

Fortran コンパイラの自動並列化性能評価

桧山澄子*・鷹野 澄*・山中佳子*

The Performance Evaluation for the Automatic Parallelization Option of Fortran Compilers

Sumiko HIYAMA*, Kiyoshi TAKANO* and Yoshiko YAMANAKA*

Abstract

We carried out a series of benchmark tests to know the ability of automatic parallelization of four Fortran compilers : SPARC Fortran 77, Cray Fortran 90, Apogee Fortran 77 and Apogee Fortran 90. We aimed to select the best compiler at the time of Operating System version up in the EIC computer. These tests revealed that the best is Apogee Fortran 77, and the worst is Cray Fortran 90 among them. Then we installed the Apogee Fortran 77 and Apogee Fortran 90 in the new-OS instead of Cray Fortran 90.

After the new-OS was installed, we again carried out the same benchmark tests for Apogee Fortran 77, Apogee Fortran 90 and SPARC Fortran 77 Ver. 4.0 under the new-OS. The results showed that SPARC Fortran 77 (Ver. 4.0) is generally better than Apogee Fortran 77 and that Apogee Fortran 90 is the best for solving a linear equation.

Key words : benchmark test, Fortran compiler, performance of automatic parallelization

はじめに

地震研究所に EIC (Earthquake Information Center) 計算機が導入されてから、2.5 年が経過した。EIC 計算機とは本研究所ではじめて導入した並列計算機である。導入後時間が経つにつれて、少しずつその性質がわかり、使用上の問題点がはっきりしてきた。

一般にユーザーが並列ジョブを実行する場合でも、ユーザ自身がプログラム中に並列化のためのデレクティブを書くことはほとんどなく、通常はコンパイラの並列化オプションによって、自動的に並列化を行っている。したがって、計算機の運用効率を高めるには、このコンパイラの自動並列化オプションの優劣が大きく影響する。しかし、使用しているコンパイラの中で、Cray Fortran 90 (以下では CF 90 と省略する) は、自動並列化の性能が低く、コンパイラ自身も巨大であり、その上複素数の倍精度 (実数 4 倍精度) が使えないなど問題の多いコンパイラである (桧山・長谷川, 1996 ; 桧山 1996)。したがって Fortran 90 レベル

でこれに代わるコンパイラの導入を検討する必要がある。

一方、EIC 計算機で使用している OS は、これまでは Solaris 2.3 であったが、これを Solaris 2.5 にバージョンアップする予定があった。それにしたがって SPARC Fortran 77 (以下 F 77 と省略) コンパイラも、新しい OS に対応したものの導入を検討する必要が生じた。

我々はこの 3 月に、Apogee Fortran 77 (以下 apf 77 と省略) および Apogee Fortran 90 (以下 apf 90 と省略) の二種のコンパイラを 2 週間テスト導入して、これまで使用していたコンパイラと共に、以下に述べるベンチマークテストを行った。ベンチマークテストとは、性能評価テストのことである。コンパイラを Fortran に限ったのは、表 1 に示したように、EIC 計算機利用の 90% が Fortran 言語であるからである。新しいコンパイラを導入する際には、現在情報センタで提供している二種の数値計算ライブラリ、LibSci および IMSL とリンクできるかどうかをチェックする必要もある。このテストも併せて実施した。

テスト方法とその結果

従来から EIC システムでサポートしているコンパイラ F 77 (Ver. 3.0), CF 90 (Ver. 1.0) と、テストのために導入

1997 年 9 月 10 日受付, 1997 年 10 月 28 日受理.

* 地震予知情報センター, (東京大学地震研究所).

* Earthquake Information Center, (Earthquake Research Institute, University of Tokyo).

表 1. EIC における各コンパイラの利用状況 (1997.1~4 の 4 ヶ月間)

コンパイラ	件数の百分率	CPU 時間の百分率	自動並列化機能
SPARC Fortran 77 Ver. 3.0	70.8	64.6	あり
Cray Fortran 90 Ver. 1.0	20.3	25.3	あり
Apogee C Ver. 3.0	1.5	3.6	あり
SPARC C Ver. 3.0	7.4	6.6	なし

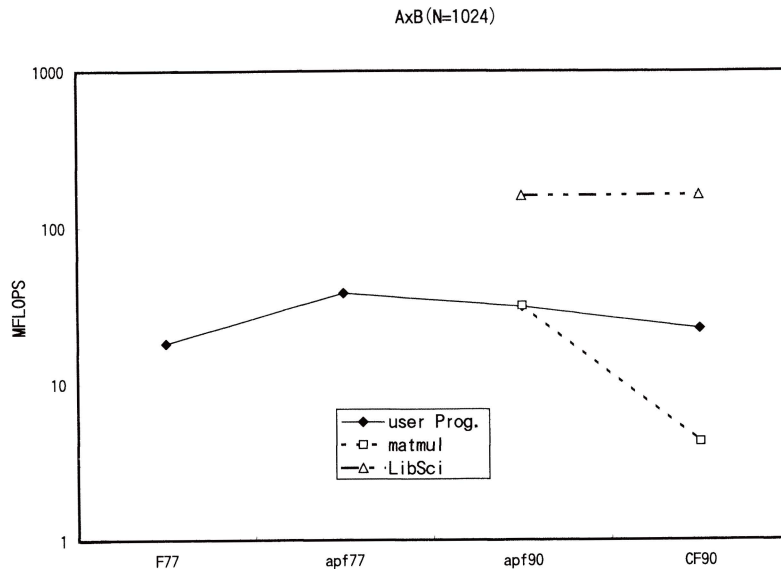


図 1. ユーザプログラム, LibSci, matmul 関数による行列積演算の各コンパイラの並列化性能比較

した apf77, apf90 との 4 コンパイラに対して, 以下の 3 つの場合についてテストを行い, 演算速度, 並列化機能の比較を行った. 並列化機能を調べるためのベンチマークプログラムは各種あるが(長嶋・妹尾, 1996; 関口・小柳, 1993), ここでは, 始めから並列計算機を意識してプログラムされたこれらのベンチマークプログラムに頼らず, ユーザにとってより身近な行列の積と連立方程式の計算を用いた.

[1] テスト 1.

倍精度演算で行列の積を求める.

◆ 方法: 行列の積を求める演算を, 各コンパイラの自動並列化オプションを使って並列計算を行い, 演算速度の比較をする. 並列計算は CPU を 1 台, 2 台, 3 台, 4 台, 6 台を使ったそれぞれの場合を調べる. プログラムは次の 2 つのものを使用する. ひとつは Fortran 77 レベルで書かれたもの, もう一つは Fortran 90 で書き, 行列の積を求めるところは Fortran 90 の組み込み関数 matmul を使用したものである. 計算はすべて倍精度演算で行い, 時間は純粹に行列積演算部分のみを計測する. Fortran 77 のプログラムは NUMPAC の MULMMV (二宮, 1988) を使用することにした.

◆ 目的: 自動並列化機能の優劣を比較し, また, Fortran 90, Fortran 77 のプログラムに対する自動並列化の機能を調べることを目的とした.

◆ 結果: 次数 $n=1024$ の行列の積を作る演算に対し, CPU 4 台で行った場合の結果を図 1 に示す. 縦軸は対数目盛による Mflops の値である. Mflops は演算速度の目安で, 1 秒あたりの浮動小数点演算回数を 100 万回単位で表したものであり, ここでは行列積を求めるための積和の演算回数(積の演算が n^3 , 和演算が $(n-1)n^2$ なので総演算回数は $2n^3$ と見なせる)を, 時間 $time$ (秒) $\times 10^6$ で割ったものである. つまり, 行列の次数 n に対し,

$$Mflops = 2n^3 / (time \times 10^6)$$

である. 同様に図 2, 図 3 には $n=512$, $n=1024$ の各場合に対して CPU の台数を変化させた時の, 結果を図示した. この場合も縦軸は Mflops である.

図 1. から, 以下の 1)~3) が結論付けられる.

- 1) Fortran 77 で作ったプログラムに対しては, apf77 でコンパイルした場合がもっとも演算速度が速く, apf90 によるものがその次に速い.
- 2) apf90 では, Fortran 77 で書かれたプログラムをコンパイルした場合と, Fortran 90 でプログラムを作

り、組み込み関数、matmul を使用した場合とでは、速度はほとんど変わらない。一方、CF 90 に対して matmul を使うと極端に遅くなる。

3) LibSci ライブラリを使用すると、コンパイラに関係なく apf 90 でも、CF 90 でも非常に速い。

図 2 と図 3 からは以下の 4) がいえる。

4) 次数 n が 512, 1024 のとき、CPU の数をふやしても F 77, CF 90 はどちらも演算速度がほとんど変化しない。apf 77 と apf 90 は CPU を増やすにつれ演算速度が速くなるが、CPU6 台の場合にはかえって速度が落ちる。

[2] テスト 2.

連立方程式を解く (単精度, 一部倍精度)

◆ 方法: 連立方程式 $Ax=b$ で、 A が 512 次および 1024 次の密行列の場合に x を求め、演算速度を比較する。算法は行列 A を上三角と下三角行列に分解し、ガウスの消去法で解くというごくありふれた方法を使用する。プログラムは SDECOM (森, 1988) を一部修正して使った。これは Fortran 77 で書かれており、計算は単精度であるが、一部に倍精度を使っている。

◆ 目的: 純粋に計算部分の速度だけを計測したテスト 1 の場合と異なり、行列の右辺および左辺に数値データを代

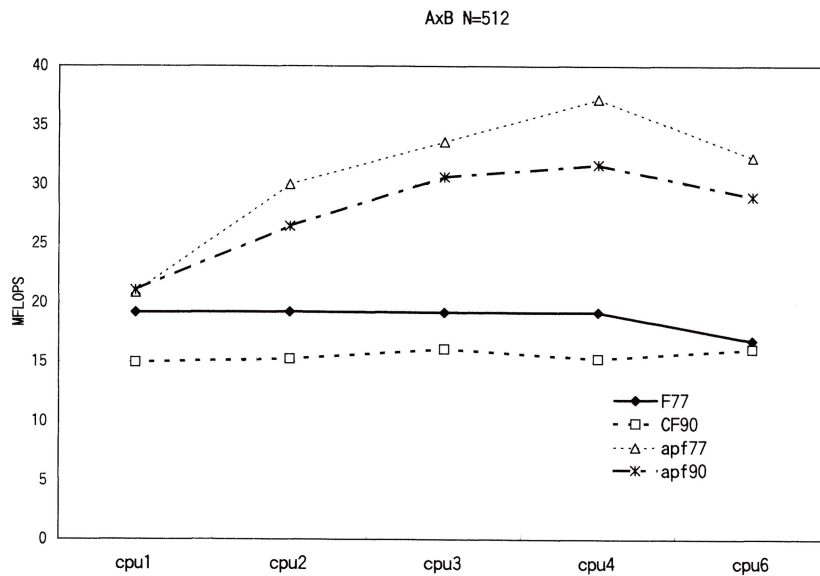


図 2. $n=512$ のときの、各コンパイラの自動並列化性能比較

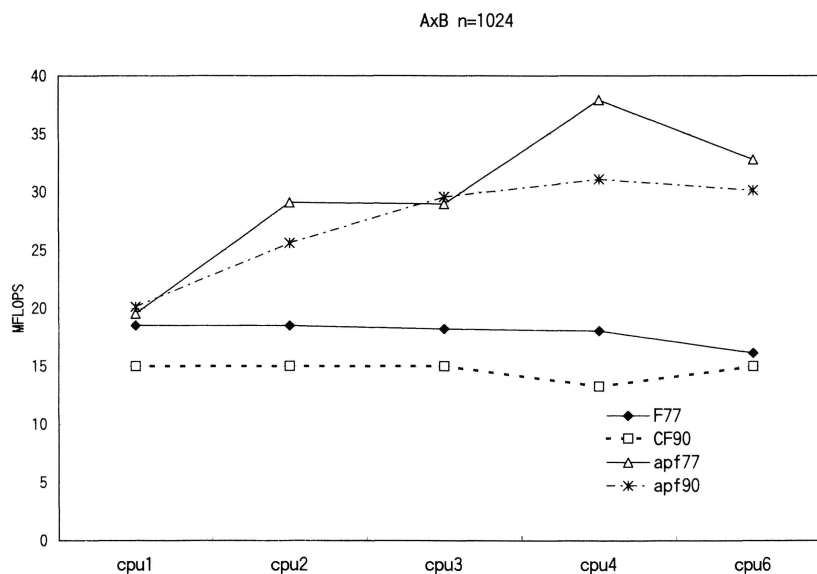


図 3. $n=1024$ のときの、各コンパイラの自動並列化性能比較

入したり、条件数を出力したり、解の一部の数個を出力するような、通常ユーザがしばしば行う操作を含めたパフォーマンスを調べることを目的とした。

◆ 結果：この場合は演算回数を $\frac{2}{3}n^3$ と見積もり、 $Mflops = 2n^3 / (3 \times time \times 10^6)$ としている。

$n=512$, $n=1024$ の場合を F77, apf77 で行った結果を図4, 図5. に示す. ここから

次の結論が得られる.

1) $n=512$ の場合は, F77, apf77 ともに CPU1 台のときに演算速度が最も速く, 並列化の利点がまったくくない.

特に F77 は CPU をふやすと速度の低下が大きく, スワッピングのためと推定される. apf77 の方は CPU1 台のときに最も速く, CPU2 台ではいったん遅くなる. さらに CPU を増すと, 徐々に速くはなるが, CPU1 台のときにはおよばない. いずれの場合も apf77 の方が優れている.

2) $n=1024$ の場合には apf77 には並列化の効果が表われているが, F77 は CPU3 台以上ではほとんど並列化の効果が無い. CPU3 台以上では, F77 は apf77 より常に演算速度が劣る.

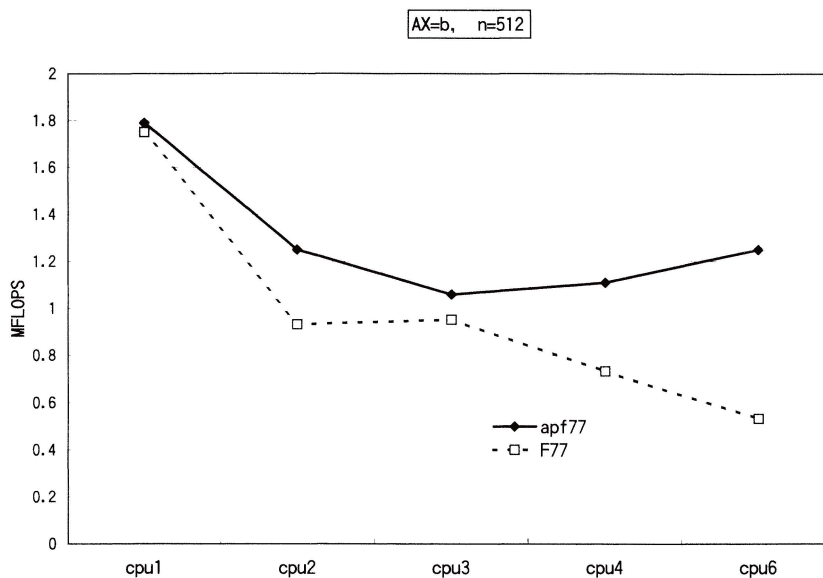


図 4. $n=512$ のときに, 連立方程式を解く場合の自動並列化性能比較

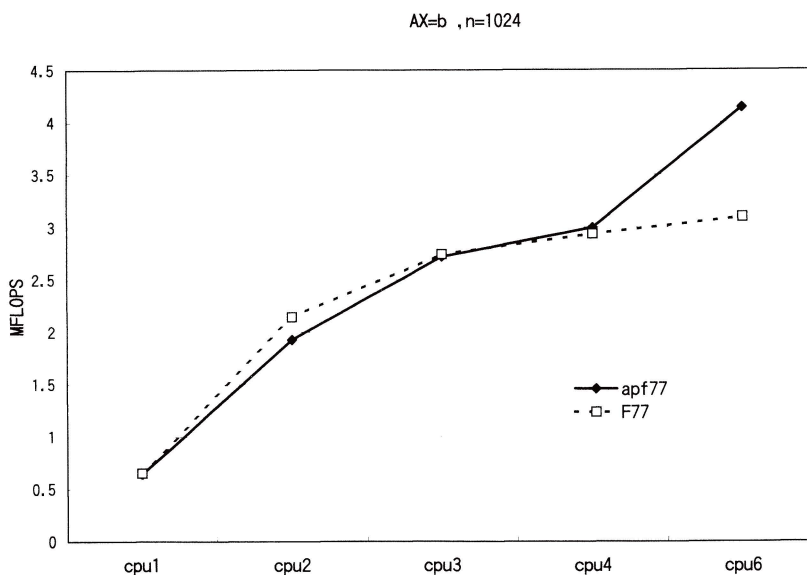


図 5. $n=1024$ のときに, 連立方程式を解く場合の自動並列化性能比較

[3] テスト 3.

情報センターで提供している数値計算ライブラリー (LibSci, IMSL) に対し, apf90 と apf77 がリンク可能かどうかをしらべる.

◆ 結果: テスト 1. で結果を得たように LibSci とのリンクは問題がなかった. しかし IMSL はリンクできなかった. IMSL を利用するためには, F77 を残す必要があると思われた.

テストより得た導入策

以上のテスト結果から, 我々は新システムに対し, 以下の Fortran コンパイラを導入する方針をとった.

- 1) CF90 は提供を止め, 代わりに apf90 を導入する.
- 2) F77 に比べ apf77 の自動並列オプションは非常に優れているので, 新たに導入する.
- 3) F77 は Ver. 4.0 にバージョンアップしたものを導入する.

新システムにおけるテスト

6月に OS を Solaris 2.5 に変更し, それに伴いコンパイラは表 2 のようになった.

そこで我々は, 新システム下で, SPARC Fortran 77 Ver 4.0 (以下 f77 V4.0 または f77 と省略し Ver. 3.0 の F77 と区別する) Apogee Fortran 77/90 (旧システム時代のものは apf77-1, apf90-1 と略し, 新システム下のものは, apf77-2, apf90-2 と略し区別する.) の 3 つの Fortran コンパイラに対し, 旧システムで行ったと同じテスト 1 から 3 を行った.

[1] テスト 4.

行列の積を求める計算をテスト 1 と同様に行った. 図 1 に対応して図 6 に次数 $n=1024$ の行列の場合の結果を示す. その結果, ライブラリ LibSci を使う場合が, コンパイラと無関係に最も速いことがわかった. User Prog (Fortran 77 で書いたプログラム) については, f77 が最も速くなっている. 数値的には, $n=512$ のとき, F77 で 19.17 Mflops だった計算が f77 では 49.12 Mflops になり, また $n=1024$ では, F77 で 18.05 Mflops だったものが, 47.52 Mflops と約 2.5 倍の速度向上になっている.

[2] テスト 5.

連立方程式を解く場合で, テスト 2 に対応するものを新システムで行った. 旧システムの F77, apf77-1 と新システムの f77, apf77-2 を比較するために, $n=512$, $n=1024$

表 2. 新システムで提供されたコンパイラ一覧

言語	変更・バージョンアップ	自動並列化機能
SPARC Fortran 77	Ver 3.0→Ver. 4.0	あり
Apogee Fortran77	新たに導入, Ver. 4.0	あり
Apogee Fortran 90	CF90 の代替新に導入, Ver. 4.0	あり
SPARC C	Ver 3.0→Ver. 4.0	Ver. 4.0 以降あり
Apogee C	Ver 3.0→Ver. 4.0	あり

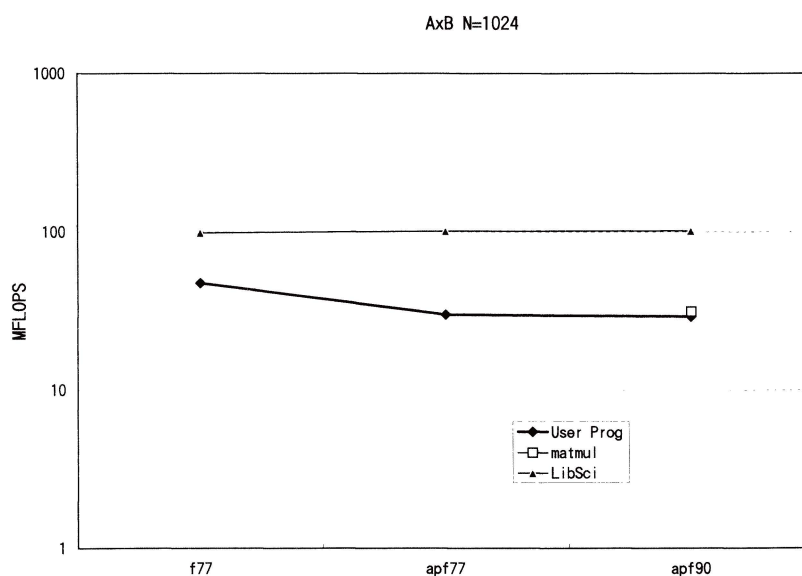


図 6. 新システム上でのユーザプログラム, LibSci, matmul 関数による行列積演算

の場合に CPU の台数を変化させて計算速度を調べた結果を、図 7、図 8 に示す。ここでは $n=512$ の場合は、CPU1 台のときがいずれのコンパイラについても最も効率がよく、それ以上に CPU が増加しても、効率は変わらない。つまり並列化の利点は全く生かされていない。しかし演算速度は、f77 が断然速い。 $n=1024$ の場合は、新システムのコンパイラの方が、旧システムのコンパイラよりいずれについても速かった。これは OS 自身のパフォーマンス向上が影響しているためと考えられる。f77 以外のコンパイラは、CPU 台数が増加するにつれて、パフォーマンスが向上するが、f77 では逆に遅くなる。しかしこの場合でも他の

コンパイラと比較すれば、f77 がもっとも速い。

また、図 9 にはこの計算を倍精度、一部を 4 倍精度で行った結果を示した。ここでは次の 3 つのコンパイラ、apf77-2、f77、apf90-2 を用いて行っており、次数は $n=1024$ の場合について示している。この場合は apf90-2 が断然速い。

同様に $n=512$ のとき、または単精度計算のテストでも、apf90-2 の自動並列化の性能は最も優れており、次いで f77 であった。apf90-2 が apf77-2 に比べて非常に良いのは以下のような理由によると考えられる。apf90-2 のプリプロセッサはいったん Fortran 90 を、Fortran 77 のソー

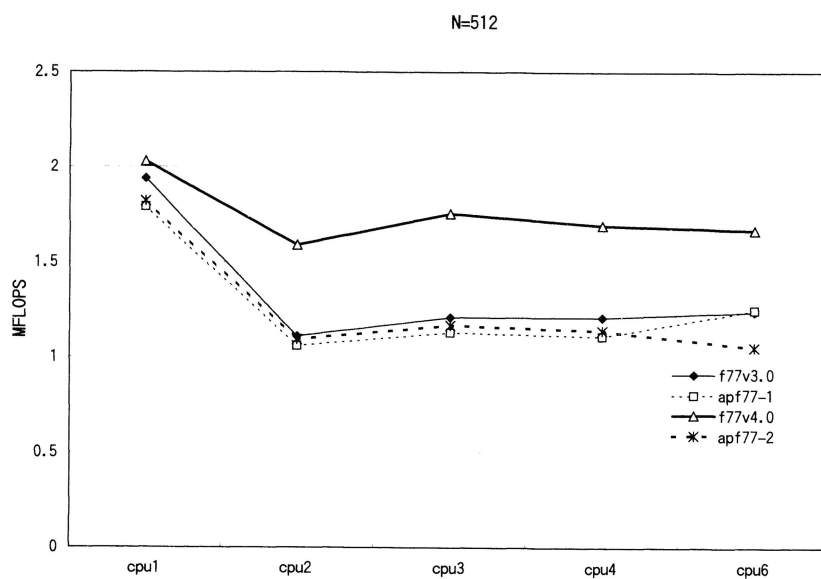


図 7. 新システム・旧システムの Fortran 77 コンパイラの自動並列化性能比較 ($n=512$)

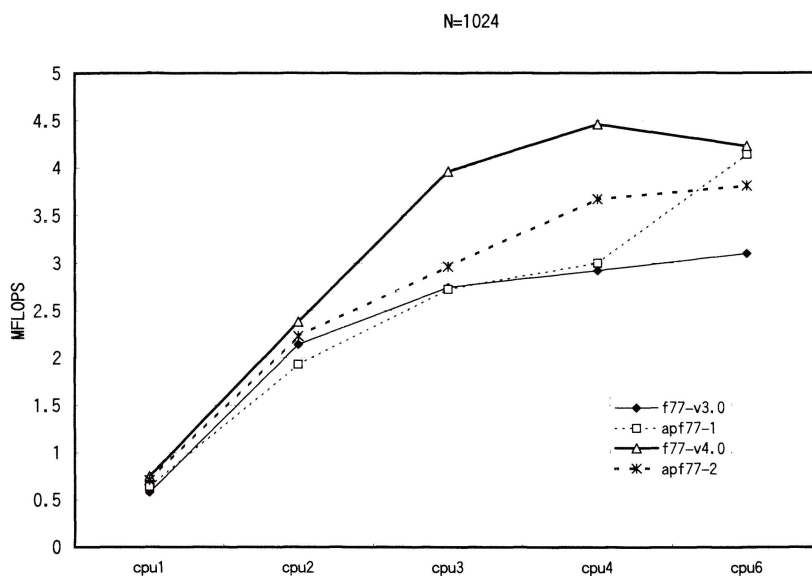


図 8. 新システム・旧システムの Fortran 77 コンパイラの自動並列化性能比較 ($n=1024$)

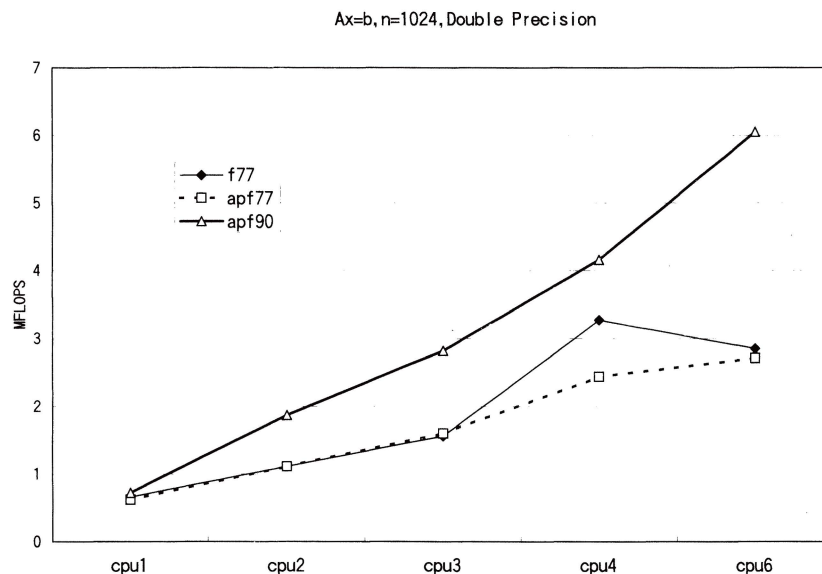


図 9. $n=1024$ のときに、連立方程式を倍精度で解いた場合の性能比較

スコードに変換する。その後、Fortran 77 のプリプロセッサが働いて、Fortran 77 の並列化したコードを出すという 2 段階構成である。従ってたとえプログラムが Fortran 77 レベルで書かれていても、それを apf 90-2 でコンパイルすればこのプリプロセッサが非常に優れた最適化した Fortran 77 のソースコードを出したために、apf 90-2 の方がよい結果を得たわけである。apf 77-2, apf 90-2 は CPU 台数が増加するにつれて、パフォーマンスが向上するが、f77 では CPU が 6 台になると逆に遅くなる。一般的には、次数の大きな行列を取り扱う場合には、倍精度以上の計算が必要になる。そういう意味でも、apf 90-2 が断然速いということは、重視されるべきことである。

[3] テスト 6.

数値計算ライブラリ LibSci や IMSL にリンクして使えるか否かを調べる。結果は以下のとおりであった。

- 1) apf 77/apf 90 と IMSL はリンクできなかった。
- 2) f77 で自動並列化のための '-autpar' オプションを指定すると、LibSci とのリンクエラーになる。

1) Apogee コンパイラ apf 77, apf 90 と IMSL とのリンクのエラーは、テスト 3 で述べたように、IMSL が F77 を使ってコンパイルしたロードモジュールの形で導入されている事実から、事前に予想は出来た。しかし、2) については全く予想外のエラーであった。

結論と今後

今回、一連のコンパイラの自動並列化機能のベンチマークテストを行った結果、現在の新システムでは f77 (Ver. 4.0) はこのバージョンで改良され、優れた自動並列化の機

能を持っていることがわかった。また、たとえ Fortran 77 レベルで書かれたプログラムでも apf 90 でコンパイルすると、テスト 5 で示したように優れた並列化コードが出ていることも解った。テスト 6 で生じた数値計算ライブラリとのリンク・エラーは、IMSL では f77 を使い、LibSci では apf 77, apf 90 を使えば避けることができる。コンパイラ間にそれほど大きな性能の開きがないので、こうした使い分けをしても問題はないと思われる。また、CPU を何台に振り分けるかの指定は、f77 については 4 台程度が望ましいことも解った。そして、我々はこれらの結果をマニュアル「EIC システム利用法」(情報センター, 1997) に反映させることができた。今後は更にベンチマークテスト用プログラムを拡充させ、システム更新の際や運用に利用する予定である。

謝辞：この一連のテストでは、日立 SE の高橋英司氏には大変お世話になった。ここに御礼申し上げる。

文 献

- 松山澄子, 1996, IMSL ライブラリのコンパイラ環境による並列化パフォーマンスの比較, 第 7 回日本ビジュアルニューメリックユーザ会コンファレンス予稿集, 34-40.
- 松山澄子・長谷川秀彦, 1996, 並列計算機 CS6400 における数値計算ライブラリの現状と問題点, 技術研究報告, 1, 65-71.
- 情報センター, 1997, EIC システム利用法.
- 森 正武, 1988, 数値計算プログラミング, 岩波書店, 52-78.
- 長嶋雲兵・妹尾義樹, 1996, 科学技術計算のベンチマーク, 情報処理, 37, 738-744.
- 二宮市三, 1988, ライブラリー・プログラム利用の手引き, 数値計算編: NUMPAC, 1, 213-214.
- 関口智嗣・小柳義夫, 1993, スーパーコンピュータの性能評価の現状, 応用数理, 3, 27-38.