentium 4機の4ノード(4CPU)程度の小規模PCクラスタを構築し、MPI(Message Passing Interface)を利用したRIAM-COMPACTのスカラー並列計算を行う予定である。MPIとは、分散メモリ型の並列計算機において、各ノード間のメッセージ(データ)のやり取りをするために用いるMPLの標準規格である。その他のMPLとしてはPVM(Parallel Virtual Machine)があるが、今やMPIが業界標準(de facto standard)である。MPIではユーザーのさまざまな要求に応えるため、多くの関数が用意されている。MPIを使ったプログラムは共有メモリ型の並列計算機(SMP)で実行することも可能である。よって、SMP上で自動並列化機能を利用した並列計算と、MPIを利用した並列計算を行い、両者のCPU時間の比較を行う予定である。

最後に、PCの性能は今後とも急速なスピードで向上することが期待され、RIAM-COMPACTの実用化はすぐ目前である。

## 謝辞

本報をまとめるにあたり、九州大学応用力学研究所の矢木雅敏先生、九州大学情報基盤センターの南里豪志先生、三井造船株式会社の林健一氏、富士通株式会社の上野潤一郎氏、坂梨末幸氏、住商エレクトロニクス株式会社の安達賢氏、ビジュアルテクノロジー株式会社の舟窪辰也氏、渡辺雅俊氏、株式会社エッチ・アイ・ティーの山田富和氏、内田盛久氏、吉田雅彦氏に多くの助言を頂いた。ここに記して感謝の意を表します。

## 参考文献

- 1) 内田孝紀, 大屋裕二, ネストグリッドを用いた複雑地形上の風況予測シミュレーション, 日本風工学会論文集, No.92, pp.135-144, 2002
- 2) 渡部善隆, VPP700/56利用の手引 第1.9版, 1998

## Appendix

以下では、それぞれの計算に使用したシェル・スクリプトや実行方法などの一例を示す。

```
C:\¥uchida¥test>type go-intel.bat
ifl /O3 /G7 /QaxW /QxW riam-compact.f
riam-compact.exe
C:\¥uchida¥test>go-intel
```

Fig.6 Case of the NetVista, Serial processing, Interactive processing, Our laboratory

```
app% cat comp-int.sh
frtpr -Kfast -o 3d.out riam-compact.f
app% csh comp-int.sh (or sh comp-int.sh)
```

### (a) Compile, Interactive processing

```
app% cat comp-bat.sh
#!/bin/csh
#@ $-q L
#@ $-eo
cd /home/OAD/WE/takanori
frt -Kfast -o 3d.out riam-compact.f
app% qsub comp-bat.sh
Request 8319.app submitted to queue: L.
```

### (b) Compile, Batch processing

```
app% cat go.sh
#!/bin/csh
#@ $-q B0
#@ $-oi -eo
cd /home/OAD/WE/takanori
timex ./3d.out
app% qsub go.sh
Request 8321.app submitted to queue: B0.
```

### (c) Execution, Batch processing

Fig.7 Case of the VPP5000/2, Serial processing, RIAM of Kyushu University

```
app% cat serial-es40.sh
#!/bin/csh
#@ $-q E
#@ $-eo
cd /home/OAD/WE/takanori
f90 -arch host -tune host -fast -O5 -o 3d.out
riam-compact.f
time 3d.out
app% qsub serial-es40.sh
Request 8316.app submitted to queue: E.
```

### (a) Serial processing

```
app% cat parallel-es40.sh
#!/bin/csh
#@ $-q S
#@ $-eo
cd /home/OAD/WE/takanori
setenv OMP_NUM_THREADS 2
kf90 -fkapargs='-concurrent' -arch host -tune host -fast
-O5 -o 3d.out riam-compact.f
time 3d.out
app% qsub parallel-es40.sh
Request 8315.app submitted to queue: S.
```

### (b) Parallel processing, Dual CPU

Fig.8 Case of the ES40, Batch processing RIAM of Kyushu University

```
kyu-vpp% cat serial.sh
#!/usr/bin/csh
#@ $-q s
#@ $-lM 7gb
#@ $-oi -eo
cd vpp-test
frt -Kfast -o 3d.out riam-compact.f
timex ./3d.out
kyu-vpp% qsub serial.sh
Request 61776.kyu-vpp submitted to queue: s.
```

### (a) Serial processing

```
kyu-vpp% cat parallel.sh
#!/usr/bin/csh
#@ $-q s8
#@ $-lPv 4
#@ $-lM 7gb
#@ $-oi -eo
cd vpp-test
frt -Wx -Kfast -o 3d.out riam-compact.f
timex ./3d.out
kyu-vpp% qsub parallel.sh
Request 61781.kyu-vpp submitted to queue: s8.
```

### (b) Parallel processing, 4PE

Fig.9 Case of the VPP5000/64 (kyu-vpp), Batch processing, Computing and communications center of Kyushu University

```
kyu-ss% cat serial-gs320-csh
f90 -arch host -tune host -fast -O5 -o 3d.out
riam-compact.f
time ./3d.out
kyu-ss% csh serial-gs320-csh
```

(a) Serial processing

```
kyu-ss% cat parallel-gs320-bsh
OMP_NUM_THREADS=4;export OMP_NUM_THREADS
kf90 -fkapargs='-concurrent' -arch host -tune host -fast
-O5 -o 3d.out riam-compact.f
time ./3d.out
kyu-ss% sh parallel-gs320-bsh
```

(b) Parallel processing, 4CPU, B shell

```
kyu-ss% cat parallel-gs320-csh
setenv OMP_NUM_THREADS 4
kf90 -fkapargs='-concurrent' -arch host -tune host
-fast -O5 -o 3d.out riam-compact.f
time ./3d.out
kyu-ss% csh parallel-gs320-csh
```

(c) Parallel processing, 4CPU, C shell

Fig.10 Case of the GS320 (kyu-ss),  
Interactive processing,  
Computing and communications center  
of Kyushu University

```
kyu-cc% cat 3d-parallel04.sh
#!/usr/bin/csh
#@-$-q sc32
#@-$-eo
cd /home/usr3/a70473a
f90 -Kparallel,reduction,instance=4,fast,eval -o 3d.out
riam-compact.f
setenv PARALLEL 4
timex ./3d.out
kyu-cc% qsub 3d-parallel04.sh
Request 9991.kyu-cc submitted to queue: sc32.
```

Fig.11 Case of the GP7000F/900 (kyu-cc),  
Parallel processing, 4 CPU  
Batch processing,  
Computing and communications center  
of Kyushu University

```
p4111 /home/tuchida> cat serial-csh
ifc -tpp7 -xW -axW -O3 riam-compact.f
time ./a.out
p4111 /home/tuchida> csh serial-csh
```

Fig.12 Case of the p4111, Serial processing,  
Interactive processing, Open-SCC of HIT

```
scc /home/tuchida> cat comp-serial.sh
ifc -tpp7 -xW -axW -O3 riam-compact.f
scc /home/tuchida> csh comp-serial.sh
(or sh comp-serial.sh)
```

(a) Compile

```
scc /home/tuchida> cat go-serial.sh
#!/bin/csh
#BSUB -q normal
#BSUB -m iax06
#BSUB -o log.out
time ./a.out
scc /home/tuchida> bsub < go-serial.sh
Job <160006> is submitted to queue <normal>.
```

(b) Execution

Fig.13 Case of the iax06, Serial processing,  
Batch processing, Open-SCC of HIT

```
[tuchida@ita1 ~/smp]$ cat serial-csh
efc -O3 -tpp2 -o 3d.out riam-compact.f
time ./3d.out
[tuchida@ita1 ~/smp]$ csh serial-csh
```

Fig.14 Case of the ita1, Serial processing,  
Interactive processing, Open-SCC of HIT