九州大学学術情報リポジトリ Kyushu University Institutional Repository

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 九州大学応用力学研究所

大屋,裕二 九州大学応用力学研究所

https://doi.org/10.15017/6770634

出版情報:九州大学応用力学研究所所報.125, pp.5-20, 2003-09. 九州大学応用力学研究所 バージョン: 権利関係:

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

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,大型ウィ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に 的を絞り, RIAM-COMPACT(Research Institute for <u>Applied Mechanics</u>, Kyushu University, <u>Com</u>putational <u>Prediction of Airflow over Complex Terrain</u>)と称する風 況予測シミュレータを開発している¹⁾. この数値モデルは, (有限)差分法FDM(<u>Finite-Difference Method</u>)に基づい たFORTRAN(<u>FOR</u>mula <u>TRAN</u>slater)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 <u>Simulation</u>)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.

*九州大学応用力学研究所

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(<u>Symmetric Multi</u> <u>Processor</u>)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・パッカード)がある.

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

Table.1	Computer	environment	of our	laboratory
---------	----------	-------------	--------	------------

Machine	Model	CPU	Memory	os
name	Vendor	Clock		
M-11	VPP5000/2	PE×2	1.5GB	UXP/V
	Fujitsu		B0 and B1 queue	V20
M-12	Alpha Server ES40	Alpha 21264 A (codename : EV67)	lGB 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	Model	CPU	Memory	os
name	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 16GB for	Tru64 UNIX
	Compaq	731MHz×32 (4CPU×8)	interactive	
M-15	GP7000F Model 900	SPARC64-GP	64GB 32GB for	Solaris7
	Fujitsu	300MHz×64	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) Mem : 2GB (512KB×4) OS : Turbo Linux 8 for AMD 64, KERNEL 2.4.20-1	
M-17	CPU : Opteron 1.8GHz×2 (Dual Processor) Mem : 2GB (512KB×4)	
	Cable.4 Computer environment of	

Visual Technology

Machine	Specification
name	
M-18	Hostname : p4103
	CPU : Pentium 4 2.4GHz (512KB cache/FSB400)
	Mem : 1GB RDRAM (PC800-45nsec)
M-19	Hostname : p4104
	CPU : Pentium 4 3.06GHz (512KB cache/FSB533)
	Mem : 1GB RDRAM (PC1066)
M-20	Hostname : p4110
	CPU : Pentium 4 3.06GHz (512KB cache/FSB533)
	Mem : 4GB DDR266 SDRAM
M-21	Hostname : p4111
	CPU : Pentium 4 3.0GHz (512KB cache/FSB800)
	M/B : i875PE chipset
	$Mem: DDR400 512MB \times 4 = 2,048MB$
	HDD : Maxtor 6Y080L0 (IDE 80GB 7200rpm)
	OS : SuSE Linux 7.3 Kernel 2.4.20
M-22	Hostname : iax01
	CPU : Xeon 2.8GHz Dual (512KB cache/FSB400)
14.33	Mem : 8GB DDR200 SDRAM
M-23	Hostname : iax02
	CPU: Xeon 2.8GHz Dual (512KB cache/FSB400)
	Mem : 4GB DDR200 SDRAM
M-24	Hostname : iax06
	CPU : Xeon 2.8GHz Dual (512KB cache/FSB533)
	Mem : 3GB DDR266 SDRAM
M-25	Hostname : iax06
	CPU : Xeon 3.06GHz Dual (512KB cache/FSB533)
	Mem : 2GB DDR266 SDRAM
M-26	CPU : Xeon 3.06 GHz \times 2 (Dual Processor)
	Memory : 2,048MB
	Chipset : E7501
M-27	Hostname : ital
	CPU : Itanium 2 1.0GHz 3MB cache Quad SMP
	Mem : 16GB DDR266 SDRAM
	ADD : SUSI 300B
M-28	Ustrome ital
141-20	CPU : Itanium 2 000MHz 1 5MB coche Qued SMP
	Mem · 32GB DDR200 SDRAM
	HDD · SCSI 36GB
	OS : Red Hat Advanced Server 2 14S
M-29	CPU : Itanium 2 1.5GHz 6MB cache Dual SMP
1	able.5 Computer environment of

the Open-SCC of HIT

分散メモリ方式では、OS(<u>Operating System</u>)とメモリ (memory)はそれぞれのCPU(<u>Central Processing Unit</u>), あるいは、PE(<u>Precessing Element</u>)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(<u>Message Passing Library</u>)

6

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・パッカード)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・フ 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 <u>Super Computing Center</u>)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 <u>C</u>ontrol <u>P</u>rotocol/<u>I</u>nternet <u>P</u>rotocol)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¹⁾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 planein 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×16h×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×cos²(π(x²+y²)^{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81×61×51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(0.2~0.55)h, z方向の格子解像度は不等間隔に(0.003~0.5)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/ν)=10⁴とした.時間刻みはΔt=2×10⁻³h/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=100~110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(<u>Input/O</u>utput)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≒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-No.5/No.3)×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	Compiler	Compiler	CPU time	CPU
name		option	(s)	time
				ratio
M-01	ifl	nothing	2667.844	8.39
	Intel Fortran		(interactive)	
<u>No.1</u>	7.0			
M-01	ifl	/03	2642.625	8.31
	Intel Fortran		(interactive)	
<u>No.2</u>	7.0			
M-01	ifl	nothing	2692.063	8.46
	Intel Fortran		(interactive)	
No.3	7.1			0.10
M-01	ifl	/03	2696.875	8.48
Not	Intel Fortran		(interactive)	
N0.4	1.1			7.50
IVI-01	111	/03	2384.094	7.50
No 5	Intel Fortran	/0/	(interactive)	
110.5	7.1	/Qax w		
M 01	:0	/Qxw	2070 429	6.51
141-01	111	103	2070.438 (interactive)	0.51
No 6	Intel Fortran		(Interactive)	
110.0	7.1	/Qax W	(new)	1
M-01	f90	nothing	3394 859	10.67
	DIGITAL	liotining	(interactive)	10.07
No.7	Visual Fortran		(
	5.0			
M-01	f90	nothing	3338.969	10.50
	Compaq		(interactive)	
No.8	Visual Fortran			
	6.1			
M-01	f90	/arch:host	2818.641	8.86
	Compaq	/tune:host	(interactive)	
No.9	Visual Fortran	/fast		
	6.1	/optimize:4		
M-01	f90	nothing	3299.641	10.37
	Compaq		(interactive)	
No.10	Visual Fortran	1		
34 04	6.6			0.00
M-01	190	/arch:host	2798.156	8.80
No. 11	Compaq	/tune:host	(interactive)	
140.11	visual Fortran	/iast		
M 01	0.0 mo	/optimize:4	2207.0(0	7 64
101-01	190	/arcn:nost	(interactive)	/.34
No 12	Visual Fortron	/fune:nost	(interactive)	
110.14	6.6	/ontimize.4	(new)	
M-01	f77	nothing	3309 703	10 41
141-01	Compag	nounng	(interactive)	10.41
No.13	Visual Fortran		(interactive)	
	6.6			

		0 11	CIDIT (1	CDV
Machine	Compiler	Compiler	CPU time	СРО
name		option	(s)	time
				ratio
M-02	f90	/arch:host	2375.609	7.47
	Compaq	/tune:host	(interactive)	
No.14	Visual Fortran	/fast		
	6.1	/optimize:4		
M-03	f90	/arch:host	3026.609	9.52
	Compaq	/tune:host	(interactive)	
No.15	Visual Fortran	/fast		
	6.1	/optimize:4		
M-04	f90	/arch:host	5606.622	17.63
	Compaq	/tune:host	(interactive)	
No.16	Visual Fortran	/fast		
	6.1	/optimize:4		
M-05	f90	/arch:host	9162.061	28.81
	Compag	/tune:host	(interactive)	20101
No.17	Visual Fortran	/fast	(
	61	/ontimize-4		
M-06	ngf90	nothing	8400.0	26 41
	PGI Fortran	nothing	(interactive)	20.41
No 18	3 2_3		(interactive)	
MAG	<u> </u>	<u> </u>		24.71
IVI-UO	pg190	-fast	7860.0	24./1
N. 10	PGI Fortran		(interactive)	
N0.19	3.2-3			
M-06	pgf90	-fast	7140.0	22.45
	PGI Fortran	-Mcache_align	(interactive)	
No.20	3.2-3	-Mvect=sse		
M-07	f90	nothing	3908.422	12.29
	Compag	U	(interactive)	
No.21	Visual Fortran		(, , , , , , , , , , , , , , , , , , ,	
	6.1			
M-07	f90	/arch:host	3343,906	10.51
	Compag	/tune:host	(interactive)	10101
No.22	Visual Fortran	/fast	(
	6.1	/optimize:4		
M-08	f90	/arch:host	3817 328	12.00
	Compag	/tune:host	(interactive)	12.00
No.23	Visual Fortran	/fast	(Interactive)	
110.23	6 1	/inst		
M_00	£0.1	/optimize.4	7140.0	22.45
111-07	Compac	/tune:hest	(interactive)	22.43
No 24	Viewal Fortran	/foot	(interactive)	
110.24	visual rortran	/iast		
M 10	0.1	/optimize:4	0512 605	-20.01
IVI-10	190	/arch:host	9543.695	30.01
No 25	Compaq	/tune:host	(interactive)	
110.25	Visual Fortran	/fast		
	6.1	/optimize:4		

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.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	Compiler	Compiler	CPU time (s)	CPU	Machine	Compiler	Compiler	CPU time (s)	CPU
name		option		time	name		option		time
NT 11	Let an fatay		202.15	ratio	M 12	1-000		1060.0	ratio
IVI-11	frt or iripx	notning	323.15 (B0 or B1 aueue)	1.02	IV1-14	KI90	-IKapargs-	(S queue)	6.10
No.26	Fortran95/VP		(2000)		No.40	Compaq KAP Fortran	-tune ev6	(2CPU) (198%)	
M-11	frt or frtpx	-Oe	321.01 (B0 or B1 queue)	1.01	M-12	kf90	-fkapargs=	1603.0	5.04
No.27	Fortran95/VP		(DU DI DI queue,		N- 41	Compaq KAP	'-concurrent'	(S queue) (2CPII)	
M-11	frt or frtpx	-Kfast	318.07	1	N0.41	Fortran		(198%)	
No.28	Fortran95/VP		(B0 or B1 queue)	standard	M-12	kf90	-fkapargs=	(new) 1620.02	5.09
M-11	frt or frtpx	-Kfast	293.32	0.92		Compag KAP	'-concurrent'	(S queue)	0.0-
No.29	Fortran95/VP		(B0 or B1 queue) (new)		No.42	Fortran	-arch host -tune host	(2CPU) (195%)	
M-11	frt or frtpx	-05	318.37	1.00	M-12	k-f90	-fast -U5	(new)	2 77
No.30	Fortran95/VP	-KA32	(B0 or B1 queue)		No 43	Compaq KAP	-concurrent' -tune ev6	(S queue) (4CPU)	5.77
M-11	frt or frtpx	-Kfast	587.56	1.85		Fortran		(394%)	L!
No 31	Fortran95/VP		(B0 or B1 queue) (Scalar SOR)		M-12	kf90	-fkapargs= '-concurrent'	1011.0 (S queue)	3.18
M-11	frt or frtpx	-Wv,-sc	7509.17	23.61	No.44	Compaq KAP Fortran		(4CPU)	
No. 27	Fortran95/VP		(B0 or B1 queue) (Scalar mode)					(392%) (new)	
N0.5∠ M-12	fqn	nothing	3492 686	10.98	M-12	kf90	-fkapargs=	1009.0	3.17
171-14	Compaq	10000000	(E queue)	10.70	No.45	Compaq KAP	-concurrent -tune ev6	(S queue) (4CPU)	1 1
No.33	Fortran	′	(99%)			Fortran		(393%)	1 1
M-12	f90 Compag	-fast	3174.305	9.98	M-12	kf90	-fkanargs=	(new)	2 16
No 34	Fortran	} '	(E queue) (99%)	1	178-84	Compace KAP	'-concurrent'	(S queue)	3.10
M-12	f90	-arch ev6	3163.536	9.95	No.46	Fortran	-arch ev6	(4CPU) (303%)	1
	Compaq	-tune ev6	(E queue)	1			-05	(new)	I
No.35	Fortran	-fast -O	(99%)	1	M-12	kf90	-fkapargs=	1019.0	3.20
M-12	f90	-arch ev6	2584.0	8.12	No 47	Compaq KAP	-concurrent -arch ev6	(S queue) (4CPU)	1
NI- 26	Compaq	-tune evo	(E queue) (00%)	i I	110.77	Fortran	-tune ev6	(393%)	1
N0.30	Fortran	-0	(new)		1		-fast	(new)	ı I
M-12	f90	-arch ev6	2938.0	9.24	M-12	kf90	-fkapargs=	1016.0	3.19
No.37	Compaq Fortran	-tune evo -O5	(E queue) (99%) (new)		No.48	Compaq KAP Fortran	'-concurrent' -arch host	(S queue) (4CPU)	
M-12	f90	-arch ev6	2639.0	8.30			-tune nost -fast	(393%) (new)	
AT. 20	Compaq	-tune ev6	(E queue)	i 📘			-0	X	
No.38	Fortran	-145t -05	(99%) (new)		M-12	kf90	-fkapargs=	1020.0 (S guerre)	3.21
M-12	f90	-arch host	2592.0	8.15	No.49	Compaq KAP	-arch host	(4CPU)	
No 20	Compaq	-tune nost -fast	(E queue) (99%)	. 1		Fortian	-tune host	(392%)	
N0.39	Fortran	-05	(new)				-1asi -05	(new)	I

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 合では、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無 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

10

ー名(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(<u>N</u>etwork Queueing <u>System</u>)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frtpxを用 いて対話的に生成した.もう一つは、frt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 α とベクトル加速率β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 α および β 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.

real (Ta)	5:23.15 (323.15 (s))
user (Tal)	5:13.75 (313.75 (s))
sys (Ta2)	0.29 (0.29 (s))
vu-user (Ta3)	4:52.03 (292.03 (s))
vu-sys (Ta4)	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

α および β 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)
(3)
$\beta = Ts2 \div Tv$
=[(Tb1+Tb2)-{(Ta1+Ta2)-(Ta3+Ta4)}]/(Ta3+Ta4)

$$\alpha \approx 0.997, \ \beta \approx 25.26$$

(4)



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 a 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)

(6)

(7)

を求める方法もある.これをVU率と言う.VU率は

(Ta3+Ta4)/(Ta1+Ta2)≒0.93

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2). 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 α 、ベクトル加速率 β 、実行性能 向上比E(=スカラーモードでの実行時間(Tb1+Tb2)/ベク トルモードでの実行時間(Ta1+Ta2), RIAM-COMPACT の場合にはE≒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 α 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



Fig.3 Conceptual figure of the order of calculation

加速率 β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、Compag 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, Compag 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 -scalaropt=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 うにする.

For the B shell (/bin/sh, bash) : export OMP_NUM_THREADS=4 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)がフル稼動した場合には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,

No.39(1CPU, 1スレッド)/No.42(2CPU, 2スレッド)≒1.60 No.39(1CPU, 1スレッド)/No.49(4CPU, 4スレッド)≒2.54 (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)のスカラー並列計算では, 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.

No.51(1PE)/No.52(2PE) ≒ 1.68 No.51(1PE)/No.53(4PE) ≒ 2.45 No.51(1PE)/No.54(8PE) ≒ 3.33 No.51(1PE)/No.55(16PE) ≒ 3.81 No.51(1PE)/No.56(32PE) ≒ 4.02

(10)

M-14のスカラー並列サーバ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 一逐次計算(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)のスカラー並列計 算(No.58), 4CPU(4スレッド)のスカラー並列計算(No.59 ~No.61), 8CPU(8スレッド)のスカラー並列計算(No.62) は, Compaq KAP Fortranの自動並列化機能を利用し て実行した. スカラー並列計算において, それぞれのス ケーラビリティは以下の通りである.

No.57(1CPU, 1スレッド)/No.58(2CPU, 2スレッド)≒1.81 No.57(1CPU, 1スレッド)/No.61(4CPU, 4スレッド)≒2.72 No.57(1CPU, 1スレッド)/No.62(8CPU, 8スレッド)≒2.22 (11)

No.58(2CPU, 2スレッド)およびNo.61(4CPU, 4スレッド) に示すスカラー並列計算では,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 Compiler		Compiler	CPU time	CPU	
name		option	(s)	time	
M 12	6.4	W.faat	211.70	ratio	
M-13 No 50	Irt	-Klast	(s queue)	0.98	
110.50	Fortran95/VP	· · · · · · · · · · · · · · · · · ·	(1PE)		
M-13	frt VDD Fortron	-Wx	335.93	1.06	
N0.51	(Fortran95/VPP)	-Klast	(1PE)		
M-13	frt	-Wx	199.77	0.63	
No.52	VPP Fortran	-Kfast	(x16 queue)		
M-13	(Fortran95/VPP)	-Wx	136.95	0.43	
No.53	VPP Fortran	-Kfast	(x16 queue)	0.45	
34.12	(Fortran95/VPP)	117	(4PE)	0.00	
M-13 No 54	VPP Fortran	-WX -Kfast	(x16 queue)	0.32	
110.54	(Fortran95/VPP)		(8PE)		
M-13	frt VDD Footoor	-Wx	88.09	0.28	
No.55	(Fortran (Fortran)	-Klast	(16PE)		
M-13	frt	-Wx	83.53	0.26	
No.56	VPP Fortran	-Kfast	(p32 queue)		
M_14	(Fortran95/VPP)	-arch ev6	(32PE) 3990.0	12.54	
No.57	Compace Fortron	-tune ev6	(interactive)	14.54	
	Compaq Portran	-05	(1CPU)		
			(99%) (new)		
M-14	kf90	-fkapargs=	2200.0	6.92	
	Compaq KAP	'-concurrent'	(interactive)		
No.58	Fortran	-arch evo	(2CPU)		
		-O5 -fast	(new)		
M-14	kf90	-fkapargs=	1495.0	4.70	
N. 60	Compaq KAP	'-concurrent'	(interactive)		
N0.39	Fortran	-tune ev6	(380%)		
	1.00.0	-05	(new)		
M-14	k190	-tkapargs=	1465.0 (interactive)	4.61	
No.60	Compaq KAP	-arch ev6	(4CPU)		
110.00	ronnan	-tune ev6	(382%)		
M-14	k f90	-O5 -fast -fkapargs=	(new)	1.65	
141-14	Compos VAD	'-concurrent'	(interactive)	4.05	
No.61	Fortran	-arch host	(4CPU)		
		-tune host -05 -fast	(379%) (new)		
M-14	kf90	-fkapargs=	1795.0	5.64	
	Compag KAP	'-concurrent'	(interactive)		
No.62	Fortran	-arch host	(8CPU) (722%)		
		-O5 -fast	(new)		
M-15	f90	-Kparallel,	9905.45	31.14	
No 43	Fortran95	instance=1.	(sc32 queue) (1CPU)		
110.03		fast,eval	(new)		
M-15	f90	-Kparallel,	5277.40	16.59	
No 64	Fortran95	instance=2.	(sc32 queue) (2CPU)		
110.04		fast,eval	(new)		
M-15	f90	-Kparallel,	3070.46	9.65	
No 45	Fortran95	instance=4	(scs2 queue) (4CPU)		
110.05		fast,eval	(new)		
M-15	f90	-Kparallel,	2013.76	6.33	
No 44	Fortran95	reduction, instance=8	(sc32 queue) (8CPU)		
110.00		fast,eval	(new)		
M-15	f90	-Kparallel,	1530.83	4.81	
No CT	Fortran95	reduction, instance=16	(sc32 queue) (16CPU)		
NO.07		fast,eval	(new)		
M-15	f90	-Kparallel,	1236.77	3.89	
N CO	Fortran95	reduction,	(sc32 queue)		
No.68		fast.eval	(new)		

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 のスケーラビリティは以下の通りである.

No.63(1CPU)/No.64(2CPU)≒1.88 No.63(1CPU)/No.65(4CPU)≒3.23 No.63(1CPU)/No.66(8CPU)≒4.92 No.63(1CPU)/No.67(16CPU)≒6.47 No.63(1CPU)/No.68(32CPU)≒8.01

(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→

Machine	Compiler	Compiler	CPU time (s)	CPU
name		option		time
	<u> </u>			ratio
M-16	ifc	-static	1268.94	3.99
i i	Intel	-02	(interactive)	
No.69	Fortran	-xKW	(ICPU)	1
	7.1 (IA32)		(new)	
MI-16	pg190	-02 Muaat	1464.00 (interactive)	4.60
NT. 70	Fortran90	=cachesize:	(ICPII)	
No.70	5.0-beta2	1048576	(new)	
	(64bit)		()	
M-16	pgf77	-02	1469.75	4.62
	PGI	-Mvect	(interactive)	
No.71	Fortran77	=cachesize:	(1CPU)	
	5.0-beta2	1048576	(new)	
M 16	(04011) naf00	- 02	1642.00	E 16
WI-10	PGI	-02 -Mvect	(interactive)	5.10
No 72	Fortran90	=cachesize:	(1CPU)	
110.72	5.0-beta2	1048576	(new)	
	(32bit)			
M-16	pgf77	-02	1647.73	5.18
	PGI	-Mvect	(interactive)	
No.73	Fortran77	=cachesize:	(ICPU)	
	$5.0-beta_2$	1048576	(new)	
M 16	(32011)	-02	1708 30	ECE
WI-10	GNU	-02	(interactive)	5.05
No 74	Fortran	-loops	(1CPU)	
140.74	(64bit)		(new)	
M-16	g77	-m32	2291.15	7.20
	GNU	-02	(interactive)	
No.75	Fortran	-funroll	(1CPU)	
M 1/	(32bit)	-loops	(new)	
M-10	lic	-parallel	(interactive)	3.53
N. BC	Fortran	-512110	(Interactive)	
N0.70	71(IA32)	-xKW	(new)	
M-16	ifc	-narallel	1120.66	3 5 7
		-static	(interactive)	3.34
No 77	Intel	-02	(Dual)	
1.0., /	Fortran 7 1 (IA22)	-xKW	(new)	
	7.1 (IA32)	-par_threshold0		
M-17	pgf90	-02	1362.00	4.28
	PGI Fortran00	-Mvect	(interactive)	
No.78	(64hit)		(ICFU)	
M-17	ngf77	-02	1359.94	1 28
141-1 /	PGI	-Mvect	(interactive)	4.20
No 70	Fortran77		(1CPU)	
110.75	(64bit)		(new)	
M-17	pgf90	-Mconcur	1444.00	4.54
	PGI	-02	(interactive)	
No.80	Fortran90	-Mvect	(Dual)	
7	(64bit)		(new)	
M-17	pgt//	-Mconcur	1457.50	4.58
	FUI Fortran 77	-02 Myraat	(interactive)	
No.81	(64bit)	-www.	(Dual)	
	(04011)		(new)	

Table.11 Numerical results of Visual Technology

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 とのご指摘を得た.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)が稼動するが,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

16

		Compiler	CDII time	CBU		Comulton	Compilor	CBU	CDU
Machine	Compiler	option	(s)	time	Nischine	Complier	ontion	(e)	time
aame		option	(3)	ratio	name		option	(3)	ratio
M-18	ifc	-03	1582.29	4 97	M-23	ifc	-03	1644.80	5.17
	Intel		(normal	1.27		Intel		(normal	
No.82	Fortran		queue)		No.99	Fortran		queue)	
M 10	7.1	02	(new)		1 1 12	7.1	01	(new)	
IVI-18	Intel	-03 -tnp7	(normal	4.42	IVI-23	Intel	-03 -tpn7	1409.04 (normal	4.62
No.83	Fortran	-xW	queue)		No.100	Fortran	-xW	queue)	
1,0,00	7.1		(new)			7.1		(new)	
M-19	ifc	-03	1338.50	4.21	M-24	ifc	-03	1388.95	4.37
N. 04	Intel		(normai			Intel		(normal	
N0.84	7.1		(new)		No.101	7.1		(new)	
M-19	ifc	-03	1218.0	3.83	M-24	ifc	-03	1248.4	3.92
	Intel	-tpp7	(normal			Intel	-tpp7	(normal	
No.85	Fortran	-xW	queue)		No.102	Fortran	-xW	queue)	
M 20	/.1	02	(new)	4.25	M 25	7.1 ifa	02	(new)	2.72
11-20	Intel	-03	(normal	4.3/	IV1-25	Intel	-03 -tpp7	(normal	3.73
No.86	Fortran		queue)		No.103	Fortran	-xW	queue)	
	7.1		(new)			7.1		(new)	
M-20	ifc	-03	1232.58	3.88	M-25	ifc	-03	1185.71	3.73
No 97	Intel Fortran	-tpp/	(normal queue)	1	N. 104	Intel Fortron	-tpp/	(normal queue)	
110.8/	7.1		(new)		No.104	7.1	-axW	(new)	
M-21	ifc	nothing	1091.20	3.43	M-26	ifc	-03	1171.97	3.68
	Intel	2	(interactive)			Intel	-tpp7	(interactive)	
No.88	Fortran		(new)		No.105	Fortran	-ax W	(1CPU)	
M_21	1.1 ifc	-03	035 47	2.04	41	7.1		(99.9%) (new)	
191-21	Intel	-tpp7	(interactive)	2.94	M-26	ifc	-parallel	1083.21	3.41
No.89	Fortran	-xŴ	(new)			Intel	-03	(interactive)	
	7.1				No.106	Fortran	-tpp7	(Dual)	
M-21	ifc	-03	935.06	2.94		7.1	-ax w	(199.2%) (new)	
N - 00	Fortran	-tpp/ -xW	(interactive)		M-27	efc	-03	1164.71	3.66
110.90	7.1	-axW	(Intel	-tpp2	(interactive)	
M-21	pgf90	-Bstatic	1378.00	4.33	No.107	Fortran		(1CPU)	
	PGI		(interactive)			7.1		(199.376) (new)	
No.91	Fortran90		(new)		M-27	efc	-parallel	699.65	2.20
M-21	pgf90	-Bstatic	1238.00	3.89	11	Intel	-03	(interactive)	
	PGI	-fast	(interactive)		No.108	Fortran	-tpp2	(Dual) (197.1%)	
No.92	Fortran90		(new)			7.1		(new)	
M_21	4.02	Betatic	1214.00	2 02	M-27	efc	-parallel	485.45	1.53
171-21	PGI	-fast	(interactive)	3.04		Intel	-03	(interactive)	
No.93	Fortran90	-Mcache_align	(new)		No.109	Fortran	-tpp2	(4CPU) (382 7%)	
34.01	4.02	-Mvect=sse	1016.00		4 8	7.1		(new)	
M-21	pg190	-Bstatic	(interactive)	.3.82	M-28	efc	-03	1315.18	4.13
No.94	PGI	-Mcache_align	(new)			Intel	-tpp2	(interactive)	
110124	4.02	-Mvect=sse			No.110	Fortran		(10PU)	
M 11	naf00	-tpp7 Batatia	1214.00	2.04		7.1		(new)	
IVI-21	hBran	-Dstatic	(interactive)	5.82	M-28	efc	-parallel	793.56	2.49
No.95	PGI	-Mcache_align	(new)			Intel	-03	(interactive)	
1,000	Fortran90	-Mvect=sse,			No.111	Fortran	-upp2	(Dual) (197.4%)	
	4.02	cachesize:262144				7.1		(new)	
M-21	ngf90	-Bstatic	1288.00	4 05	M-28	efc	-parallel	558.41	1.76
	POT	-fast	(interactive)	T.U.J		Intel	-03	(interactive)	
No.96	PGI	-Mcache_align	(new)		No.112	Fortran	-cpp2	(4CPU) (383 7%)	
· ·	Fortran90	-Mvect=assoc,				7.1]	(new)	
	7.02	prefetch			M-29	efc	-03	946.75	2.98
		-tpp7				Intel	-tpp2	(interactive)	
M-22	ifc	-03	1619.94	5.09	No.113	Fortran	1	(ICPU) (99.9%)	
	Intel		(interactive)	÷,		7.1		(new)	
No.97	Fortran 71		(new)		M-29	efc	-parallel	566.31	1.78
M-22	ifc	-03	1459.02	4 50	1	Intel	-03	(interactive)	-
	Intel	-tpp7	(interactive)	7.37	No.114	Fortran	-tpp2	(Dual) (198.0%)	
No.98	Fortran	-xW	(new)			7.1		(190.970) (new)	
	7.1						L	1.(

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ーラビリティは以下の通りである.

(M-27)No.107(1CPU)/No.108(2CPU) \approx 1.66 No.107(1CPU)/No.109(4CPU) \approx 2.40 (13) (M-28)No.110(1CPU)/No.111(2CPU) \approx 1.66 No.110(1CPU)/No.112(4CPU) \approx 2.36 (14)

(M-29) No.113(1CPU)/No.114(2CPU)≒1.67

(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 エルでは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





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

- 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 frtpx -Kfast -0 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 -0 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 #@\$-IM 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

(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

20