VPP700/56 の演算性能

渡部 善隆 *

新スーパーコンピュータ VPP700/56 の運用にあたり，センターでは何本かのテストプログラムを走らせ，データを収集しました．本稿では，それらの実測データの中から， Fortran 90 の組み込み関数と，メモリーアクセス，および SSL II/VPP の代表的なサブルーチンについて得られた結果を紹介します．なおプログラムはすべて Fortran の倍精度 (64 ビット) 実数データ型の演算を行なうものです．

1 Fortran 90/VP

1.1 Fortran 90 の組み込み関数の性能

Fortran 90 では，ベクトルの内積，行列積などのよく使う手続きが新たに組み込み関数として標準でサポートされるようになりました ([3])．この節では，これらの組み込み関数が速度に関して “使いものになるか” を調べてみました．

•行列積 (MATMUL)

Fortran 90 には，組み込み関数 MATMUL が標準でサポートされています ([3])．アルゴリズムは処理系依存です．ブラックボックスとなるため気味が悪いことは否定できませんが，従来の FORTRAN 77 のサブルーチンの様に整合寸法や次元数をいちいち入力する手間が省けることは魅力です．

テストでは次元 n の正方行列 A, B を乱数発生の関数 RANDOM_NUMBER で作成しました．また，比較のため富士通提供の SSL II/VP([1]) の DVMGGM と， 2 回アンローリングを施した自作の行列積サブルーチン MYLIB を用意しました． CPU 時間の測定は組み込み関数 CLOCKV を手続きの前後に埋め込んで行ないました．図 1 は演算数を $2n^3$ 回とした FLOPS 値をプロットしたものです．

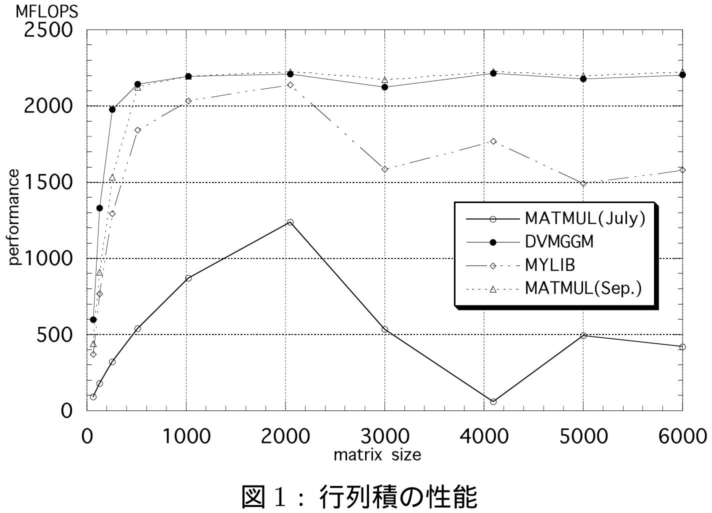

図 1： 行列積の性能

*九州大学大型計算機センター・研究開発部 E-mail： watanabe@cc.kyushu-u.ac.jp

“MATMUL(July)” はレベルアップ以前に測定したデータ，その他は 1997 年 9 月 2 日の結果です．

Fortran 90/VP のレベルアップにより，MATMUL の性能が DVMGGM と同等にまで向上したことがわかります．

●ベクトルの内積 (DOT_PRODUCT)

次の結果は，同じく Fortran 90 で組み込み関数となったベクトルの内積を求める関数 DOT_PRODUCT と，自作の関数 (何もテクニックは使っていません) MYLIB とを比較したものです．横軸は対数とし，ベクトルの長さ n を意味します．縦軸は演算を $n(n-1)$ 回とした FLOPS 値をプロットしています．図 2 がベクトル演算，図 3 がスカラー演算の結果です．ここで「ベクトル演算」とは，省略値翻訳オプションを採用して翻訳・実行した結果 (従って自動ベクトル化されたオブジェクトを出力します)，また，「スカラー演算」とは，ベクトル化を抑止するオプションを付加して翻訳・実行した結果 (従って，スカラーユニットの演算性能を調べることになります) を意味しています．

測定は 1997 年 9 月 2 日に行ないました．“○”, “●” の部分が測定した n に対応しています．

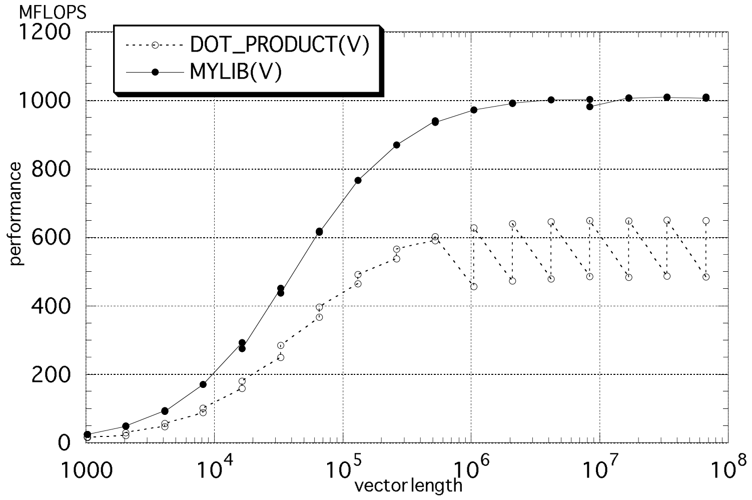

図 2： ベクトルの内積の性能 (ベクトル演算)

グラフが “ふらふら” しているのは，バンクコンフリクト (cf.[5]) の影響と思われます (意図的に n を 2 のベキ乗になるような値を設定しています)．

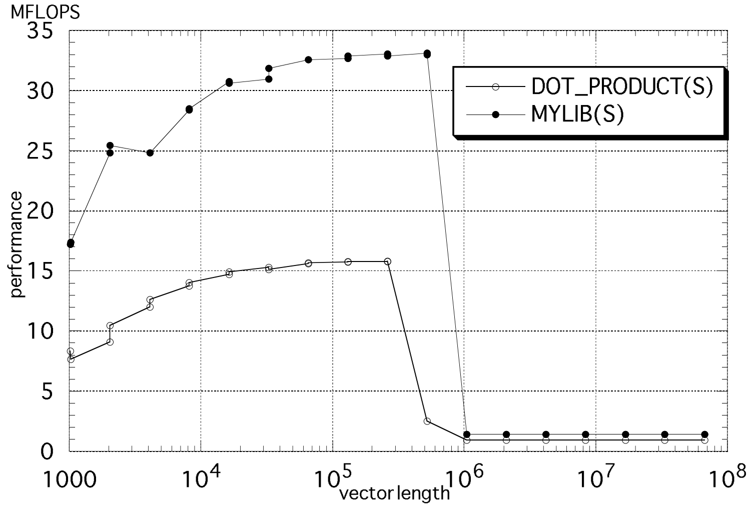

図 3： ベクトルの内積の性能 (スカラー演算)

スカラー演算で極端に性能が悪くなった原因は今のところわかりません．

両方のテストともに，DOT_PRODUCT は自作の関数に差をつけられています．今後のチューニングに期待します．

●行列の転置 (TRANSPOSE)

次は，行列を転置する組み込み関数 TRANSPOSE と自作の関数 (何もテクニックは使っていません) MYLIB とを比較しました．テストは次元 n の正方行列を RANDOM_NUMBER で生成し，転置する手続きに要する CPU 時間を測定するというものです．図 4 がベクトル演算，図 5 がスカラー演算の結果です．測定は 1997 年 9 月 2 日に行ないました．

行列を転置する演算は n が $2n$ と $2n+1$ に設定して行ないました．それぞれの場合でかなりの差が出ましたので，分けてプロットしています．

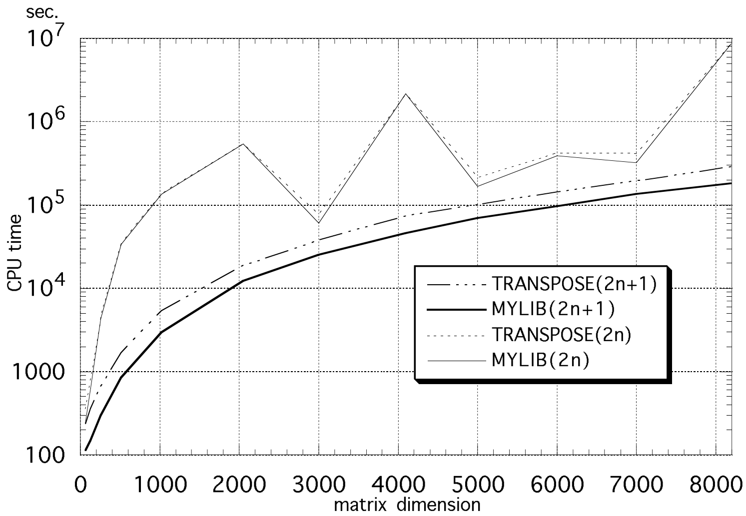

図 4： 行列の転置の性能 (ベクトル演算)

性能に差が出た原因はバンクコンフリクトの影響だと思われます．特に n が 2 のベキの場合極端に性能が劣化することがわかります．ベクトル化プログラミング (cf.[2]9 章) をする際は，自分でプログラミングした方が今のところ速いという結果になりました．

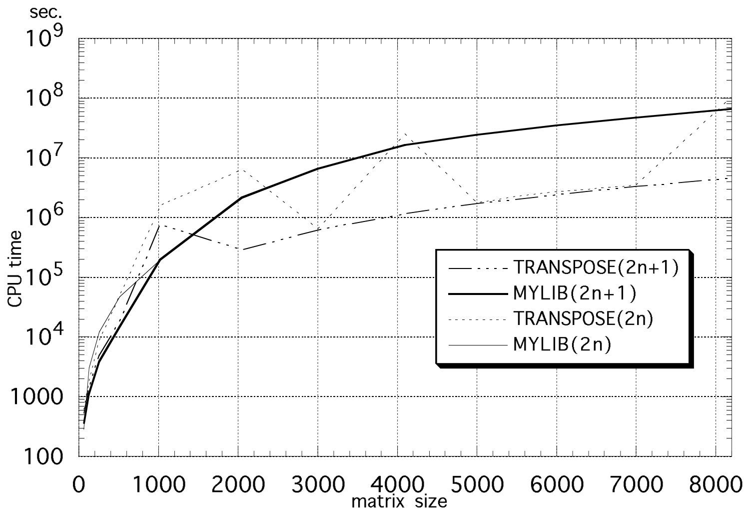

図 5： 行列の転置の性能 (スカラー演算)

スカラー性能では TRANSPOSE の方が速いようです[1]．

1.1.1 最大値 (MAXVAL)

最後は，ベクトルの最大値を求める関数 MAXVAL を自作の関数 (何もテクニックは使っていません) MAX と比較した結果です． Fortran 90 には最小値を求める MINVAL もあります．図 6 がベクトル演算，図 7 がスカラー演算の結果です．測定は 1997 年 9 月 2 日に行ないました．

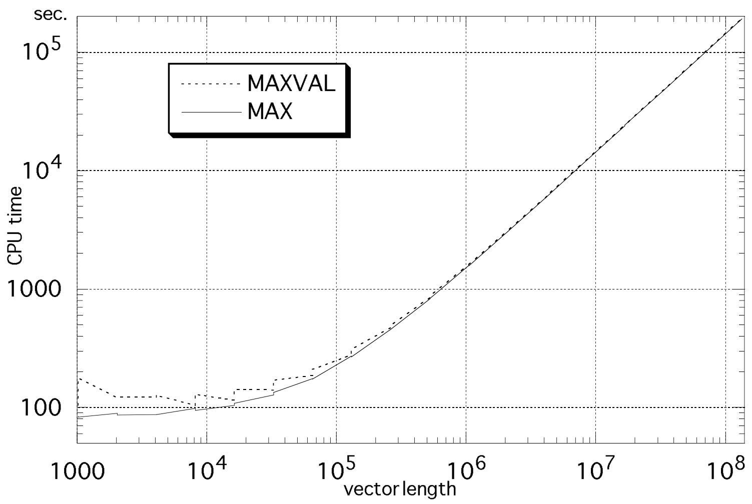

図 6： ベクトルの最大値 (ベクトル演算)

こちらは大きな差はないようですが，ベクトル長が短い時の MAXVAL は “冴え” がありません．

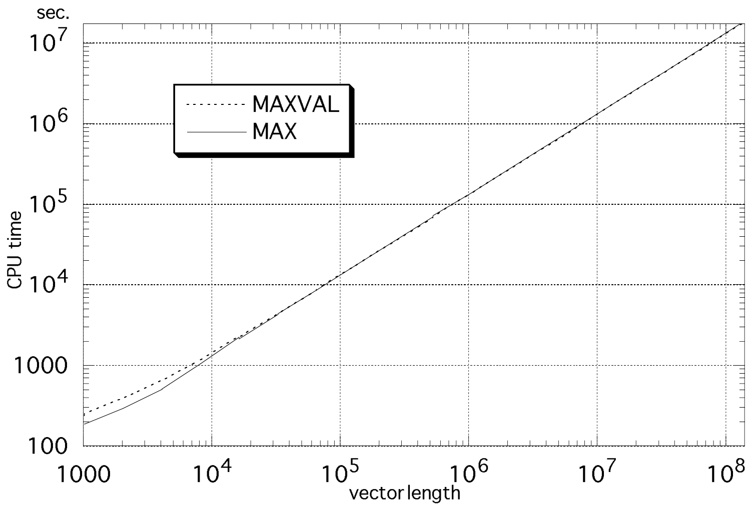

図 7： ベクトルの最大値 (スカラー演算)

Fortran 90 の組み込み関数の性能を調べてみたところ，行列積はレベルアップによりピーク性能に近い値を出すようになり，速度の面から安心して利用できそうです．その他の組み込み関数は，それほど性能が出ていません．ぎりぎりのチューニングをする場合には注意が必要です．

[1] スカラー演算で勝ってもあまり意味がないのですが...

1.2 メモリーアクセス性能

Fortran で行列にアクセスする場合，行方向の連続したアクセスが最も効率良い方法です．C の場合は逆で列方向のアクセスが効率良くなります．

ここでは，n 次元の正方行列を以下の方法でアクセスした場合に要する CPU 時間の合計を測定してみました．

`COLUMN`	列ごとに行方向を連続してアクセス
`ROW`	行ごとに列方向を連続してアクセス
`R-COLUMN`	列ごとに行方向をランダムに異なりなくアクセス
`R-ROW`	行ごとに列方向をランダムに異なりなくアクセス

図 8 がベクトル，図 9 がスカラーで翻訳・実行した値をプロットしたものです．図では “`COLUMN`” を 1 とし，他のアクセスがその何倍になるかを示しています．測定は 1997 年 9 月 2 日に行ないました．

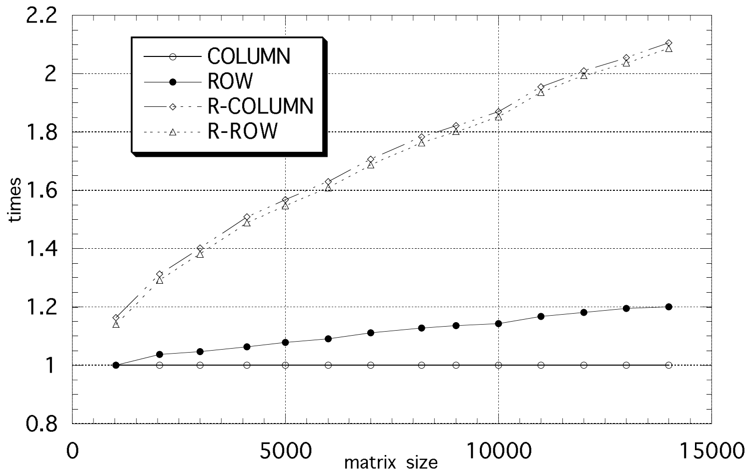

図 8： メモリーアクセス (ベクトル演算)

ベクトル演算では，行 / 列の連続アクセスに思った程の差は出ませんでした．むしろアクセスの “連続性” が大切なことがわかります．

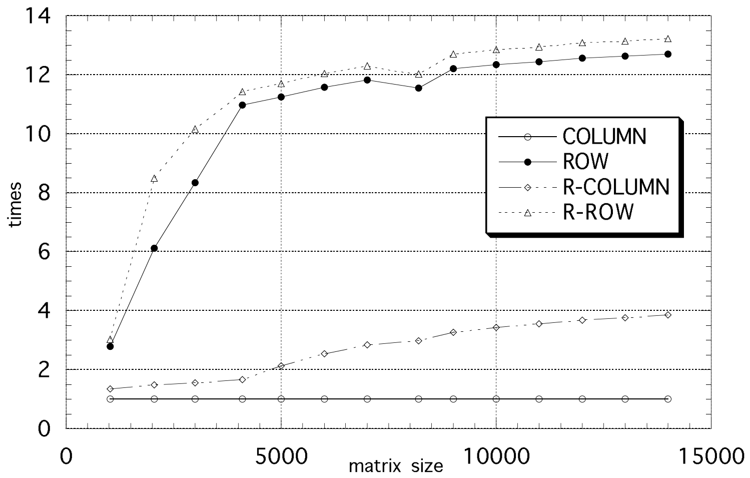

図 9： メモリーアクセス (スカラー演算)

スカラー演算では行 / 列のアクセス方向によって大きな差が出ることがわかります．ワークステーションやパーソナルコンピュータでプログラムを組む場合，この性質を知っていると有利です．

2 Fortran 90/VPP

並列プログラムの演算性能として，SSL II/VPP の代表的なサブルーチン性能を性能を測定したデータを幾つか紹介します．

2.1 56PE でのピーク性能試験

1996 年 12 月 5 日，富士通沼津工場で VPP700/56 の現地テストを行ないました．その際，ピーク性能試験として，行列積演算の Fortran プログラムを 56PE で並列実行しました．プログラムは，次数 n の倍精度実数データ型の正方行列 $A,\ B$ の積を計算し，結果を次数 n の正方行列 C に格納するものです．

行列積演算には SSL II/VPP の `DP_VMGGM` を用います[2]．行列のデータは各要素を $A_{ij},\ B_{ij}$ とするとき，

$$A_{ij} = B_{ij} = \left(\frac{2}{n+1}\right)^{\frac{1}{2}} \sin\left(\frac{ij\pi}{n+1}\right)$$

で定義します．なお π は $4\tan^{-1}1$ とします．

このように定義した理由は，行列積演算の精度の確認として理論的に $AB = E(E$ は単位行列$)$ となることを用いて，行列 C に対する要素を C_{ij}，E の要素を E_{ij} とするとき，最大行和ノルムの意味での誤差 $(||E||_\infty = 1$ より絶対誤差かつ相対誤差となります．$)$

$$||C-E||_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |C_{ij} - E_{ij}|$$

が知りたかったからです．経過時間はウォールクロックにより行列積演算の開始から結果格納までを測定しました．次元 n の大きさと $A,\ B,\ C$ に必要な記憶領域および経過時間，誤差は表 1 の通りです．経過時間は小数点第 1 位以下を，誤差は小数点第 4 位以下を切捨てています．

表 1：56PE でのピーク性能試験結果

n	記憶領域	経過時間	誤差
18000	7.2GB	105sec.	0.138e-09
40000	35.7GB	1135sec.	0.393e-09
55000	67GB	2865sec.	0.713e-09
60000	80GB	3683sec.	0.681e-09
65000	94GB	5197sec.	0.819e-09

個人的な感想としては，演算数の増大によって着々と増え続ける誤差の方が気になります．もちろん，誤差ノルムの計算自体に混入する丸め誤差も考慮に入れる必要がありますが，大規模な数値計算を行なう場合の丸め誤差の評価は今後きちんと考える必要があるのではないでしょうか．

図 10 は演算数を $2n^3$ とした GFLOPS 値をプロットしたものです．“`Theoretical_Peak`” は，56 台での理論ピーク性能値です．

[2]普通に行列積のプログラムを組むと数行で済むのですが，ソースリストを見たところ，A4 判 29 頁におよぶ長大なものでした．

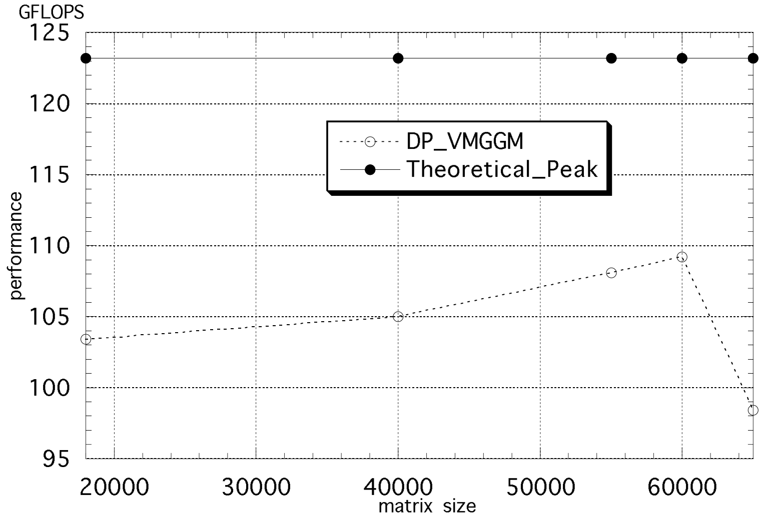

図 10：56PE でのピーク性能試験結果

$N = 65000$ で処理性能が低下した原因をメーカーに問い合わせたところ，“たまたま” この値に対して性能低下を起こす要因があり，DP_VMGGM のソースの修正により，$N = 60000$ 以上の性能が出るとの報告を受けました．しかし，56PE の性能試験を行なうためにはすべての PE の割り当てを変更する必要があるため，運用を開始した現在ではなかなかテストできません．

2.2 台数効果

現在，利用者が使用できる最大 PE 台数は 32 です．以下，4 つの SSL II/VPP サブルーチンに対し，PE 台数を変化させ，台数効果を調べてみました．測定は 1997 年 4 月～ 6 月に行ないました．

FLOPS 値は，Analyzer([4]) の浮動小数点演算数の採取機能 PEPA(PE Performance Analyzer) の値を信用して用いています．

○行列積

図 11 は 56PE でのピーク性能試験で使用した同じプログラム (サブルーチンDP_VMGGM) の $n = 8500$ に対し PE 台数を変化させたものです．“Theoretical_Peak” は，その台数での理論ピーク値です．

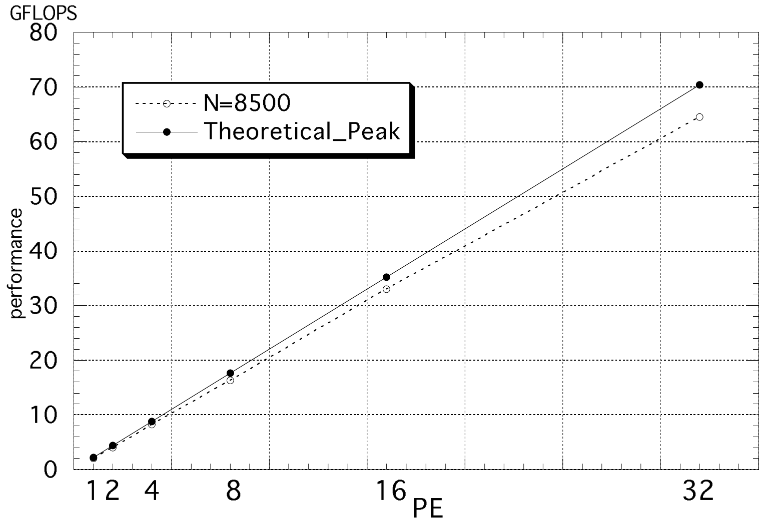

図 11：DP_VMGGM の台数効果

$N = 8500$ にした理由は，ぎりぎり 1PE で実行可能だからです．各 PE 台数分のメモリーの上限に迫るように N を変化させたデータも採取しましたが，それほど性能は変わりませんでした．

図 12 は，実スパース行列と実ベクトルの積を求める DP_VMVSD の性能を示すものです．実スパース行列は，領域 $\Omega = [0,1]^3$ を各方向 $N+1$ 等分割したときの楕円型偏微分演算子を中心差分を用いて離散化した時に得られるものです．[1] による対角形式格納法により格納されています．

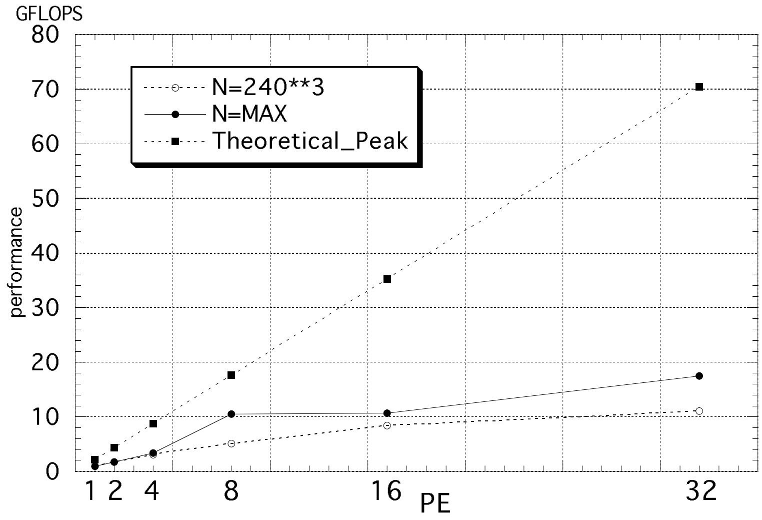

図 12：DP_VMVSD の台数効果

“N=MAX” は，各次元の分割数 N を実行可能な記憶領域の上限近くに設定して採取した値をプロットしたものです．

○連立 1 次方程式

図 13 は密実行列に対する連立 1 次方程式 DP_VLAX の性能を示すものです．行列は DP_DVMGGM と同じものを用いました．

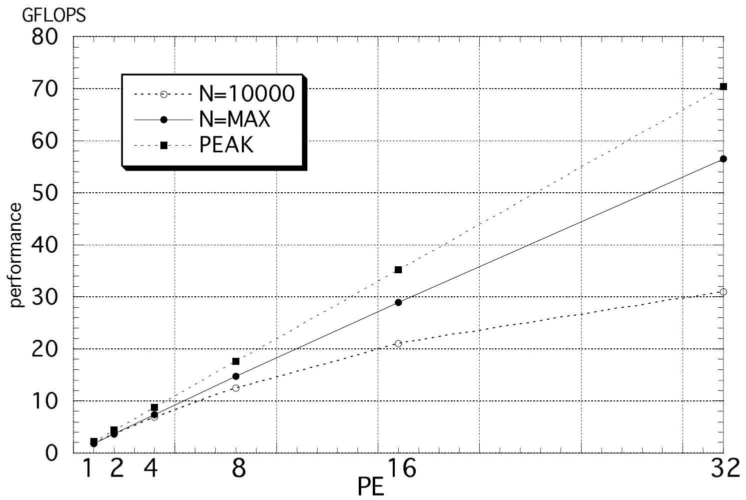

図 13：DP_VLAX の台数効果

“N=MAX” の意味は先ほどと同じです．

図 14 は非対称または不定値のスパース行列に対する連立 1 次方程式 (MGCR 法) DP_VCRD の性能を示すものです．行列は DP_VMVSD と同じものを用いました．

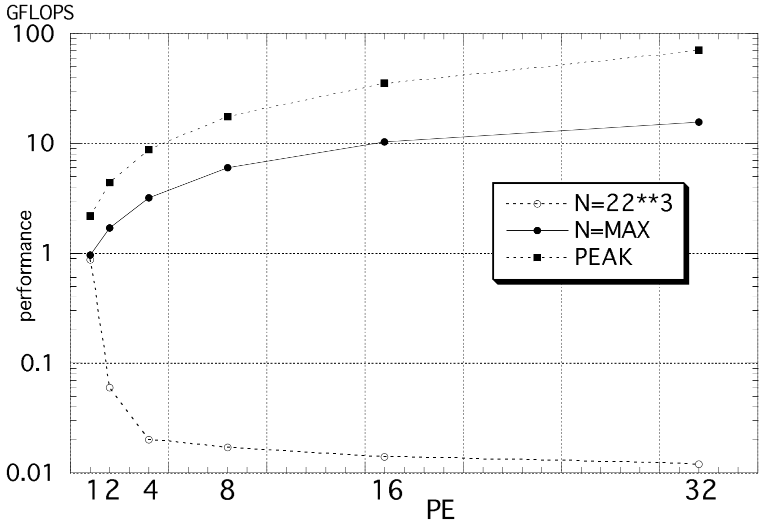

図 14 : DP_VCRD の台数効果

“N=MAX” の意味は先ほどと同じです．このテストでは，N(ここでは分割数です) の値を小さめに設定してみました．

$N = 22$(従って方程式の次元は 22^3) の場合に方程式を解く時間は数秒です．しかし，PE が増えるに従い，浮動小数点演算に要する時間よりも PE 間のデータ転送にかかるコストが増大するため，性能は悪くなっていきます．このような小規模な問題では PE 台数を増やしてもあまり意味がないことがわかります．

参考文献

[1] FIJITSU SSL II 拡張機能使用手引書 (科学用サブルーチンライブラリ), 99SP–4070, 富士通株式会社 (1991)．

[2] VPP700/56 利用の手引 (第 1.0 版)， 九州大学大型計算機センター・プログラムライブラリ室 (1997).

[3] M. Metcalf, J. Reid (西村 恕彦, 和田 英穂, 西村 和夫, 高田 正之 訳) : 詳解 Fortran 90, bit 別冊, 共立出版 (1993).

[4] UXP/V アナライザ使用手引書 V12 用, J2U5–0131, 富士通株式会社 (1997).

[5] 島崎 眞昭 : スーパーコンピュータとプログラミング, 計算機科学 / ソフトウェア技術講座 9, 共立出版 (1989).