

京都大学大型計算機センターのベクトル計算機システム FACOM VP 100 とある種の線形計算ライブラリのベクトル化について

島崎 真昭

(京都大学大型計算機センター)

1. まえがき

近年超高速計算機が科学・技術の研究に必要不可欠との認識が高まり、超高速計算機に対する関心が強まっている。⁽¹⁻³⁾長い間 CRAY-1, CDC Cyber 205に代表されるいわゆるスーパーコンピュータは米国製のみで、我国の大学の関係者は民間の施設を利用するか、外国の研究機関で利用するほかなかった。しかし1982年夏以後我国計算機製造会社より、続々とこれらのスーパーコンピュータを上回るような新製品が発表され(1982年7月 FACOM VP 100/200, 1982年8月 HITAC S810 model 10, 20, 1983年4月 NEC SX-1, SX-2), 全国共同利用大型計算機センター関係でも、東京大学大型計算機センターに HITAC S-810 model 20が1983年10月に設置され、京都大学大型計算機センターに FACOM VP 100が1984年4月に設置され、大学関係者も手軽にスーパーコンピュータを利用できるようになっている。この種のスーパーコンピュータはパイプライン方式のベクトル計算機で、従来の汎用機と性格が異なり、ピーク性能が高いが、そのピーク性能を引き出すのは必ずしも容易でないという問題がある。従来の汎用機では、プログラミングの違いによる実行速度の違いは、3倍程度であったが、スーパーコンピュータでは10倍違うということもあり得る。理想的には特別に意識せずに作成したプログラムも、最高性能で実行できるように、計算機が必要ならばプログラムの変更、修正を行ってこれればよい。このためには自動ベクトル化と呼ばれる技術が重要で、事実最新のベクトル計算機では自動ベクトル化の範囲が従来のものと比較して拡大され、ベクトル計算機の汎用性が飛躍的に高められた。この技術は日進月歩の状態にあり、特に我国の FORTRAN 自動ベクトル化技術は世界最高水準にあると考えられる。しかしこの技術が如何に進歩しても、数値計算法、アルゴリズムの問題は本質的に残る。1982年に開かれた米国 NSF での“科学・工学に於ける大規模計算”に関するパネル⁽³⁾の結論の提言でも、スーパーコンピュータを効果的に能率良く使用するために必要な数値解析、ソフトウェア、アルゴリズムの研究を強化すること、科学・技術計算のための人材を養成することの必要性が強調されている。欧米にはベクトル計算機利用について約十年の経験と実績がある。昨年夏カナダのウォータールーで開催された線形計算に関する Gatlinburg シンポジウムで得た印象では、並列計算・ベクトル計算に関する関心が一段と強まっているようであった。本稿では京都大学大型計算機センターに設置されている FACOM VP 100 を数値解析の観点から概観し、資料によって CRAY-1, Cyber 205 と比較しつつ、その特徴の抽出を試みる。

2. 京都大学大型計算機センターのベクトル計算機システムとそのハードウェア

京都大学大型計算機センターに設置されているベクトル計算機 FACOM VP 100 のシステム構成を図1に示す。主記憶装置は1984年4月のシステム導入時は32 MBであったが、1984年8月に増強され、64 MB となっている。ベクトル処理装置のブロック図を図2に示す。またカタログ的仕様を表1に示す。

VP 100 はスタンドアロンの計算機として使用することも可能であるが、京都

大学大型計算機センターでは、汎用計算機システム FACOM M382 のバックエンドプロセッサとしており、M382 を経由して入力されるバッチ・ジョブの処理に VP 100 は専念している。大容量記憶装置(MSS: Mass Storage System)には VP 100 からは直接アクセスすることが出来ないが、複数のジョブ・ステップを組み、まず、最初 M382 で MSS から共有の磁気ディスク装置にデータを転送し、次のジョブステップで共有ディスク装置のデータを用いて計算し、結果を共有のディスク装置に書き、最後のジョブ・ステップで、データを MSS に送ることもできる。

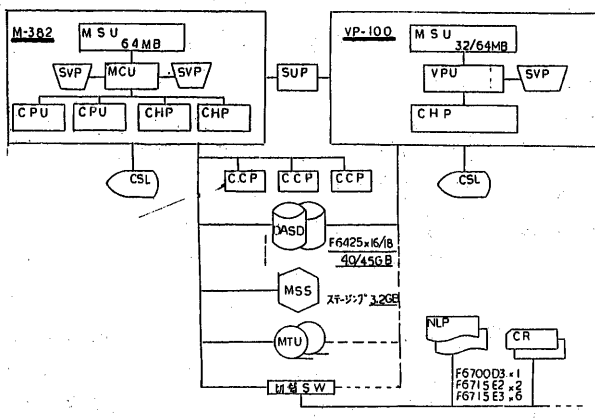


図1 京都大学大型計算機センターの主システム

表1 ベクトル計算機仕様

	CRAY-M1	Cyber 205	VP 100
最大性能(MFLOPS)	160	400 (64ビット) 800 (32ビット)	267
マシンサイクル	12.5ns	20ns	7.5ns
データ形式(ベクトル命令)			
論理型	64ビット	32, 64ビット	1, 64ビット
固定小数点型	64 "	32, 64	32 "
浮動小数点型	64	32, 64	32, 64 "
主記憶装置			
最大容量	32MB	128MB	128MB
インターリーブ(max)	16	128	128
ロード/ストアパス	1本	3本x4	2本
最大転送速度	8バイト/12.5ns	96バイト/20ns	32バイト/15ns
演算パイプラインピッチ	6	2本x4	3本
演算器性能(隔)			
加算	80	200 (64ビット) 400 (32ビット)	183
乗算	80	200 (64ビット) 400 (32ビット)	183
除算	25	33	19
ベクトルレジスタ	4Kバイト	なし	32Kバイト

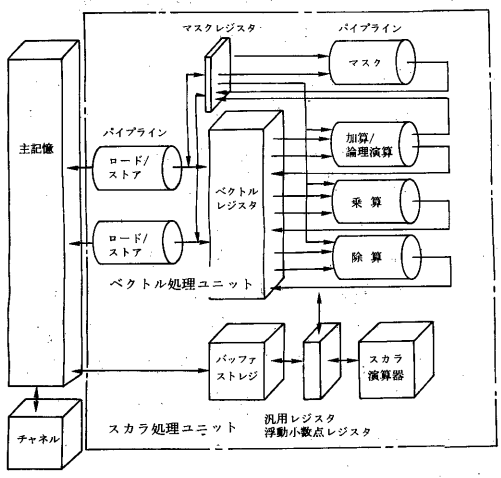


図2 ベクトル処理装置

ベクトル計算機の分類上の一つの観点はベクトルレジスタの有無でCRAY-1, Cyber 205はそれぞれの代表といえる。即ちCRAY-1はレジスタ-レジスタ型のベクトル計算機であって、ベクトル演算の対象となるデータはまず主記憶よりベクトル・レジスタにロードされる。パイプライン演算器にはデータはベクトル・レジスタより供給され、演算結果もベクトル・レジスタに入る。必要に応じて、ベクトル・レジスタより主記憶に送られる。ベクトル・レジスタは64個分のデータが収容でき、それ以上のベクトル長データの演算には64毎にいわゆる立ち上がり時間(演算器にデータが加えられはじめてから最初のデータが出て来る迄の時間)が必要となる。最大主記憶は32MBであり、仮想記憶方式ではない。

一方Cyber 205は大規模の実メモリをもち、仮想記憶方式のメモリー-メモリー型のベクトル計算機である。パイプライン演算器へのデータは原則として主記憶装置よりストリーム-ユニットを通して送られ、演算結果と同様主記憶装置に送られる。CRAY-1と比較すると立ち上がり時間は長いが、演算速度は速い。

VP 100はCRAY-1に近いレジスタ-レジスタ型であって、ベクトルレジスタの容量は32KBと大きく、その構成は必ずしも固定的ではなく柔軟性がある。レジスタには通常最大512要素格納される。VP 100は汎用機M382と互換性がある。即ちデータ形式はM382と同一で、M382と同じ機械命令語を持つほか、ベクトル命令が追加されている。したがって、M382で作成されたプログラム(ロードモジュール)はそのまゝVP 100上で実行可能である。また、VP 100でスカラモードを指定してコンパイル、実行すればM382の如く使用できる。これはデバッグ上有用である。浮動小数点の内部形式はIBM互換即ち16進形式となっている。CRAY-1, Cyber 205では64ビットまたは32ビットが用いられているが、2進形式である。

基本演算について汎用機との違いの一つは除算で、ベクトル計算機の特徴があらわれて、乗算・除算の速度比が計算機により異なる。VP 100の場合除算速度は乗算速度の1/7であり、CRAY-1の約1/3より小さく、むしろCyber 205のそれに近い。最近の話題として、除算の計算複雑さが $O(\log n)$ であることが示されたという(4) 実用的アルゴリズムの開発に発展することを期待したい。

ベクトル計算機上でのプログラムの実行効率を支配する要素として、ベクトル長とベクトル化率がある。全計算のうち w_k をベクトル処理ユニットで、 $(1-w_k)$ をスカラ処理ユニットで処理し、これらが時間的に重なっていると仮定すると全計算をスカラ処理ユニットのみで処理したときと比較した計算速度の加速率 β は

$$\beta = \frac{\alpha}{w_k + (1-w_k)\alpha} \quad (1)$$

ただし、 α はベクトル処理ユニットの計算速度がスカラ処理ユニットのそれの α 倍とする。

VP 100で α は最高20倍、VP 200で α は最高40倍といわれる。図3に $w_k - \beta$ のグラフを示す。 w_k はいわゆるベクトル化率となる。

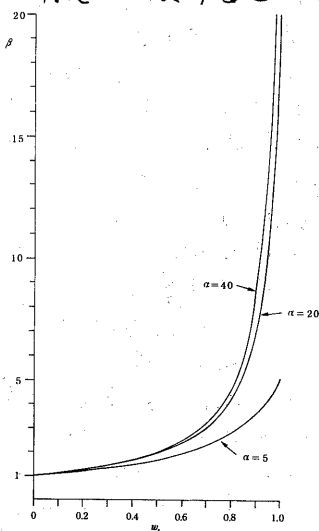


図3 ベクトル化率と加速率

3. システム・ソフトウェア

ベクトル計算機のシステム・ソフトウェアとしては、言語処理系、チューニング用ソフトウェア、科学技術計算ライブラリーが、効率的なプログラム開発に大きく影響する。我国のベクトル計算機が世界的に注目されているのは、ハードウェアの性能の他システム・ソフトウェアの良さにあると考えられる。実際 FORTRAN 自動ベクトル化技術は世界最高水準にあるといえる。この技術は現在も日進月歩の状態にあり、論じにくい。基本的な注意事項を述べるにとどめる。

基本的には DO ループがベクトル化の対象である。DO ループ内に複数の実行文が含まれている場合、その DO ループのベクトル実行はスカラ的にみたととき、文の実行順序の変化を招く。ある変数が式の左辺に出現したとき、その変数が定義されたといい、式の右辺に出現したとき引用されたという。ベクトル実行とスカラ実行の計算結果が変わらないためには、すべての変数について、ベクトル実行しても、定義・引用の順序関係がスカラ実行のときと同じである必要がある。

次の例を考えよう。

```
DO 20 I=1, N
  C(I)=A(I)+B(I)
  E(I)=C(I+1)-D(I)
20 CONTINUE
```

(a)

```
DO 10 I=1, N
  C(I)=A(I)+B(I)
10 CONTINUE
DO 20 I=1, N
  E(I)=C(I+1)-D(I)
20 CONTINUE
```

(b)

```
DO 10 I=1, N
  E(I)=C(I+1)-D(I)
10 CONTINUE
DO 10 I=1, N
  C(I)=A(I)+B(I)
20 CONTINUE
```

(c)

プログラム (a) をプログラム (b) が与えられたかの如くベクトル実行しようとしたとする。プログラム (a) とプログラム (b) とでは C(I) の引用・定義の順序関係が異なるので、(a) と (b) とでは実行結果が異なる。プログラム (a) のループ内の実行順を入れ換えて、ループ分割したのがプログラム (c) である。(a) と (c) とでは変数の定義・引用関係が同じで、プログラムの実行結果は同じである。プログラム (a) はそのままではベクトル化出来ないが、(c) の様な変形を許せばベクトル化可能となる。このような変形は自動ベクトル化コンパイラが行う。FORTRAN 77/VP コンパイラは ADVANCED モードのとき、DO ループ内の各文に対するベクトル化命令の順序をプログラムの意味を保つ範囲内で適当に変更し、ベクトル化率を高める。NOADVANCED モードではベクトル化命令の順序をソースにおける文の順序に係つことを条件にベクトル化する。したがって NOADVANCED モードでは (a) はスカラ実行される。このモードの選択機能は、精度に敏感なプログラムの実行やデバッグに有用である。

著名な線形計算パッケージのベクトル化の際に筆者の経験した一例を述べる。あるルーチンを NOADVANCED モードでベクトル化すると結果が正しく、ADVANCED モードでは結果が誤った。そのルーチンは整数演算が中心で誤差は無関係であった。FORTRAN の初歩的制限事項として、“異なる仮引数に同一の実引数が対応する場合、サブルーチン内では仮引数の値を定義してはならない” という項目がある。そのルーチンでは二つの仮引数配列に一つの实引数配列を対応させ呼びが定義があった。コンパイラの検出不可能な文法違反の例である。NOADVANCED モードのベクトル化の範囲では偶然結果はスカラ実行のそれと同一であったが、ADVANCED

モードでは、実引数の定義引用順序に変化が生じ、結果が変わったのであつた。このようにベクトル化によりそれ迄潜在的に存在していた、文法違反、精度に過度に敏感なアルゴリズム上の問題点が顕在化することがある。なおベクトル処理ユニットでは、一般に中間結果も倍精度で計算されるので、スカラ実行より精度的に条件がよい場合が多い。式の評価や総和の計算についても同様である。

4. FORTRANによる性能調査

ベクトル計算機は個性の強い計算機であり、そのすべての面をうまく表わす簡単な指標を与えることは容易ではない。現実にはバモア・ループ等各種のループによる性能評価が試みられている。基本的なループによる評価のほか、サンプルプログラムによる評価も試みられ、結果も公表されている⁵⁻⁷⁾。一方 Hockney 等⁵⁾はベクトル計算機の総合的能力、性格をできるだけ簡潔にとらえるという目的で、指標 γ_{∞} , $n_{1/2}$, π_{∞} を提案している。ここでは基本的な演算の幾つかについて、FORTRAN プログラムにより、計算時間の実測を行うとともに、 γ_{∞} , $n_{1/2}$ 等の評価も試みる。

被計測ループは一般に次の形とする。

```
DO 2 I = N1, N2, N3
  S
2 CONTINUE
```

S: FORTRAN 実行文

DO ループの実行時間が短く、精度よく計測するのが困難な場合が多い。そのとき、二重ループを形成し、平均値として、一回当りの実行時間を求める。その形は次のようにした。

```
CALL CLOCK(T1, 2, 2)
DO 4 J = 1, M
  CALL DUMMY
  DO 2 I = N1, N2, N3
    S
  2 CONTINUE
4 CONTINUE
CALL CLOCK(T2, 2, 2)
DO 6 J = 1, M
  CALL DUMMY
6 CONTINUE
CALL CLOCK(T3, 2, 2)
```

T1 (8バイト実数)にCPU時間が
マイクロ秒単位でセットされる

} 被計測ループ

CLOCK ルーチンを1回呼び出すときに必要な CPU 時間を T_0 とする。
 $(T_2 - T_1) - T_0$ は DO 4 の二重ループの全実行に使われた CPU 時間となる。
 一方 $(T_3 - T_2) - T_0$ は外側ループの実行、DUMMY ルーチンのコールに必要な CPU 時間を表わす。 $\{(T_2 - T_1) - (T_3 - T_2)\} / M$ が被計測ループを1回実行するのに必要な平均 CPU 時間となる。被計測文中の配列変数は COMMON 領域にとつた。このことと DUMMY ルーチンの CALL 文の挿入により、二重 DO ループに対

し、計時に不都合なコンパイラ最適化機能が動作するのを防いでいる。

実例について計測結果を示すまえに、計算時間に関するモデル式を示す。若干の記法の違いは別として、ほぼHockney⁵⁾に従う。

ベクトルレジスタの要素長を n_r とし、被演算ベクトルのベクトル長 n が、 n_r より大きいとき、 n_r 毎に区切って、ベクトル命令を実行するとする。CPU時間 t をモデル的に次式で表わす。

$$t = t_0 + \left[\frac{n}{n_r} \right] t_s + t_{eo} n \quad (2)$$

ただし

t_{eo} は実線の傾斜で、1要素当りの計算時間

t_0 はループ準備時間、

t_s は長さ n のベクトルを n_r 要素毎に区切ってベクトル命令を実行するとし、このために必要なオーバーヘッド時間である。

式(2)のグラフは図4の実線になる。

Hockneyはこのグラフを点線で近似し、その式を

$$t = \gamma_{\infty}^{-1} (n + n_{\frac{1}{2}}) \quad (3)$$

で表わし、 γ_{∞} 、 $n_{\frac{1}{2}}$ を指標とすることを提案している。

$Q = (\frac{1}{2} n_r, t_s + \frac{1}{2} t_{eo} n_r)$ 、 $R = (\frac{3}{2} n_r, t_0 + t_s + \frac{3}{2} t_{eo} n_r)$ であるから、

$$\gamma_{\infty} = \frac{n_r}{t_s + t_{eo} n_r} \quad (4)$$

$$n_{\frac{1}{2}} = \frac{n_r (t_0 - \frac{1}{2} t_s)}{t_s + t_{eo} n_r} \quad (5)$$

単位時間に計算される要素数は式(3)より

$$\frac{n}{t} = \gamma_{\infty} \left(1 + \frac{n_{\frac{1}{2}}}{n} \right)^{-1} \rightarrow \gamma_{\infty} \quad (n \rightarrow \infty)$$

各要素の計算に m 個の浮動小数点演算が必要であるとすると、MFLOPS値は $m n/t$ となる。 γ_{∞} は漸近的な速度値を表わし、限界値の $\frac{1}{2}$ を達成するベクトル長が $n = n_{\frac{1}{2}}$ であることがわかる。Hockneyはアーキテクチャの特有性能を表わす値として、

$$\pi_{\infty} = \gamma_{\infty} / n_{\frac{1}{2}} \quad (\text{MFLOPS}) \quad (6)$$

の使用を提案している。

(4)-(6)より

$$\pi_{\infty} = \frac{1}{t_0 - \frac{1}{2} t_s} \quad (7)$$

となる。 $n_{\frac{1}{2}}$ 、 π_{∞} に影響する t_0 について考える。FORTRANで前述の手法で計測すると、そのオーバーヘッドは t_0 に含まれる。たとえばDOループの制御変数の初期値、終値を定数で与えるか変数で与えるかにより影響を受ける。ベクトル長が n_r より小さいことが、コンパイラに判明すれば、一回のベクトル命令の実行で完了するわけで、オブジェクトコードは単純化できるからである。

図5-図10に幾つかの演算に関する実測データを示す。

実測に使用したコンパイラはFORTRAN 77/VP V10L10である。

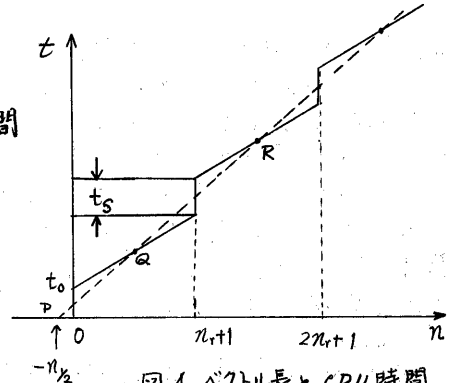


図4 ベクトル長とCPU時間

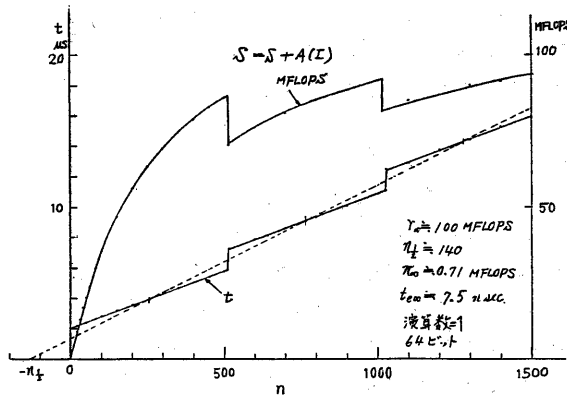


図5 $S = S + A(I)$ の性能

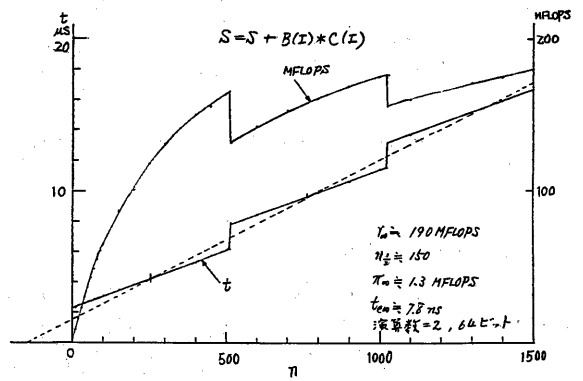


図6 $S = S + B(I) * C(I)$ の性能

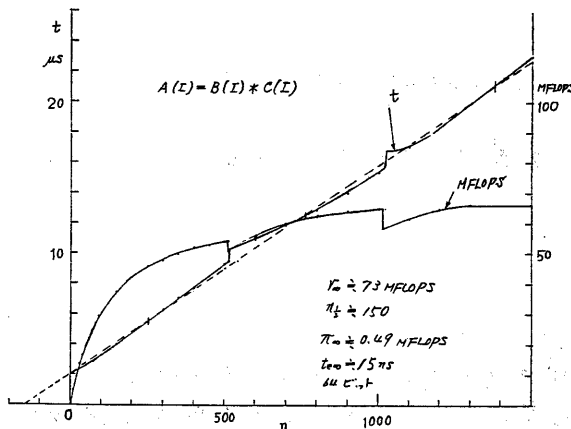


図7 $A(I) = B(I) * C(I)$ の性能

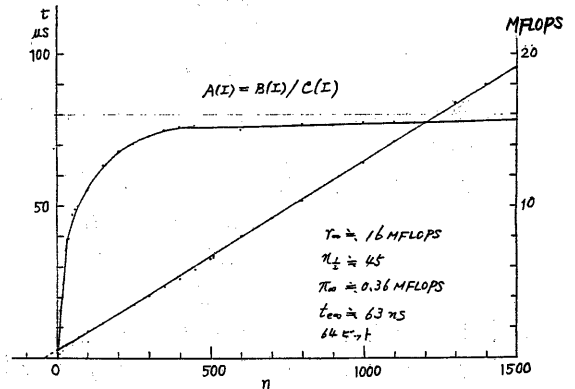


図8 $A(I) = B(I) / C(I)$ の性能

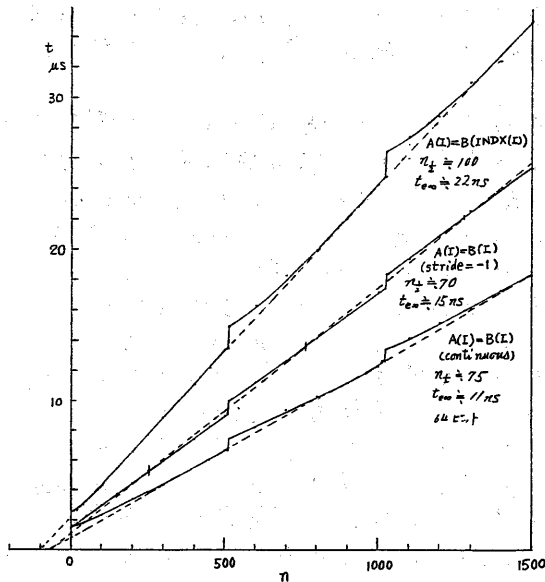


図9 $A(I) = B(I)$ 型の性能

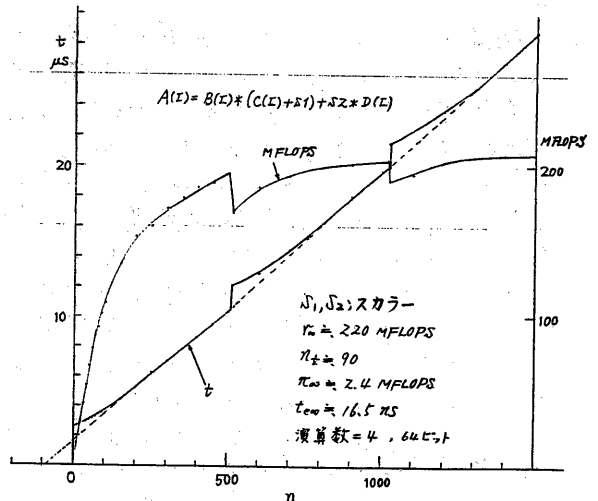


図10 $A(I) = B(I) * (C(I) + S1) + S2 * D(I)$ の性能

実測結果について、 $\pi-t$ のグラフを見ると、モデルの図4によくあうものと若干ずれるものがある。よくあうのは、 $S=S+A(I)$, $S=S+B(I)*C(I)$, $A(I)=B(I)$ [stride=-1] である。一方 $A(I)=B(I)$, $A(I)=B(INDX(I))$, $A(I)=B(I)*C(I)$, $A(I)=B(I)*(C(I)+S1)+S2*D(I)$ 等は $\pi=\pi_r$ 近辺の形が図4と異なる。

図5について説明する。 $S=S+A(I)$ のループの計算は

VLD A (Vector Load Long) (1)
VSMD (Vector Sum Long) (2)

の2命令で計算される。ロードパイプと加算パイプは独立に動作できる。したがってロードパイプからデータが出始めるとすぐ加算パイプで計算が始められ、(1) (2)は並行して動作できる。1チャーム(chime)で計算ができる。VP 100 のパイプラインのピッチは 7.5 ns で、実線の傾きは $t_{eo} = 7.5 \text{ nsec}$ でピッチに一致しており、予想通りの結果が得られていることがわかる。

$S=S+B(I)*C(I)$ も

VLD B
VLD C
VMD (Vector Multiply Long)
VSMD

で計算される。ロードパイプ2本、乗算、加算パイプが並行して動作し、1チャームとなる。

$A(I)=B(I)*C(I)$ については

VLD B
VLD C
VMD ; VSTD A

となる。すなわち2チャームとなり、実線の傾き $t_{eo} = 15 \text{ ns}$ はこれを示している。図9について説明する。図9より記憶装置とレジスタ間のデータ転送は連続アクセス、コンスタントストライド、インデックスアクセスの順に効率低下する。インデックスアクセスは連続アクセスの約半分程度となる。Cyber 205では連続アクセス以外は効率が非常に低下するが、VP 100の場合は図9に示した程度である。

Hockney のモデルの π_{\pm} 等の指標について考える。FORTRAN による計測では t_0 がかなり大きく、そのことは考慮しなければならない。ここでは概略値として、上記の結果を用いると、 $\pi_{\pm} \sim 45 \sim 150$; $\pi_{\infty} = 0.36 \sim 2.4 \text{ MFLOPS}$ となる。Hockney によれば、CRAY-1では $\pi_{\pm} \sim 10 \sim 20$; $\pi_{\infty} \sim 4 \sim 16 \text{ MFLOPS}$ であり、Cyber 205では、 $\pi_{\pm} \sim 100$, $\pi_{\infty} \sim 1 \sim 2 \text{ MFLOPS}$ の結果が示されている。CRAY-1について $\pi_{\pm} \sim 10 \sim 20$ と小さく、ほとんど直列計算機として動作しているとしている。VP 100 はアーキテクチャ的には CRAY-1 に大変近いのであるが、 π_{\pm} , π_{∞} の値からは Cyber 205 に近い。上記はループ中の文が一つの場合について評価したが、複数の文等を持つ場合も、典型例の選択がむずかしいが、考慮する必要がある。

次に考慮すべきパラメータとして、スカラ実行とベクトル実行が均合いベクトル長 n_b (vector breakeven length) または交叉ベクトル長がある。ループ制御のオーバーヘッドがスカラ実行のときとベクトル実行で異なるので厳密には容易でない。文の種類によるが VP 100 のとき、 $n_b \sim 3.5 \sim 10$ である。Hockney のテ

-7では, CRAY-1で $n_6 \sim 15 \sim 3$, Cyber 205で $n_6 \sim 11$ となっている。
VP100はCRAY-1とCyber 205との中間位にあるといえる。

5. 線形計算パッケージによる性能評価

著者はCRAY-1やCyber 205を直接使用する機会がないので, van der Vorst, van Katsの文献のデータと比較する。^{8,9)}文献よりCRAY-1, Cyber 205のテスト条件を示す。

CRAY-1A: Serial-9, at ECMWF, Shinfield Park, Reading, UK.

16 Bank Memory

CRAY Operating System COS 1.12 (11/10/83)

FORTRAN Compiler CFT 1.11 (07/30/83) options ON=CELMP@RSTV

Cyber 205: 2-pipe version, at Control Data France, Paris, France

8 Bank Memory

Operating System VMS VZOL575H (133010)

FORTRAN Compiler FORTRAN 2.1 CYCLE OTS21N (12/01/83)

options O=BOU T

紙数の関係もあり, 2,3の例をあげるにとめる。

内積 (MFLOPS 値)

N	CRAY-1		Cyber 205 (2-pipe)		VP 100
	FORTRAN	SDOT	FORTRAN	Q8SDOT	FORTRAN
50	10	17	17	17	32
1000	27	63	80	80	165
5000	29	72	95	95	184

行列積 (MFLOPS 値)

N	CRAY-1		Cyber 205 (2-pipe)		VP 100	
	FORTRAN(内積)	MXMA	FORTRAN(内積);(ROWWISE B)	(unrolling) SMXPY4	FORTRAN(内積)	SSL/VP MGM1
10	5.4	29	5.8	5.5	8	8
100	38	126	39	53	76	198
200	37	121	56	85	119	239
300	57	138	66	106	145	253

帯行列・ベクトル積 (MFLOPS 値) $b = Ax$, $N = 2000$

$NZ = N - 2$

DO 10 I=3, NZ

$B(I) = A(I-2, 3) * X(I-2) + A(I-1, 2) * X(I-1) + A(I, 1) * X(I) + A(I, 2) * X(I+1) + A(I, 3) * X(I+2)$

10 CONTINUE

CRAY-1: 46 MFLOPS, 2-pipe Cyber 205: 88 MFLOPS, VP 100: 130 MFLOPS

次に大規模対称行列固有値問題に対するLanczos法ルーチンの結果について述べる。プログラムはvan der Vorst, van Katsによる。行列の次数は $N = 1024$ である。Lanczos法の部分のルーチンGENLANについて会話型バクトライザで調べるとベクトル化率は99.5%である。f. EKSCANルーチンについては73.8%であった。

Lanzos法ルーチン (N=1024)				CPU時間(秒)			
GENLAN				EVJCAN			
CRAY-1	Cyber 205	VP100	M382	CRAY-1	Cyber205	VP100	M382
0.159	0.089	0.060	0.864	1.805	1.922	1.036	1.060

偏微分方程式を差分近似して得られる連立一次方程式の繰り返し型解法として前処理付の ICCG 法, PCR法等が重要視されている。ここでは van der Vorst による ICCG(1,1) のルーチンについての結果の一例を示す。プログラムの版の違い、収束回数の違いがあるので、VP100, M382の結果のみを示す。全話型ベクトライザの解析では ICCG11 のベクトル化率は 99.8% であった。一次方程式の次数は 4549, 繰り返し回数 104 回。ICCG11 ルーチンの CPU 時間は VP100/M382 の 881/1.806 であった。

6. おわりに

京都大学大型計算機センターの VP100 について述べ、幾つかの線形計算の問題について CRAY-1, Cyber 205 の結果と比較した。入手できるデータの関係上、CRAY-1 と比較したが、CRAY-1 XMP との比較も必要であろう。VP100 はよく CRAY-1 と比較されるが、Hockney の $\pi_{1/2}$, π_{α} のパラメータからみると VP100 は性質としては CRAY-1 と Cyber 205 の間にあると考えるべきであろう。Hockney は CRAY-1 のパラメータよりほぼ直列型計算機とみなしているようである。VP100 を最大限に有効に利用するためには、並列度に注意しなければならない。幸い、FORTRAN コンパイラの性能は CRAY-1 や Cyber 205 より格段によりと考えられるが、並列度を高めるべく、プログラミングだけでなく数値解析アルゴリズムにも注意することが重要であろう。

文献

1. 特集: スーパーコンピュータの応用, 情報処理, Vol. 22, No. 13, 1981.
2. 特集: スーパーコンピュータ, 日経エレクトロニクス, 1983年4月11日号, pp. 105-184.
3. P. D. Lax (Chairman): report of the panel on Large Scale Computing in Science and Engineering, NSF, December, 1982.
4. 矢島脩三: VLSI アルゴリズム, 岩波講座マイクロエレクトロニクス, 月報3, 1984. 12.
5. R. W. Hockney, G. Jesshope: Parallel computers, Adam Hilger Ltd, Bristol, 1981. (奥川峻史, 黒住祥祐共訳: 並列計算機, 共立出版, 1984年)
6. 島崎真昭: ベクトル計算機入門(2), 京都大学大型計算機センター広報, Vol. 17, No. 1, pp. 9-18, 1984.
7. 春野審世: 非公式レポート
8. H. A. van der Vorst, J. M. van Kats: Comparative Performance Tests on the CRAY-1 and the Cyber 205, Academisch Computer Centrum Utrecht, TR-15, 1983.
9. H. A. van der Vorst, J. M. van Kats: The Performance of some Linear Algebra Algorithms in FORTRAN on CRAY-1 and Cyber 205 Supercomputers, Academisch Computer Centrum Utrecht, TR-17, 1984.