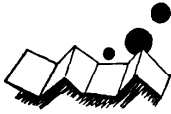


## 解説

# スーパーコンピュータと行列計算†



唐木幸比古††

### 1. はじめに

理学や工学の分野では、実験と理論によって自然現象の解明や加工物の設計を行うのが主流であったが、最近ではスーパーコンピュータの登場により、実験の一部または相当部分を仮想実験（数値シミュレーション）によって補うケースが増えている。計算物理学、計算化学、計算生物学など（まとめて計算理学）、あるいは計算力学、計算流体力学など（まとめて計算工学）の学問分野が拡大し、スーパーコンピュータをめぐるさまざまな研究会や国際会議などが開催されるようになった。

特に、これまで CRAY, CDC (米国) の独占市場であったこの分野に、1982 年より日本勢が参入してからは、世界の科学技術計算分野に活性化現象が生じた。世界最高速スーパーコンピュータの水準は、

(64 ビット演算最高性能)

ILLIAC IV	80 MFLOPS (1973)
CRAY-1	160 MFLOPS (1976)
Cyber 205	400 MFLOPS (1981)
S-810/20	750 MFLOPS (1983)
VP-400	1100 MFLOPS (1985)
SX-2	1300 MFLOPS (1985)

(MFLOPS=Million Floating-point Operations  
メガフロップス  
Per Second)

へと向上し、多重プロセッサとしては

CRAY-2 (4 多重) 1700 MFLOPS (1986)

まで到達している。CRAY を中心に世界の主要な科学技術研究所、大学、民間企業などに 200 台以上普及し、日本国内だけでも、VP, S-810, SX, CRAY X-MP など合計 60 台以上普及している。

スーパーコンピュータは、現在の第 3 世代から次の新世代へと移行しつつあり、現在発表されているスー

パコンピュータの到達性能 (64 ビット演算) は

単体プロセッサ 3 GFLOPS (S-820/80)

多重プロセッサ 5 GFLOPS (ETA<sup>10</sup> (8 多重))

である。(GFLOPS=1000 MFLOPS)

ギガフロップス

本稿では、現状のスーパーコンピュータについて概観し<sup>1)-3)</sup>、現在の主流となっている演算パイプライン方式とその並列化（または多重化）の特徴を述べ、大規模科学技術計算の中心課題である行列計算への適用性と高速化技法<sup>4)</sup>について述べる。

本稿の構成は、スーパーコンピュータの現状、並列パイプライン方式の特徴、高速化技法の例、行列計算への適用及びまとめである。

### 2. スーパーコンピュータの現状

1983 年 10 月に初の国産スーパーコンピュータが稼動を開始してから 4 年が経過した。この間に我が国に設置されたスーパーコンピュータ（ピーク性能 100 MFLOPS 以上とする）の台数は 69 を数える。これは開発時期の目標設置台数 30 を大幅に超えるものであり、加えて、それまで米国の独占状態にあった世界のスーパーコンピュータ市場を大変に活気のあるものにした。その結果、米国クレイ社、コントロール・データ社のスーパーコンピュータを始めとして、日立製作所、富士通および日本電気の日本 3 社のスーパーコンピュータが広く普及し、日本の大学ネットワークなどを通じて利用技術も向上している。汎用機内蔵型（または付加型）のアレイプロセッサや、100 MFLOPS 以下の水準ながら通常の汎用大型機（スカラプロセッサ）の 10 倍近い性能を有するセミスーパーコンピュータ（ミニスーパーコンピュータ）を含めると、科学技術計算用の高速コンピュータの需要はここ数年急激に伸びている。最近は特に、半導体をめぐる貿易上の摩擦と並んで、スーパーコンピュータをめぐる技術的および通商貿易面の競争が激しさを増しており、性能比較、信頼性、使い勝手、ソフトウェア蓄積、設置導入手続きなどさまざまな側面からの比較論議が活発になって

† Supercomputer and Matrix Calculation by Yukihiko KARAKI (School of Commerce, Senshu University).

†† 専修大学商学部

いる。

スーパーコンピュータになぜこれだけの注目と関心が寄せられるのか。それは、コンピュータの評価が演算能力の高さと密接に関連しているからであり、米国と同等もしくはそれを上回る演算能力をもつ国産スーパーコンピュータの登場によって、スーパーコンピュータ市場のみならず汎用計算機市場にも少なからず「速さの威力」の影響が生じ、同時に、コンピュータを利用する理学・工学の諸方面の技術開発や科学研究に対する影響自体をも無視できなくなっているからにほかならない。

ここで、スーパーコンピュータの現状について概観し、現代の主流にある演算パイプライン方式とその並列化（または多重化）の特徴を述べる。

#### スーパーコンピュータの定義

同時代のコンピュータの中で、抜きん出て超高速の演算能力をもつものはスーパーコンピュータ (super-computer) と呼ばれる。

演算能力の基準は時代の推移とともに向上する。

現在は、100 MFLOPS 以上のピーク性能（最高演算能力）をもつものをスーパーコンピュータと呼んでいるが、この数値は少し古くなった感がある。

過去のスーパーコンピュータを幾つかの世代に分類することがある。分類の方法は、年代で大別し高速演算機能の範囲で特徴づける。表-1 に、スーパーコンピュータの世代分類の例を示す。現在のスーパーコンピュータは第三世代に属するが、新世代に属するとすべきものも登場し始めている (ETA<sup>10</sup> など<sup>5), 6)</sup>。表-2 に現代のスーパーコンピュータの一覧を示す。

表-1 スーパーコンピュータの世代分類

世代	特徴	例
第0世代	スカラプロセッサ	CDC 6600 (1964)
	オーバーラップ処理	CDC 7600 (1968)
	パイプライン処理	IBM 360/195 (1969)
第一世代	アレイプロセッサ	TI ASC (1972)
	拡張 FORTRAN	ILLIAC IV (1973)
	長いベクトル	CDC STAR-100 (1973)
第二世代	アレイプロセッサ	CRAY-1 (1976)
	FORTRAN	CDC Cyber 205 (1981)
第三世代	アレイプロセッサ	HITAC S-810 (1983)
	FORTRAN 77	CRAY X-MP (1983)
	IF 文	FACOM VP (1984)
	リストベクトル	NEC SX (1985)
	短いベクトル	CRAY-2 (1985)

#### ハードウェアの特徴

現在のスーパーコンピュータは、演算パイプライン方式のベクトルプロセッサ (vector processor) と逐次演算方式のスカラプロセッサ (scalar processor) を基礎として構成される。

スーパーコンピュータには下位から上位までの性能水準があり、その組立て方には並列パイプライン方式と多重プロセッサ方式がある。たとえば、ピーク性能で 300 MFLOPS のベクトルプロセッサを 1~4 並列（または 1~4 多重）に複数化することによっておよそ 1200 MFLOPS の高速性能モデルを得るなどである。国産機は現在、並列パイプライン方式によって上位性能を達成しており、米国機の多重プロセッサ方式と対比を成している。並列パイプライン方式は単体プロセッサとしての性能水準が高く、多重プロセッサ方式は、通常、複数ジョブ実行時に全体のスループットを上昇させ効果を発揮するが個々のジョブの処理性能自体は単体性能水準で動作するケースが多く、むしろ他のジョブを止めて単一ジョブに限れば多重度が生きる。ハードウェアの実効性能は、単体プロセッサの能力によって規定されるから、並列パイプライン方式は多重プロセッサ方式より正統的（ハードウェア高水準）である。

スーパーコンピュータの演算能力を特徴づけるもう一つのハードウェア要素は高速記憶装置である。主記憶を中心に、通常、いくつかの階層で構成される。現在は、高速側に中間記憶（ベクトルレジスタ、バッファ）、低速側に拡張記憶または共有記憶などの階層を置いて、主記憶を含め 2 階層または 3 階層の記憶階層をもつものが多い。

現状では、論理回路及び中間記憶をシリコン・バイポーラ素子で構成し、主記憶を MOS スタティック RAM、拡張（または共有）記憶を MOS ダイナミック RAM などのシリコン記憶素子で構成するものが多い。ETA<sup>10</sup> では、論理回路に CMOS シリコン素子を用いて液体窒素冷却 (77°K) とした。

シリコン記憶素子は、64 K bit SRAM (アクセス速度 40 nsec 程度) の主記憶と 256 K bit DRAM (アクセス速度 120 nsec 程度) の拡張（または共有）記憶といった水準にあり、集積度の向上で現在の記憶容量（主記憶 256 MB、拡張記憶 2~4 GB）に近い将来には少なくとも 2~4 倍は向上するであろう (S-820/80 では主記憶 512 MB、拡張記憶 12 GB)。

論理素子については、現状の 200 psec 級の回路

表-2 スーパーコンピュータの一覧

最高性能 100 MFLOPS 以上

機種名	最高性能 MFLOPS	主記憶 MB	拡張記憶 GB	特徴
<u>Cray</u>				
CRAY-1 (S, M)	160	32	1	単体, ベクトルレジスタ
X-MP/1	235	64	1	単体, ベクトルレジスタ
X-MP/2	470	128	2	2多重, ベクトルレジスタ
X-MP/4	940	128	4	4多重, ベクトルレジスタ
CRAY-2	976	1024		2多重, ベクトルレジスタ
CRAY-2	1950	2048		4多重, ベクトルレジスタ
<u>Control Data</u>				
Cyber 205	200	32		単体 (2パイプ), ストア
Cyber 205	400	64		単体 (4パイプ), ストア
<u>ETA Systems</u>				
ETA-10	4571	256	2*	8多重, ストア & バッファ
<u>FPS</u>				
M64/MAX	341	58		15多重, ソフトパイプ
T/20	192	16		2モジュール (16ノード)
T/100	768	64		4モジュール
T/200	1536	128		8モジュール
T/40000	196 GFLOPS			1024*T/20, ハイパキューブ
<u>IBM</u>				
IBM 3090/VF (180 E)	116	64	2	単体, ベクトルレジスタ
IBM 3090/VF (600 E)	698	256	2	6多重, ベクトルレジスタ
	MFLOPS	MB	GB	
<u>日立製作所</u>				
S-810/5	214	128	3	単体, ベクトルレジスタ
S-810/10	428	128	3	単体, ベクトルレジスタ
S-810/20	857	256	3	単体, ベクトルレジスタ
S-820/60	1500	256	6	単体, ベクトルレジスタ
S-820/80	3000	512	12	単体, ベクトルレジスタ
<u>富士通</u>				
VP-50	142	128		単体, ベクトルレジスタ
VP-100	285	128		単体, ベクトルレジスタ
VP-200	571	256		単体, ベクトルレジスタ
VP-400	1142	256		単体, ベクトルレジスタ
VP-400 改良型	1700	1024		単体, ベクトルレジスタ
<u>日本電気</u>				
SX-1 E	285	128	2	単体, ベクトルレジスタ
SX-1	571	256	2	単体, ベクトルレジスタ
SX-2	1333	256	2	単体, ベクトルレジスタ

最高性能は64ビット演算での数値

\* 共有記憶 (Shared Memory)

備考 表中, ETA-10 は現在4多重のモデルが稼動しており, 年内には最大構成の8多重システムが稼動する予定である。また, S-820/60 及び S-820/80 は今年度第4四半期出荷の予定である。VP-400 改良型は VP シリーズ E モデル (最高性能を 1.5 倍性能向上) の最上位機種であり, 今年12月からの出荷予定である。FPS T シリーズの稼動最大構成は現在8モジュール (T/200) までである。

速度 (シリコン) を, 化合物半導体 (ガリウム砒素, HEMT など), 超伝導素子 (量子磁束素子を含む) などどこまで向上させられるかの開発競争が展開されている。ETA<sup>10</sup> では, 論理素子の性能向上は (たとえガリウム砒素であっても) 高高2倍程度であると見

越して, 多重プロセッサ方式 (最大多重度8) を採用した。ETA<sup>10</sup> の単体プロセッサ性能は現在 571 MFL-OPS (マシンサイクル 7 nsec) (64 ビット演算) であり, NEC SX-2 の 1.3 GFLOPS (マシンサイクル 6 nsec) (64 ビット演算) の約 1/2 の性能である。多重

度8の場合に最大 4.5 GFLOPS (32 ビットではその2倍) のピーク性能を達成する。S-820 は今年度第四半期からの出荷が予定されているが、単体プロセッサとしては最高速の 3 GFLOPS (マシンサイクル 4 nsec) となる。

ソフトウェアの概要

数値シミュレーション用プログラム言語の主流にあるのは FORTRAN である。FORTRAN は機械語やアセンブラより遙かに使いやすく記述の誤りも少ない高級言語として登場した最初のものであり、特に浮動小数点演算のハードウェア化と一体になって普及し、今日、科学技術計算の分野で最も多用され、かつ蓄積プログラムも膨大なものとなっている。したがって言語コンパイラの最適化技術の水準が高く、同時に言語仕様の標準化が利用者を主体として行われている。

スーパーコンピュータ用の主要言語が現在 FORTRAN 77 になっているのは、上記の理由による。

浮動小数点演算のハードウェア化の時代のプログラ

ミングの課題は算法記述の容易化にあった。逐次処理に加えて浮動小数点演算などの並列処理 (演算パイプライン方式や並列処理方式) をハードウェア化した現代のプログラミングの課題は、算法記述の継続化と最適化である。その課題に沿って国産スーパーコンピュータを中心に、FORTRAN 77 自動ベクトル化コンパイラとベクトル化支援ツールの開発が行われ、一定の能力水準を達成した。表-3 は、逐次処理ハードウェアと並列処理 (演算パイプライン方式) ハードウェア上での FORTRAN 77 コンパイラの最適化水準を示したものである<sup>7)</sup>。加減乗算や内積・積などの配列演算を中心として高水準の演算性能が得られている。

ベクトル化支援ツールは、逐次処理時代の FORTRAN 支援ツールを受け継いで、演算パイプライン方式ハードウェアに対処しその機能を拡大したものである。従来の解析情報 (実行比率など) に加えて演算パイプライン方式に関連する情報 (最内側 DO ループ

表-3 FORTRAN 77 自動ベクトル化の効果 (L.L.N.L. MFLOPS KERNELS (14 LOOPS) の例)

機 種	S-810	K-382	SX-2	M-680H	M-680H IAP	VP-200	S-810 model 20	SX-2	SX-1	SX-1
言 語 ビット長	FORT 77 Scalar 64	FORT 77 Scalar 64	FORT 77 Scalar 64	FORT 77 Scalar 64	FORT 77 Vector 64	FORT 77 Vector 64	FORT 77 Vector 64	FORT 77 Vector 64	FORT 77 Vector 64	FORT 77 Scalar 64
LOOP	MFLOPS	MFLOPS	MFLOPS	MFLOPS	MFLOPS	MFLOPS	MFLOPS	MFLOPS	MFLOPS	MFLOPS
1	9.4	10.8	18.1	25.3	83.9	325.7	303.8	754.1	312.6	15.4
2	9.6	12.5	31.4	27.3	43.3	178.9	277.5	415.6	217.9	26.9
3	9.1	7.2	15.9	25.7	54.5	325.9	344.1	562.1	337.3	13.6
4	8.2	5.9	11.5	18.9	48.2	78.6	66.1	124.1	89.8	9.9
5	7.4	10.0	15.9	16.5	14.9	8.3	23.6	14.6	12.5	13.6
6	7.4	8.9	16.7	17.8	14.6	8.6	26.1	15.6	13.3	14.3
7	10.0	15.6	32.2	32.8	48.9	332.3	241.3	804.0	448.9	27.4
8	10.0	14.0	46.2	24.5	27.5	97.3	114.4	153.6	116.1	39.6
9	9.4	13.8	32.0	28.0	67.3	157.6	234.5	719.1	423.1	27.7
10	7.4	7.7	18.5	15.8	19.8	48.1	47.4	130.1	104.6	16.2
11	9.8	4.6	8.2	22.6	11.0	3.8	16.8	24.3	18.7	7.1
12	7.7	4.7	8.3	16.0	53.5	110.9	113.2	254.5	114.7	7.1
13	1.8	2.3	9.3	3.7	3.6	6.6	6.9	8.2	6.8	7.9
14	2.7	4.1	14.3	6.8	6.8	13.6	12.6	24.6	19.5	12.3
H. MEAN	6.01	6.56	15.26	14.34	16.19	17.51	32.09	37.77	30.41	13.09
AVERAGE	7.86	8.73	19.89	20.12	35.56	121.17	130.59	286.04	159.70	17.07

備考

H. MEAN (調和平均) = 14 / SUM (1/MFLOPS 値)

AVERAGE (平均) = SUM (MFLOPS 値) / 14

LOOP 内容

- |              |                 |          |             |
|--------------|-----------------|----------|-------------|
| 1. 流体        | 5. 三角化消去 (下三角)  | 9. 整数予測  | 13. 2次元粒子推進 |
| 2. MLR 内積    | 6. 三角化消去 (上三角)  | 10. 差分予測 | 14. 1次元粒子推進 |
| 3. 内積        | 7. 状態方程式        | 11. 逐次総和 |             |
| 4. 帯型連立一次方程式 | 8. 偏微分方程式 (ADI) | 12. 差分   |             |

の実行比率、ベクトル化率、推定加速率、並列パイプライン稼働率などが表示される。FORTRAN プログラムにおける DO ループ (または FORTRAN 文) のベクトル化の可否とその理由などの情報は、国産機ではコンパイルの時点でソースリストに重ねて表示される。

米国製スーパーコンピュータでは、当初、アセンブラ・プログラミングを主体としていたが、利用者の拡大とともに FORTRAN 77 自動ベクトル化機能を強化し、ベクトル化プリプロセッサなども開発している。また、FORTRAN 以外の言語コンパイラでは、Pascal, Ada, C などを支援している例もある<sup>9)</sup>。

基本ソフトウェア (OS) には、おのおののハードウェアに対する専用 OS を採用しているが、最近のスーパーコンピュータ (CRAY-2 や ETA<sup>10)</sup>) ではこのほかに UNIX を支援する例もみられる。

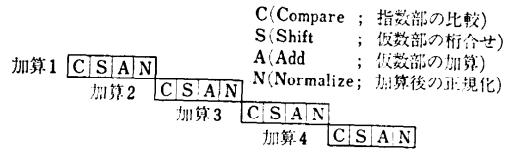
数学ソフトウェアでは、処理系定義の数学関係のベクトル化は当然として、スーパーコンピュータの応用分野の中心課題は各種の行列形に対応しうる行列計算のベクトル化である。特に、大規模な行列 (疎、帯または密行列) を係数とする連立一次方程式 (および固有値問題) の安定で高速な解法の開発が肝要である。

### 3. 並列パイプライン方式の特徴

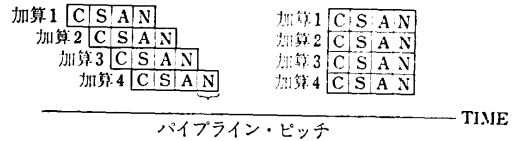
パイプライン処理には、主に、次の三つがある<sup>9)</sup>。

- (1) 命令パイプライン
- (2) 演算パイプライン
- (3) 転送パイプライン

命令パイプラインは、命令の読み込み処理をパイプライン化するものであり、複数個の逐次命令 (スカラ命令) を順次読み込む方式 (ソフトウェア・パイプライン方式) と、演算対象を拡大して一つの命令で複数個の演算対象 (ベクトル・データ) を扱う命令 (ベクトル命令) を含む命令列を読み込む方式 (ハードウェア・パイプライン方式) とがある。後者の場合には、ベクトル命令を解釈して複数個の部分作業に展開し、そのおのおのを命令パイプラインに乗せる手続きが必要になるので、一定の前処理準備時間 (オーバーヘッド) が必要。現代のスーパーコンピュータの多くは、ハードウェア・パイプライン方式を採用している。ベクトル命令のセットはスーパーコンピュータによって異なる。ベクトル命令によるパイプライン処理を行う演算装置のことをベクトル処理装置 (ベクトルプロセッサ, vector processor) という。



(a) 逐次処理 (スカラ処理)



(b) パイプライン処理 (ベクトル処理) (c) 並列処理 (パラレル処理)

図-1 処理方式の比較 (浮動小数点加算の例)

演算パイプラインは、演算の実行をパイプライン処理する。図-1 に、浮動小数点加算の場合の処理方式 (逐次処理、パイプライン処理及び並列処理) の違いを示す。パイプライン演算器には、浮動小数点演算器 (加算器, 乗算器, 除算器), 整数論理演算器 (加算器), マスク演算器などがある。パイプライン演算器を1組だけもつものを単一パイプラインと呼び、複数組もつものを並列パイプラインと呼ぶ。演算パイプラインの並列化によって、単一パイプラインの演算性能を並列度だけ倍化することができる。並列演算には同種演算だけを許す単純並列方式のものと、異種演算までも許す多重並列方式のものがある。多重並列の場合には、通常、チェイニング機能 (ベクトル演算の結果を主記憶に返さずに他のベクトル演算対象として用いて、パイプライン・ピッチ当たり複数個の演算を処理する) を基礎としてその並列化 (多重チェイニング) を行うか、ベクトル命令セットを多機能化してパイプライン処理を多重並列化する。S-810/20 および S-820/80 では、4つのベクトル内積命令と4つのベクトル加算命令が並列に多重チェイニング処理されて最大12演算並列、改良中の新型 VP-400 ではスカラとベクトルの混在する4つの内積と加算が並列にチェイニング処理されて最大12演算並列、NEC SX-2 では4つのベクトル加算処理と4つのベクトル乗算処理が4つの論理演算処理と4つのシフト演算処理と並列パイプライン動作して最大16並列 (浮動小数点演算8並列) となる。CRAY-1, X-MP および CRAY-2 では、スカラ\*ベクトル+ベクトルの乗算命令と加算命令をチェイニングする単一パイプライン方式をとり、並列パイプライン機能はもたない。ETA<sup>10)</sup> では、Cyber 205 を改良して、ベクトル命令の演算対象とし

て直前のベクトル命令の演算結果を用いる新命令の追加（チェイニング機能）によって、ベクトルレジスタなしで多重並列パイプラインの効果を実現した。ETA<sup>10</sup>の単体プロセッサはCyber 205の2並列パイプラインモデルを機能強化した最大4並列の多重並列プロセッサである。

転送パイプラインは、演算対象を主記憶から読み出したり、主記憶へ格納する処理をパイプライン化する。主記憶と演算器の動作速度は、通常、1桁以上異なるので、その速度格差を埋めるために主記憶を幾つかの部分空間（バンク）に分割し、対象データの記憶番地（アドレス）によって転送路（インターリーブ・ウェイ）を分配し、複数の転送路を用いて逐次的にデータの転送を処理する。転送パイプラインの並列動作（転送データ量）の幅は、並列化（1本の転送路でデータ幅を広げる）方式と多重化（複数の転送路を設定する）方式の組合せで決まる。

SX-2, S-810 および ETA<sup>10</sup> では、クロックごとに8個の64ビットデータを転送する。VP-400 および VP-200では、クロックごとに4個の64ビットデータを転送する。X-MP(単一プロセッサ)では、クロックごとに3個の64ビットデータを転送する。CRAY-1とCRAY-2(単体プロセッサ)では、クロックごとに1個の64ビットデータを転送する。

スーパーコンピュータのクロック（マシンサイクル時間）は、表-4を参照されたい。

並列パイプライン演算の概略の性能は次式のようになる。

$$M_v(l, p) = \frac{1000p}{\tau_0/l + \alpha\tau^*} \text{ (MFLOPS)} \quad (1)$$

ただし、 $l$ : ベクトル長、

表-4 マシンサイクルとベクトルレジスタ長

機種名	マシンサイクル ベクトルレジスタ長	
	NSEC	WORD
CRAY-1 (S, M)	12.5	64W
X-MP/1	8.5	64W
CRAY-2	4.1	64W
Cyber 205	20	
ETA-10	7	
FPS M64/MAX	182	2048W
T/20	125	
IBM 3090/VF (180 E)	17.2	128W
S-810	14	256W
S-820	4	512W
FACOM VP	7	32W*n
SX-1	7	128W
SX-2	6	256W

$\tau^*$ : 実効マシンサイクル時間 (nsec)

$\alpha$ : 演算クロック数 (ベクトル処理)

$\tau_0$ : 前処理準備時間 (nsec)

$p$ : 演算パイプライン並列数

CRAY や VP では、 $\tau_0$  が一定の周期（ベクトルレジスタ長）で変化する。

最大性能（ピーク性能）は次式で与えられる。

$$M_v(\infty, p_{max}) = \frac{1000p_{max}}{\tau^*} \quad (2)$$

ただし、 $p_{max}$ : 最大演算パイプライン並列数  
極限性能の半分の性能を与えるベクトル長（半性能長）は次式で与えられる。

$$l_{1/2} = \frac{\tau_0}{\tau^*} \quad (3)$$

逐次演算の概略の性能（スカラ処理性能）は次のようになる。

$$M_s(l) = \frac{1000}{\alpha_s \tau_s^*} \quad (4)$$

ただし、 $\tau_s^*$ : 実効スカラ・マシンサイクル時間 (nsec)

$\alpha_s$ : 演算クロック数 (スカラ処理)

逐次演算とパイプライン演算の速さが等しくなるときのベクトル長（交叉ベクトル長）は次のようになる。

$$l_c = \frac{\tau_0}{p\alpha_s \tau_s^* - \alpha\tau^*} \quad (5)$$

演算パイプライン方式のスーパーコンピュータは、ベクトル処理の前処理準備時間 ( $\tau_0 \neq 0$ ) によって特徴づけられる。すなわち、

$$\begin{cases} 1 \leq l \leq l_c & \text{では } M_v(l) \geq M_s(l, p) \\ l \geq l_c & \text{では } M_v(l) \leq M_s(l, p) \end{cases} \quad (6)$$

となる。

スーパーコンピュータの速さの例を表-5に掲げる。

浮動小数点ベクトル演算では、通常、内積、積和などが最も速く、次いで加算と乗算、次いで除算となる。IF文を含む演算やリストベクトル（間接添字付き配列演算）の速さはスーパーコンピュータによってバラつきがある。SXではIF文を含む演算が加減乗算並みに速く、リストベクトルではS-810, ETA<sup>10</sup>などが速い。多項演算で並列演算性能が発揮される。

SX, ETA<sup>10</sup>, CRAY などでは、逐次演算の一部もオーバーラップ処理する。特にSXではスカラマシンサイクル時間がベクトルマシンサイクル時間に等しいので、逐次演算の速さが他に抜き出ている。

ピーク性能では、多重プロセッサ方式によってCRAY-2やETA<sup>10</sup>が世界最高速を保持しているが、

表-5 スーパーコンピュータの速度例

S-810/20

演算	FORTRAN 文	加速率 (MFLOPS)			
		ベクトル長		10 <sup>4</sup>	10 <sup>5</sup>
		100	300	10 <sup>4</sup>	10 <sup>5</sup>
加算	A(I)=B(I)+C(I)	11	26	33	120.5
乗算	A(I)=B(I)*C(I)	16	25	42	117.0
除算	A(I)=B(I)/C(I)	18	34	48	62.1
総和	SUM=SUM+C(I)	8	18	27	120.5
内積	SUM=SUM+B(I)*C(I)	13	24	45	256.0
積和	A(I)=B(I)+C(I)*D(I)	19	37	53	227.6
多項演算	V=V+S*V*V+S*V+S*V+S*(×5行)	20	27	31	510.4
多項演算	SUM1=SUM1+(B(I)+C(I))*2(×4行)	25	38	86	712.3
IF文	IF( )THEN A(I)=A(I)+B(I)*C(I) ELSE A(I)=A(I)-B(I)*C(I)	10	14	20	62.1
リストベクトル	A(I)=B(I)+C(J(I)+K)	29	40	49	93.2
逐次演算	A(I)=A(I-1)+B(I)*C(I)	3	3	4	20.2

備考 加速率=ベクトル速度/スカラー速度

リバモア・ループ、LINPACK、基本演算などの性能調査では、SX-2 の高速性が他に抜きん出ており、S-820 がそれを受け継ぐことになる。

#### 4. 高速化技法の例

この章では、並列パイプライン方式のスーパーコンピュータにおける幾つかの高速化技法について紹介する。この方式のスーパーコンピュータは、すでに述べたように逐次処理速度（以後スカラー速度と呼ぶ）とパイプライン処理速度（以後ベクトル速度と呼ぶ）の2種類の速さをもち、利用者プログラムの中のベクトル化可能部分だけがベクトル速度で処理され、ベクトル化不能部分はスカラー速度で走る。ベクトル化の可否については言語コンパイラが判断し自動的にベクトル命令とスカラー命令の混在する実行プログラムに変換する（自動ベクトル化）。しかしながら、言語コンパイラの自動ベクトル化能力には一定の限界があり、演算対象（配列要素）の間の演算独立性について、ソースプログラムの静的解析だけでは見極めがつかないことがある。また、パイプライン演算器の性能は、前章の式(1)で示したようにベクトル長とパイプライン並列度に依存する。したがって、高速化の技法は主に次の三つになる。

技法A 配列要素間の演算独立性を高める

技法B ベクトル長を大きくする

技法C パイプライン並列度を上げる

このほかに、転送パイプラインにおけるデータ探索順（アクセス順）もベクトル速度に影響を与え、連続アクセスの場合が最も速い。ただし、実記憶方式ではメモリ衝突（バンクコンフリクト）を除けば性能の低

下は比較的小さい。そこで

技法D メモリ衝突を避ける

のも高速化の技法の一つになる。多重プロセッサ方式の場合にはこのほかに、タスク並列度を高めるという配慮も必要になるが、ここではそれは考えない。

以上のA～Dの各技法をソースプログラムに盛り込むためには、新開発の場合を除けば、ベクトル化支援ツールを用いて解析し、ベクトル化対策を講じる必要がある。解析においては、実行比率の高い配列演算に注目し、そのベクトル化および高速化を行うと作業効率が上がる。プログラムのベクトル化対策の作業効率を知る目安となるものは次の二つである。

i) ベクトル化率の向上

ii) 加速率の向上

ベクトル化率とは、実行プログラムの全実行ステップ数に対するベクトル化可能部分の実行ステップ数（スカラー命令で展開した場合）の比率をいう。

加速率とは、スカラー命令で展開された実行プログラムと、スカラーおよびベクトル命令に展開された実行プログラムにおける各CPU時間の比率（高速化の倍率）をいう。

ベクトル化率を向上させるには、ベクトル化可能な配列演算を増やす必要がある。つまり技法Aの対策が必要になる。

加速率を向上させるには、パイプライン演算器の性能を十分に引き出すように仕向けるべきである。すなわちB～Dの技法を適用する必要がある。

以下に、高速化技法の幾つかの例を述べる。

例1. 逐次計算を独立演算にする方法（技法A）

(イ) 1次元配列の場合

```

R(0)=A
DO 1 I=1, M*N          (原型)
1 R(I)=R(I-1)*P
  ↓
R(0, 0)=A
PM=1.0
DO 10 I=1, M
  PM=PM*P
10 R(I, 0)=R(I-1, 0)*PM
  DO 11 J=1, N
    DO 11 I=1, M          (ベクトル型)
11 R(I, J)=R(I, J-1)*PM

```

原型プログラムでは、一つ前に実行された演算の結果を用いて計算するのでパイプライン演算器が有効に働かない。M 個の演算をまとめて行うよう改訂した。

(ロ) 2次元(または多次元)配列の場合

```

DO 1 J=1, NY
  DO 1 I=1, NX          (原型)
1 U(I, J)=(U(I-1, J)+U(I+1, J)
  +U(I, J-1)+U(I, J+1))/4
  ↓
DO 10 L=1, NX+NY-1
  DO 10 K=KS(L), KE(L)
    I=II(K, L)          (ベクトル型)
    J=JJ(K, L)
10 U(I, J)=(U(I-1, J)+U(I+1, J)
  +U(I, J-1)+U(I, J+1))/4

```

原型プログラムでは、 $U(I-1, J) \rightarrow U(I, J)$  の部分が逐次計算になっているが、ベクトル型では、(I, J) 座標系を  $45^\circ$  回転した (K, L) 座標系において、 $L=I+J-1$  を保ちながら、各 L についてその境界 KS(L) と KE(L) の間で K を動かして独立演算にしている<sup>4)</sup> (座標回転法または超平面法ともいう)。

例2. ベクトル長を大きくする方法 (技法B)

(イ) 配列の次元の減少が簡単な場合

```

DO 1 J=1, N
  DO 1 I=1, M          (原型)
1 C(I, J)=A(I, J)+T*B(I, J)
  ↓
EQUIVALENCE (A(1, 1), AA(1)), (B(1, 1),
  BB(1)), (C(1, 1), CC(1))
DO 10 I=1, M*N        (高速型)
10 CC(I)=AA(I)+T*BB(I)

```

原型プログラムでは、最内側 DO ループのベクト

ル長が M であるが、高速型ではベクトル長が  $M*N$  になっている。M が小さくて N が大きい場合には、通常、原型の I と J を入れ換えるだけでもベクトル長は大きくなり高速化できる。なおこの例のような場合、最近の FORTRAN コンパイラは自動ベクトル化機能を強化して、原型プログラムのままで高速型処理を行うものが増えており、利用者側の負担は軽減されつつある。

(ロ) リストベクトルを用いる場合

```

DO 1 J=1, NY
  DO 1 I=1, NX          (原型)
1 U(I, J)=A*U(I-1, J)+B*U(I, J-1)
  ↓
DO 10 L=1, NX+NY-1
  DO 10 K=LL(L)+1, LL(L+1)
    I=II(K, L)
    J=JJ(K, L)          (中間型)
10 U(I, J)=A*U(I-1, J)+B*U(I, J-1)
  ↓
DO 10 K=1, NX*NY
  I=L(K)                (高速型)
10 UU(I)=A*UU(I-1)+B*UU(I-NX)

```

原型の逐次計算を座標回転法で独立演算 (中間型) とし、さらにリストベクトル L(K) によって L と K を統合化して配列次元を下げ、高速化している。

例3. 並列パイプラインを活用する方法 (技法C)

```

DO 1 J=1, N
  DO 2 K=1, N
2 C(K, J)=0. DO          (原型)
  DO 1 I=1, N
    DO 1 K=1, N
1 C(K, J)=C(K, J)+A(K, I)*B(I, J)
  ↓
DO 10 J=1, N, 2
  DO 20 K=1, N
    C(K, J)=0. DO
20 C(K, J+1)=0. DO      (高速型)
  DO 10 I=1, N, 2
    DO 10 K=1, N
      C(K, J)=C(K, J)+A(K, I)*B(I, J)
      +A(K, I+1)*B(I+1, J)
10 C(K, J+1)=C(K, J+1)+A(K, I)*B(I, J+1)
  +A(K, I+1)*B(I+1, J+1)

```

原型に対して、高速型では最内側ループの積と和の



演算を4倍に増やして並列演算性能や転送パイプラインのロード・ストア回数低減を活用している。なお、技法Cのような変更はプログラムの簡潔表現を犠牲にしており、本来のプログラム技法からは好ましくない。そこで最近の FORTRAN コンパイラでは、原型のまま高速型に自動展開する例もみられる。

例4. メモリ衝突を避ける方法 (技法D)

(イ) 配列宣言に仕掛けをする場合 (64バンクの例)

```

DIMENSION A(64, 64)
DO 1 K=1, 64
  DO 1 I=K+1, 64          (原型)
    T=-A(I, K)/A(K, K)
    DO 1 J=K+1, 64
      1 A(I, J)=A(I, J)+T*A(K, J)
    ↓
DIMENSION A(65, 64)
DO 1 K=1, 64
  DO 1 I=K+1, 64          (高速型)
    T=-A(I, K)/A(K, K)
    DO 1 J=K+1, 64
      1 A(I, J)=A(I, J)+T*A(K, J)
  
```

原型では、A (I, 1)~A (I, 64) が同一バンクに配置

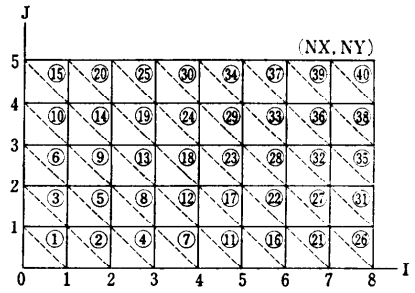


図-2 座標回転法によるリストベクトルの作成手順

されるので最内側ループでメモリ衝突する。高速型ではデータの配置を変えてメモリ衝突を避けている。

(ロ) リストベクトル作成の場合

例2の(ロ)のプログラム例におけるリストベクトル L(K) の作成方法の例を図-2 および表-6 に示す。

この方法によって、大規模疎行列や帯行列における連立一次方程式の反復解法 (PCG 法, PBCG 法など) が高速化される。

以上の例のほかに、技法Aに関連してサブルーチンや関数のインライン展開, GO TO 文によるループの DO ループ化, 技法Cに関連して幾つかのループを一

表-6 リストベクトルの作成法

表-6のa 座標回転法による変数Kの変域 (KS ≤ K ≤ KE)

L	1	2	3	4	5	6	7	8	9	10	11	12
KS(L)	1	2	4	7	11	16	21	26	31	35	38	40
KE(L)	1	3	6	10	15	20	25	30	34	37	39	40

表-6のb KS(L) と KE(L) の LL(L) による統合化

L	1	2	3	4	5	6	7	8	9	10	11	12	13
LL(L)	0	1	3	6	10	15	20	25	30	34	37	39	40

$$KS(L) = LL(L) + 1, \quad KE(L) = LL(L + 1)$$

$$NX + NY = N + 1 = 13, \quad NX * NY = LL(N + 1) = 40$$

表-6のc 変数Iのリストベクトル L(K) (I=L(K))

K	L(K)	K	L(K)	K	L(K)	K	L(K)
1	1+8*0=1	11	5+8*0=5	21	7+8*0=7	31	8+8*1=16
2	2+8*0=2	12	4+8*1=12	22	6+8*1=14	32	7+8*2=23
3	1+8*1=9	13	3+8*2=19	23	5+8*2=21	33	6+8*3=30
4	3+8*0=3	14	2+8*3=26	24	4+8*3=28	34	5+8*4=37
5	2+8*1=10	15	1+8*4=33	25	3+8*4=35	35	8+8*2=24
6	1+8*2=17	16	6+8*0=6	26	8+8*0=8	36	7+8*3=31
7	4+8*0=4	17	5+8*1=13	27	7+8*1=15	37	6+8*4=38
8	3+8*1=11	18	4+8*2=20	28	6+8*2=22	38	8+8*3=32
9	2+8*2=18	19	3+8*3=27	29	5+8*3=29	39	7+8*4=39
10	1+8*3=25	20	2+8*4=34	30	4+8*4=36	40	8+8*4=40

つにまとめるなどの方法が考えられる。言語コンパイラの自動ベクトル化機能を強化して、技法 B や C の一部などを原型のままですべて高速化するケースもある。

## 5. 行列計算への適用

数値シミュレーションの主要な課題は、時空における場の物理量の解析である。場の物理量は、通常、なんらかの保存法則によって偏微分方程式で記述される。偏微分方程式を離散化すると、連立一次方程式（または固有値問題）になる。ここでは特に、現代のスーパーコンピュータの重要な課題となっている3次元移流拡散方程式を取り上げ、その数値解析法とスーパーコンピュータへの適合性について述べる。

移流拡散方程式は次のように記述される。

$$\frac{\partial \mathbf{u}}{\partial t} + \text{div}(-k \nabla \mathbf{u} + \mathbf{b} \mathbf{u}) = f \quad (7)$$

拡散項    移流項

ここで、 $\mathbf{u}$  は場の物理量、 $k$  は拡散係数、 $\mathbf{b}$  は移流速度、 $f$  は資源（ソース）である。 $k$  や  $\mathbf{b}$  が  $\mathbf{u}$  に依存するか否かで非線形または線形問題に分かれる。 $\mathbf{u}$  として流速  $\mathbf{b}$  を選べば流れの方程式（Navier-Stokes 方程式）になり、解析の面倒な非線形問題となる。

式(7)を離散化（差分法、有限要素法など）すると連立一次方程式になる。連立一次方程式の規模は、離散化の格子点数によって決まる。空間を  $N \times N \times N$  の格子点に離散化し、各点で  $q$  個の物理量を得ようとすれば、たとえば7点差分法の場合の係数行列は、帯幅7で元数  $q \times N^3$  の疎行列になる。流れの方程式では、熱流を考えなければ  $q=5$ （3方向の流速、密度、圧力）であるが、その場合に  $N=100$  とすれば係数行列の非零要素の数は  $35 \times 10^6$  となる。各物理量を64ビット浮動小数点（8バイト）で表現すれば、この問題を解くのに必要な主記憶の容量は280MBとなる。現在のスーパーコンピュータの主記憶容量の最大値は256MBであるから、この例の程度の規模が計算限界になる。拡張記憶2~4GBを使えば、 $N=200$  クラスまでとなる。

式(7)の移流項がなければ、係数行列は対称になり記憶容量は先の見積り半分で済む。純拡散問題や構造解析で扱う係数行列は対称である。

移流項と拡散項の比率に応じて数値解析の難易度が決まる。この比率を与える目安には、レイノルズ数 ( $Re$ ) とセルペクレ数 ( $Pec$ ) がある。レイノルズ数は移流速の大きさ ( $|\mathbf{b}|$ ) と拡散係数 ( $k$ ) の比に物理対象（流体中の物体）の大きさ ( $a$ ) を掛けた数（物理側）

であり、セルペクレ数は移流速と拡散係数の比に格子間隔 ( $\Delta x$ ) を掛けた数（数値解析側）でありいずれも無次元量である。

$$\begin{cases} Re = |\mathbf{b}| a / k & (a \sim n \Delta x) \\ Pec = |\mathbf{b}| \Delta x / k \end{cases} \quad (8)$$

あるレイノルズ数の流れを高精度で数値解析するには、 $Re^{9/4}$  の格子点数が必要になる<sup>10)</sup>。たとえば、乱流の発生するような物理現象 ( $Re \geq 10^4$ ) の解析には、 $10^9 = 1000^3$  ( $N=1000$ ) の格子点が要求されるが、現状のスーパーコンピュータは  $N=100 \sim 200$  の段階にある。

一方、 $Pec \leq 1$  の条件が満たされると、係数行列は対角優位（詳しくは  $M$ -行列；すなわち行列  $A$  の要素が  $a_{ii} > 0$ ,  $a_{ij} \leq 0$  ( $i \neq j$ ),  $A^{-1} \geq 0$ ) になり、数値解析の信頼性（安定性）が得られる。数値解析の側の主要課題は、数値解析上の安定性を  $Pec$  数のどこまで広げられるかということになる。

連立一次方程式の高速算法の例を以下に示す<sup>4)</sup>。

直接解法	{	改訂コレスキー法……………対称行列
		ガウス消去法 (またはクラウト法)……………一般行列
反復解法	{	前処理付き共役勾配法 (PCG 法)……………対称行列
		前処理付き双対共役勾配法 (PBCG 法)……………一般行列

直接解法は反復解法に比べて、CPU 時間が少なく済む（通常 1/3 以下）ので、主記憶容量に余裕のある問題に対して有効である。直接解法の計算主要部は  $U \cdot D \cdot U$  分解 ( $D$  は対角、 $U$  は上三角行列) や  $LDU$  分解 ( $L$  は下三角行列) ( $LU$  分解) であり、ベクトルの内積または積和である。したがって、演算パイプライン方式のスーパーコンピュータにきわめて良く適合する。並列パイプラインに対しては、2列同時2列消去などの方法（3. の技法 C）でさらに高速化できる。

主記憶（または拡張記憶）の限界近くで計算する問題に対しては、使用記憶量の少ない反復解法が必須になる。反復解法の良し悪しの目安は、与えられた精度（相対残差）をいかに少ない反復回数で得ることができるかということになる。スーパーコンピュータにおいてはこのほかに、解法のベクトル化と高速化（加速率向上）が重要になる。

$Pec=0$ （移流項なし）の場合に反復回数が少ない算法としては ICCG 法（不完全コレスキー分解による共役勾配法）が知られる。これを改良した MICCG 法（改良 ICCG 法）の反復回数はさらに少ない。これらの方法（まとめて PCG 法）は、従来の反復解法

表-7 スーパーコンピュータの行列演算性能例

S-810/20

問題例	計算規模	解法	CPU 時間		加算率
			スカラ処理	ベクトル処理	
			sec	sec	
連立一次方程式	300 元	ガウス消去法	2.94	0.08	37.7
	500 元	ガウス消去法	13.42	0.28	47.7
固有値問題	200 元	ハウスホルダ二分法	11.49	0.82	14.1
	300 元	ハウスホルダ二分法	23.24	2.06	17.6
拡散問題	40×40×40	ICCG 法	110.0	1.5	73.3
移流拡散問題	40×20×20	ILUBCG 法	25.5	0.51	50.0
移流拡散問題	511×201	ヤコビ陽解法	1669.	18.9	88.3
乱流 LES 計算	32×32×21	スペクトル法	5.7	0.4	14.3

備考 加算率=スカラ処理 CPU 時間/ベクトル処理 CPU 時間

(SOR 法, 擬 SOR 法, SLOR 法, SPOR 法など) に比べて収束が速く, 通常, 1 桁以上高速化される. 演算パイプライン方式に対しては, リストベクトルを用いた座標回転法 (超平面法) によって高水準の加算率が得られる. スーパーコンピュータによっては, 反復回数を犠牲にして, 前処理のベクトル化で全体の高速化を得る例 (対角要素を正規化した共役勾配法 SCG 法) もある.

$Pec > 0$  (移流項あり) の場合は係数行列が非対称になるので問題は難しくなる.  $Pec > 1$  に対しても安定に収束する解法として ILUBCG 法 (不完全 LU 分解による双対共役勾配法) および MILUBCG 法 (改良 ILUBCG 法) (合わせて PBCG 法) が知られる. このほかに ILUCGS 法や ILUCR 法などもあるが, 問題によって収束性に変動がある. ただし, 従来法 (SOR 法など) に比べれば遙かに安定で高速な解法である.

$Pec > 0$  の場合は,  $Re^{3/4}$  則によって格子点数の制約があり, 現状のスーパーコンピュータでは

$Re \leq 400$  (主記憶制限)

$Re \leq 2000$  (拡張記憶制限)

となる. 拡張記憶を用いて  $Re$  の大きい問題や複雑な形状 (有限要素法, 帯幅が増す) の問題を解析するためには, 物理対象を部分構造に分けて連立させる方法 (部分構造反復法) と PBCG 法などを組み合わせる方法が考えられる<sup>11)</sup>. 表-7 に行列演算の性能例を示す.

## 6. まとめ

スーパーコンピュータの現状を概観し, その特徴と高速化技法の例を述べ, 行列計算への適用性について論じた. 現在, 移流拡散問題に対しては  $Re \leq 2000$  の制約下で数値シミュレーションの信頼性が得られる.

$Re \geq 10000$  の問題に対しては, 4. ( $Re^{3/4}$  則) で

も述べたように  $10^9$  格子点が必要になり, 5 つの物理量を 7 点差分法で計算するには 280 GB の容量が要る. これは現状の主記憶 256 MB の 1000 倍強の容量であり, それに見合う演算速度を合わせると現在の 1000 倍の記憶容量と演算速度が必要になる.

## 参考文献

- 1) 日本物理学会編: スーパーコンピュータ, 培風館 (1985).
- 2) Hockney, R. W. and Jesshope, C. R.: Parallel Computers, Adam Hiliter Ltd. (1981); (邦訳) 奥川, 黒住訳: 並列計算機, 共立出版 (1984).
- 3) 日経データプロ: 国産スーパーコンピュータのすべて, 日経マグロウヒル社 (1985).
- 4) 村田健郎, 小国 力, 唐木幸比古: スーパーコンピュータ-科学技術計算への適用-, 丸善 (1985).
- 5) 日本 CDC: ETA<sup>10</sup> スーパーコンピュータ・システム (1987. 6. 11).
- 6) 日立製作所: S-820 アレイプロセッサ・システム (1987. 6. 30).
- 7) 片山 博: SX-1 FORTRAN 77 の概要, SEN-AC, Vol. 19, No. 2 (1986. 4); SX-1 および SX-2 以外のデータは 1987 年 6 月 18~30 日に大学ネットワーク (東京大学および京都大学) を通じて測定した最新のものである.
- 8) Perrenod, S., 榎原 悟: DSP 9000 シリーズ (Alliant 社 FX シリーズ), コンピュートロール, No. 19, コロナ社 (1987).
- 9) Hwang, K.: Computer Arithmetic—Principles, Architecture and Design—, John Wiley & Sons Inc. (1979); (邦訳) 堀越 彌監訳: コンピュータの高速演算方式, 近代科学社 (1980).
- 10) 吉沢 徹: 乱流の数値シミュレーション, スーパーコンピュータと数値シミュレーション・シンポジウム報文集, 東京大学大型計算機センター (1983. 3).
- 11) 村田健郎, 長谷川秀彦: 情報処理学会数値解析研究会 13-1 (1985. 7).

(昭和 62 年 7 月 6 日受付)