U.D.C. 681.322-181.2-185.4.001.7

スーパーコンピュータシステムの展望

A Perspective of Supercomputer Systems

長島重夫* *Shigeo Nagashima*
小高俊彦** *Toshihiko Odaka*

スーパーコンピュータは，大規模な解析・実験，更には設計を数値計算だけによって可能とする必す(須)の手段となりつつある。本論文では，まず，現在のスーパーコンピュータの基本方式となっているベクトル演算方式の概要と高速化の課題を述べる。次に，これまでのスーパーコンピュータが高速化に向けていかに工夫してきたかについて，ベクトル演算性能向上とベクトル化率向上の観点から分析する。更に将来のスーパーコンピュータについて展望する。今後の性能向上には演算器数を大幅に増加させるために複数のプロセッサで構成する方式が中心になると予想される。この実現方式として並列プロセッサが注目されている。また数値計算分野から，より広範な分野への適用範囲の拡大も期待される。

1 緒　　言

スーパーコンピュータを使用することにより，数値計算だけによって大規模な解析，実験，更には設計まで行う方式が注目されている。すなわち，数値計算に必要な費用が実際に大規模な実験を行うのに比べてはるかに少ないこと，実際には目に見えない，分からないことを計算によって知ることができること，設計期間を大幅に短縮できることなどのメリットがあるからである。一方，半導体や実装技術の絶え間ない進展によって，このようなスーパーコンピュータの実現が現実化しつつあることも，高速コンピュータへの期待を高める一因となっている。

スーパーコンピュータは，その時代の最高位のはん(汎)用コンピュータよりも格段に高速処理が可能なコンピュータに冠せられる名称である。現在のはん用コンピュータの浮動小数点演算性能は，およそ20MFLOPS(Million Floating-point Operations Per Second，10^6演算/秒)である。これに対し，最新のスーパーコンピュータはこの100倍，2GFLOPS(Giga FLOPS，10^9演算/秒)の演算が可能になっている。

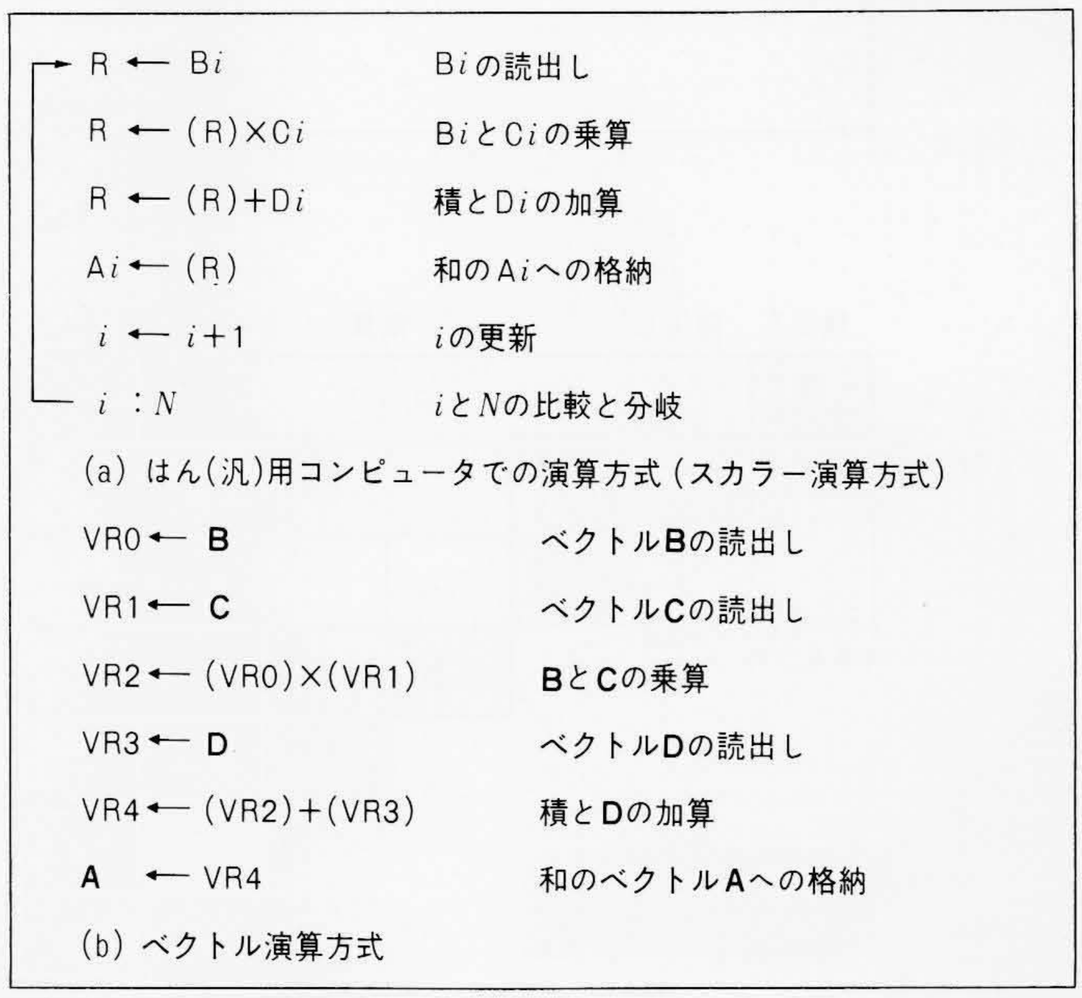
本論文では，この高速性がどのように実現されているかについて述べるとともに，主としてハードウェアの視点からこれまでのスーパーコンピュータの発展過程を分析し，今後のスーパーコンピュータの発展方向を展望する[1),2)]。

2 スーパーコンピュータの基本方式と高速化の課題

一般に科学技術計算では，

$$Ai=Bi\times Ci+Di \quad (i=1,\ 2\cdots,\ N)$$

のような繰返し演算が計算全体の多くの時間を占める。このような繰返し演算をはん用コンピュータでは**図1**(a)のように処理する。あるiについての処理に6命令を必要とし，すべてのiについて処理するためには$6N$命令を要することになる。一方，Ai，Bi，Ci，Di($i=1,\ 2\cdots,\ N$)のデータをそれぞれまとめてベクトル**A**，**B**，**C**，**D**と考え，このベクトルを単位として演算する方式が考えられる。これがベクトル演算方



注：略語説明　R(レジスタ)
VRi(i番ベクトルレジスタ)

図1　$Ai=Bi\times Ci+Di(i=1,\ 2,\cdots,\ N)$の演算式　はん(汎)用コンピュータでは各データごとに命令の実行が必要で，一つのiの処理に6命令，全体で$6N$命令を要する。一方，ベクトル演算方式では，ベクトルを単位として演算するので6命令で済む。

* 日立製作所中央研究所　** 日立製作所神奈川工場

式である。上記の繰返し演算の場合には同図(b)のように演算することになる。すべての演算を6ベクトル命令で処理することができる。はん用コンピュータでの演算方式は，ベクトルという用語に対比してスカラー演算方式と呼ばれている。

ベクトル演算を分析すると，

(1) ベクトルのすべての要素に対して同一演算を行う。

(2) あるiと他のiの間に相互干渉がない。

ことが分かる。このことから，ベクトル演算を**図2**(b)に示すようなパイプライン演算で実行することができる。この場合，あるiの処理と次のiの処理は1サイクル分時間をずらして実行されるので，iが十分大きければ等価的に一つのiを1サイクルで処理できることになる。スカラー演算の場合には同図(a)に示すように，あるiの処理が完了してから次のiの処理を開始するので，一つのiの処理に13サイクルを要する。この場合には，ベクトル演算方式ではスカラー演算方式より13倍高速化できることになる。

このようなベクトル演算は，科学技術計算の大部分に対して適用することができるが，すべてをこれで実行できるわけではない。あるプログラムをはん用コンピュータで実行した場合の実行時間をTsとし，このうちベクトル演算が可能な部分の割合(ベクトル化率)をα，ベクトル演算による性能向上率をVとすると，スーパーコンピュータでのこのプログラムの実行時間Tvは次式で表される(**図3**参照)。

$$Tv=\{(1-\alpha)+\alpha/V\}\,Ts \quad \cdots\cdots(1)$$

したがって，プログラム実行からみた総合性能向上率$P(=Ts/Tv)$は次式のようになる。

$$P=Ts/Tv=1/((1-\alpha)+\alpha/V) \quad \cdots\cdots(2)$$

VとPの関係を幾つかのαの場合について示すと**図4**のようになる。同図から明らかなように，ベクトル化率αが十分大きくないと総合性能向上率PはVに近づかない。したがって，高速のスーパーコンピュータを実現するには，

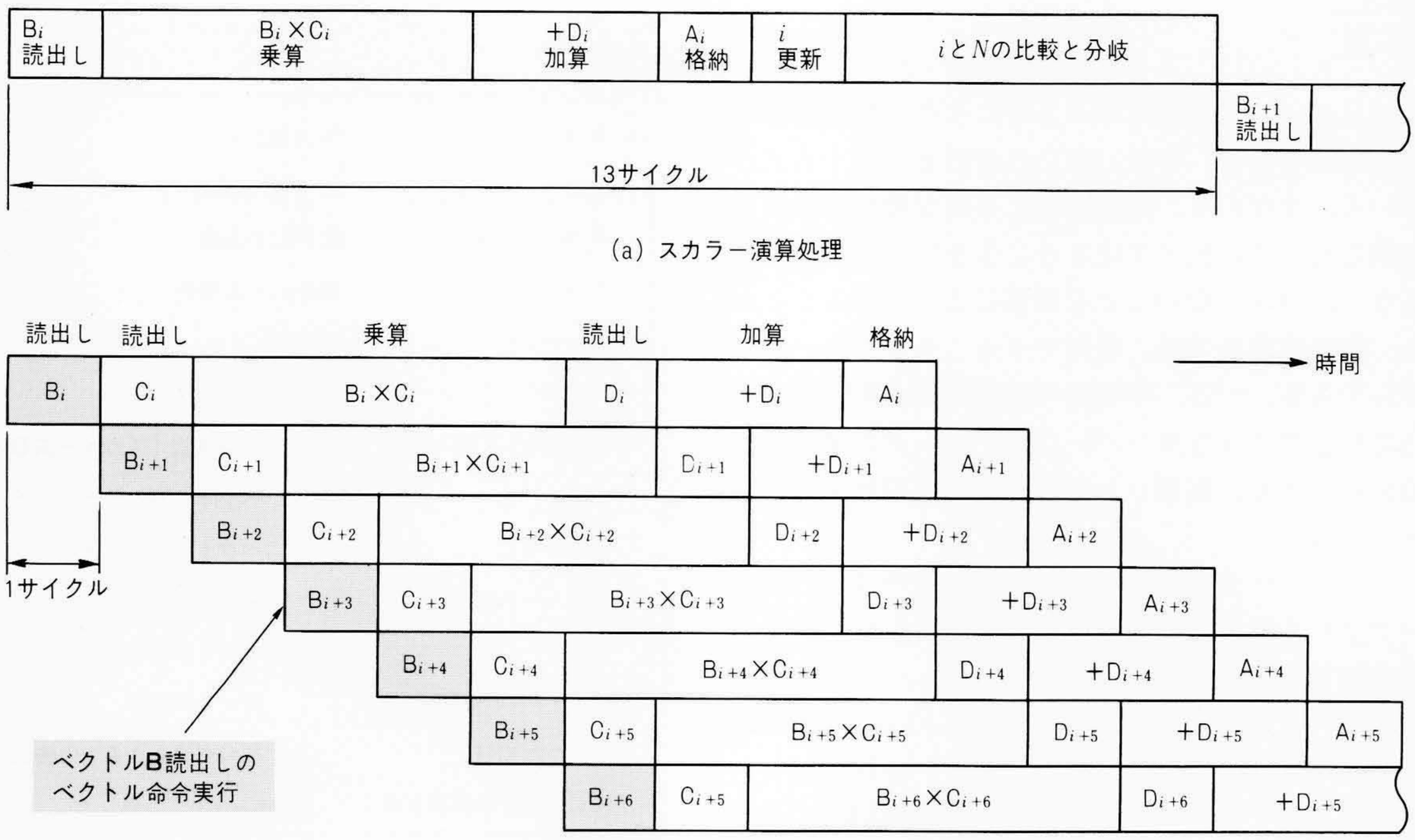


図2 スカラー演算処理とベクトル演算処理の比較　ベクトル演算では，一つのベクトル命令で多数のベクトル要素を操作することができる。また複数のベクトル命令を並列に実行することができ，これらによって一つのiについての演算を1サイクルで処理することができる。

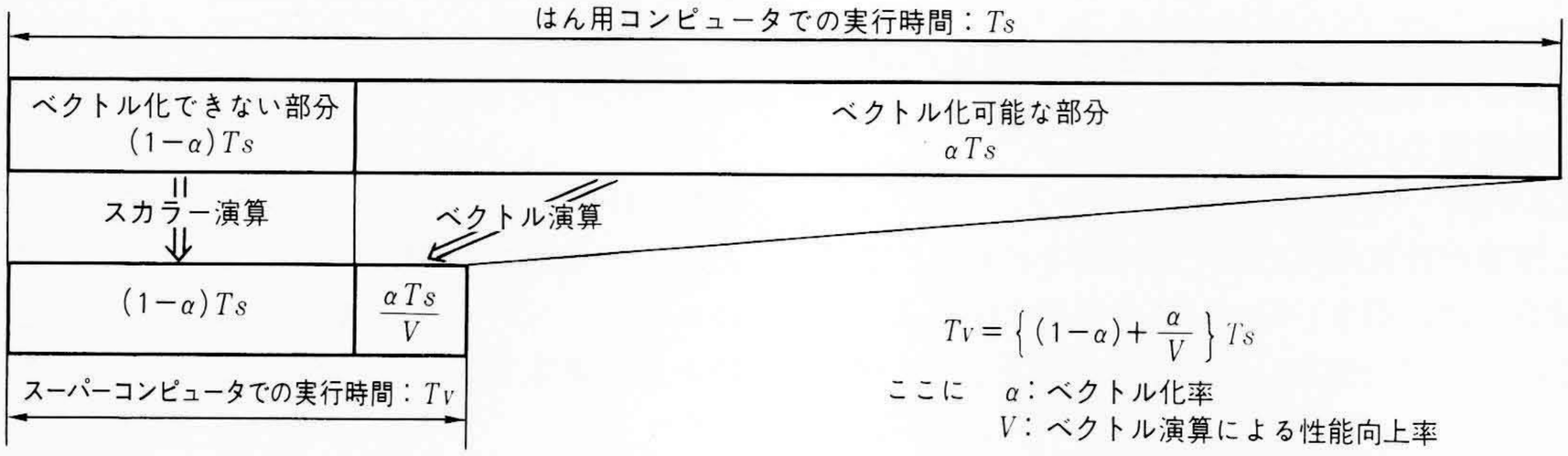


図3 スーパーコンピュータによるプログラム実行時間の短縮　ベクトル化可能な部分はベクトル演算方式によって大幅に短縮されるが，ベクトル化できない部分はスカラー演算方式によるので短縮されない。

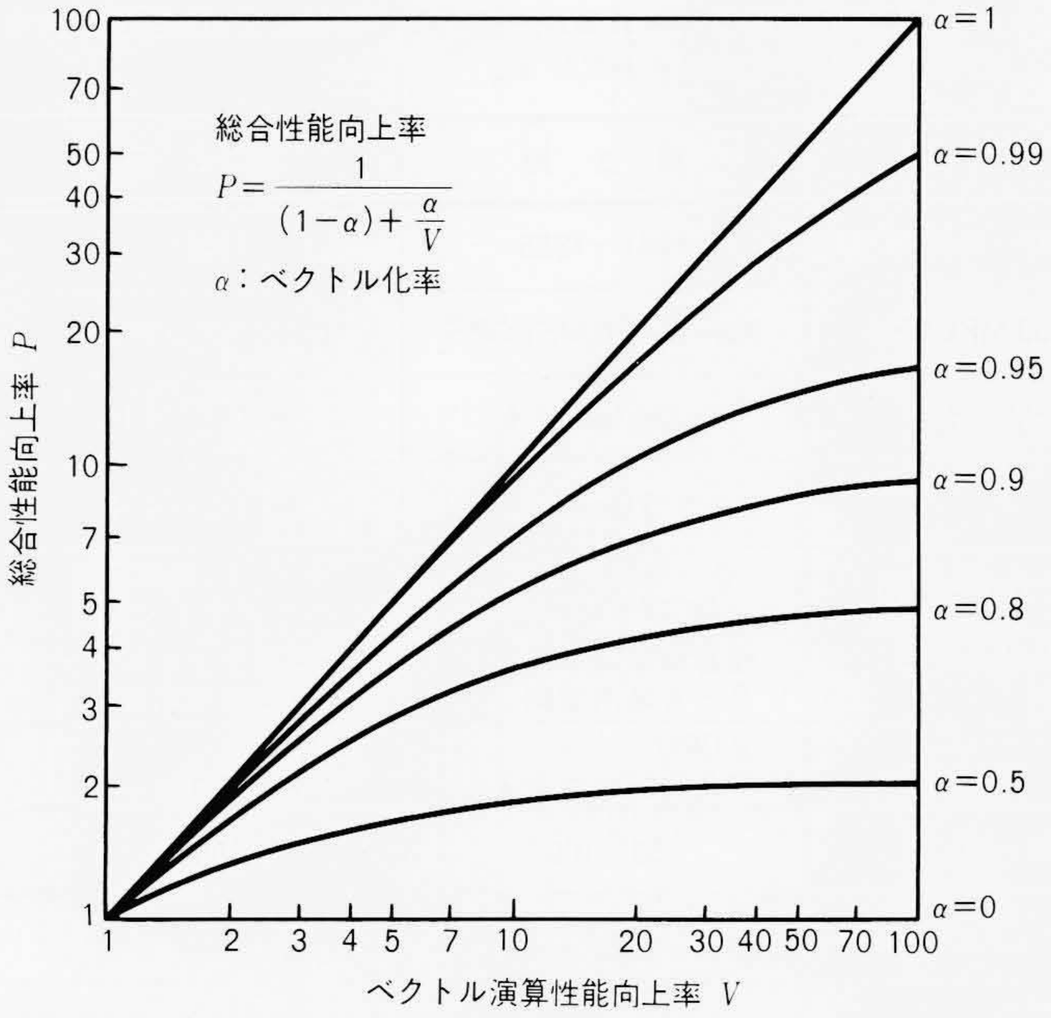
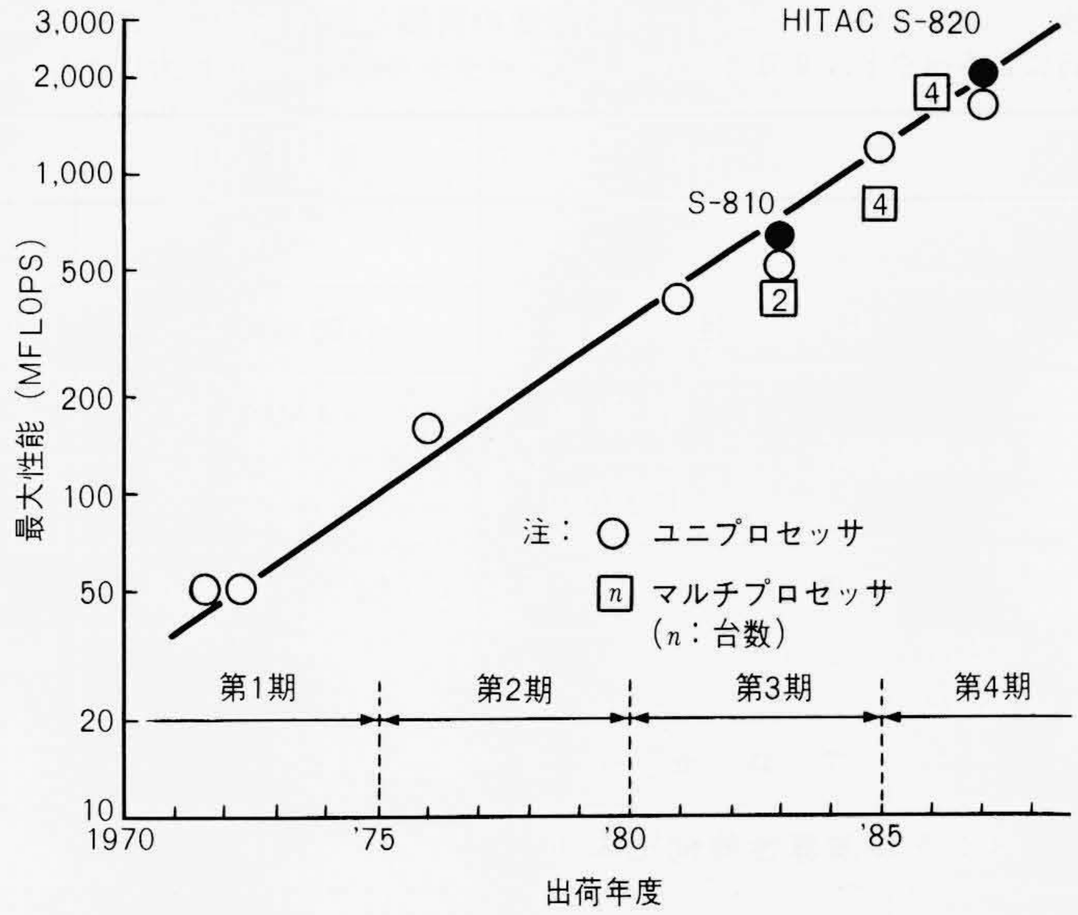


図4　ベクトル演算性能向上率と総合性能向上率との関係　　ベクトル化率を十分大きくしないと，総合性能向上率はベクトル演算性能向上率に近づかない。ベクトル化率の向上が重要である。



注：略語説明　MFLOPS〔Million Floating-point Operations Per Second（10^6演算/秒）〕

図5　スーパーコンピュータの最大性能のトレンドと期の区分
1970年代に登場した第Ⅰ期のスーパーコンピュータ以降，最大性能は数十倍向上している。現在は第4期にある。

(1)　ベクトル演算の高速化（Vの向上）
(2)　ベクトル化率の向上（αの向上）
が必要であり，また残された部分の高速化のために，
(3)　スカラー演算の高速化
も重要である。いつの時代のスーパーコンピュータも，これらを追及し高速化を図ってきている。

3　スーパーコンピュータの発展過程

現在主流となっているスーパーコンピュータ高速化技術は，1970年代に実用化された。これ以前にも科学技術計算専用コンピュータがあり，やはりスーパーコンピュータと呼ばれていたが，この演算方式はスカラー方式であった。ベクトル演算が導入されたのは1970年代前半であり，これ以降のスーパーコンピュータはすべてこのベクトル演算方式を採用してきている。1970年代前半のマシンを第1期とすると，現在までのスーパーコンピュータは性能の視点から4期に分けられる。**図5**に最大性能のトレンドと期の区分を，**表1**にこの4期についての概要と技術動向を示す。同表には各期のベクトル演算の性能向上率Vとベクトル化率αの概略値も合わせて示してある。

これまでのスーパーコンピュータの発展過程を，ベクトル演算性能の向上とベクトル化率の向上の観点から分析する。

(1)　ベクトル演算性能の向上方式

これまでの十数年間に，ベクトル演算性能は数十倍の高速化がなされている。この大幅な性能向上は，方式的には各種のレベルで処理を並列化することによって実現されている。処理の並列実行方式の発展を整理すると，**表1**の並列実行レベルの項目のようにまとめることができる。

並列実行は次のような幾つかのレベルに分けられる。

(a)　演算の並列実行

前述のように，スーパーコンピュータは基本的にはパイプライン演算によって数個のベクトル要素の演算を同時に実行し，高速化を実現している。第3期に入って更に複数のベクトル要素を同時に演算する並列パイプライン演算が導入された。この方式は，例えば，同一のパイプライン演算器を2個設け，一方は偶数番号の，もう一方は奇数番号のベクトル要素を同時に演算させることによって，演算性能を2倍向上させる方式である。第3期は2～4要素の並列演算であった。第4期には4要素の並列演算が一般化してきている。

(b)　ベクトル命令の並列実行

第2期になって，主記憶と演算器間にベクトルレジスタが導入された。この主目的は，演算途中の結果を毎回主記憶に戻すことを不要とすることであったが，ベクトルレジスタを利用することによって，前命令の演算がすべて完了する前に次の命令が開始できる，すなわちベクトル命令の並列実行ができるチェイニング方式が実現された。ただし，このチェイニングには前後の命令間の時間関係に制約があり，並列実行が可能な場合が限定されていた。第3期以降は，前命令の実行状態に無関係に後続命令のチェイニングが実行できる技術が確立され，命令の並列実行の可能性が大幅に向上された。これによって，初めて**図2**(b)に示すようなパイプライン演算が実現された。

(c)　マルチプロセッサによる並列処理

このレベルの並列処理は第3期から一部のスーパーコンピュータに導入されたものである。この方式はベクトル演算の性能を向上させるとともに，ベクトル処理が不可能な部分についても処理時間を短縮できる可能性を含んでいる。

表1 スーパーコンピュータの発展の概要と技術動向 スーパーコンピュータの発展は4期に分けられるが，この間の性能向上は処理の並列実行によるベクトル演算性能の向上と，ベクトル演算範囲の拡大によるベクトル化率の向上によって実現されている。

項目		第1期	第2期	第3期	第4期
年代		～1975	1976～1980	1981～1985	1986～
最大性能		～100 MFLOPS	100～500 MFLOPS	500～1,000 MFLOPS	1,000 MFLOPS～
最大記憶容量	主記憶	～4 Mバイト	～32 Mバイト	～256 Mバイト	～1 Gバイト
	拡張記憶	—	—	～3 Gバイト	～12 Gバイト
並列実行レベル	演算	パイプライン演算	パイプライン演算	2～4要素並列 パイプライン演算	4～(8)要素並列 パイプライン演算
	命令	—	限定されたチェイニング 2命令並列	任意のチェイニング 2～4命令並列	任意のチェイニング 2～4命令並列
	プロセッサ	—	—	2 CPU マルチプロセッサ	4 CPU マルチプロセッサ
ベクトル演算性能向上率 V		～20	20～50	50～100	100～
ベクトル演算種類	四則演算	○	○	○	○
	マクロ演算(内積・総和)	○	○	○	○
	条件付き演算	×	○	○	○
	間接指標ベクトルを含む演算	×	×	○	○
ベクトル化率 α		≈0.5	0.6～0.7	≈0.8	≈0.8

注：略語説明など MFLOPS(Million Floating-point Operations Per Second)，CPU(中央処理装置)
○(可能)，×(不可能)

しかし，現在のところでは，演算や命令レベルの並列実行がハードウェアやコンパイラによってユーザーにほとんど負担をかけることなく実現されているのに対し，プロセッサレベルの並列処理はユーザーがすべて指示する必要がある。

以上のような並列実行によって，第4期のスーパーコンピュータのベクトル演算性能は，スカラ演算性能に対して，パイプライン演算で約10倍，要素並列演算によって4倍，命令の並列実行あるいはマルチプロセッサによって2～4倍向上し，全体として100倍以上の性能向上率を実現しているとみることができる。

(2) ベクトル化率の向上方式

ベクトル処理の対象となる演算種類の観点から各期のスーパーコンピュータのベクトル演算の実現範囲を整理すると，**表1**のベクトル演算種類の項目のようになる。四則演算，マクロ演算(総和・内積演算など)は当初から実現されているが，条件付き演算(繰返し演算中にIF文――演算条件を指定する文――を含む演算)が可能になったのは第2期から，また間接指標ベクトル(あるベクトルの要素番号が他のベクトルによって指定されるベクトル)を含む演算は第3期から可能となった。このようなベクトル演算が可能となった背景には，FORTRANベクトル化コンパイラ技術の向上も大きく貢献している。

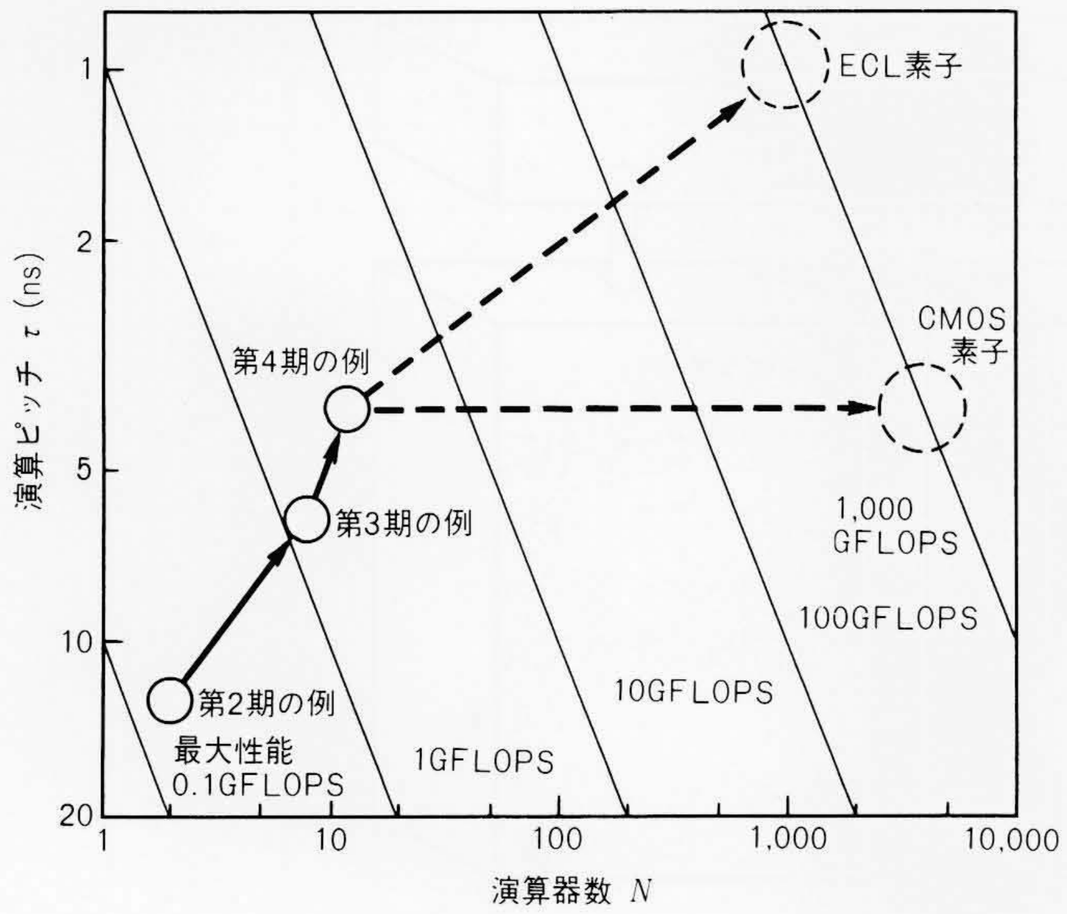
ベクトル化率はプログラムにより大幅に異なるが，現在では典型的な大規模科学技術計算プログラムのうち，約80%の部分をベクトル演算で処理することができる。残りの20%の部分は繰返し演算でなかったり，繰返し演算であってもベクトル化が不可能なものであり，マクロにはこれ以上のベクトル化の向上は望めないところまできていると言える。

4 スーパーコンピュータの今後の展望

4.1 スーパーコンピュータの性能と構成方式

先にも述べたように，スーパーコンピュータの高速化はベクトル演算性能の向上とベクトル化率の拡大に支えられてきたが，後者はある程度限度に達したとみることができる。したがって，今後はベクトル演算性能のいっそうの向上と，ベクトル化できない部分を高速に処理するためのスカラー処理性能の向上にかかることになる。

ベクトル演算の最大性能は，基本的には同時動作可能な演算器数Nと，その演算器が結果を出力するピッチτで決定され，N/τが最大(ピーク)ベクトル演算性能となる。ここでピッチτは，主として半導体素子技術及び実装技術によって決定され，この短縮はハードウェア テクノロジーの進展に依存するが，今後これを大幅に短縮するには新しい素子の開発を待つ必要があろう。現在の半導体技術の延長で考えると演算器数Nの増加が性能向上のかぎとなる。Nとτの観点から将来のスーパーコンピュータを予測すると**図6**のようになろう。現在は例えばHITAC S-820ではNは12，τは4nsで，3GFLOPSのレベルにある。今後もこれまでの性能向上のトレンドが継続し，いずれは1,000GFLOPS，すなわち1TFLOPS(Tera FLOPS,



注：略語説明　GFLOPS〔Giga Floating-point Operations Per Second (10^9演算/秒)〕
ECL(Emitter Coupled Logic)
CMOS(Complementary Metal Oxide Semiconductor)

図6　演算器数と演算ピッチから見たスーパーコンピュータの予想
今後のスーパーコンピュータの性能は，演算器数の大幅な増加によって向上されよう。この場合使用する素子によって二とおりの実現方式が考えられる。

10^{12}演算/秒)の性能が実現されると予想される。

このような高速プロセッサの実現方式として，使用するLSI素子の観点から，現在のスーパーコンピュータと同じECL(Emitter Coupled Logic)素子を使用する方式とマイクロプロセッサなどで使用されるCMOS(Complementary Metal Oxide Semiconductor)素子を使用する方式が考えられる。

一般にECL素子は，CMOS素子と比べて数倍から一けた高速である。したがって，同一性能を実現するにはECL素子による演算器の数は，CMOS素子に比べて数分の一から十分の一で済むことになる。一方，LSIの集積度の点から比較すると，CMOS素子は低消費電力という特長を生かして，ECL素子よりも一けた以上高集積化することが可能である。このため演算器数を含めて考えると，実装体積はほぼ同等，消費電力はCMOS素子のほうが有利となる。この相違は大規模化するほど顕著に現れる。

ECL素子を使用する場合には，例えばτを1nsとすることができよう。こうすると演算器数は1,000台規模となる。τが極めて小さい時間であることを考慮すると，1プロセッサを大規模化することは信号遅延時間の点で不利である。したがって，すべての演算器を1台のプロセッサに収納するよりも，複数のプロセッサに演算器を分散させるほうが実現性は高いであろう。このように構成した例を**図7**(a)に示す。この構成は主記憶を共有するマルチプロセッサである。

一方，CMOS素子を使用する場合には，τはたかだか数ナノ秒であると予想される。したがって，演算器数は数千台が必要となる。このような多数の演算器を持つプロセッサの構成方式として，**図7**(b)に示すような並列プロセッサがある。この構成では各プロセッサが主記憶を共有しないので，プロセッサ間でデータを転送するためのなんらかの手段が必要になる。このためにネットワークが設けられている。並列プロセッサは今後重要な方式として，特に注目を集めている。

一方，スカラー演算性能の向上は，ベクトル演算性能の向上ほどには期待できないと予想される。スカラー演算方式は基本的にははん用コンピュータと同等であり，この性能はこの10年間に10倍程度の向上にとどまっているからである。ベクトル演算性能が向上するにつれて，ベクトルとスカラーの演算性能比は拡大する一方であり，スカラー演算の高速化は総合性能向上のために残された困難な課題のひとつになっている。

高速処理を実現するためには，基本的には高ベクトル化率のプログラムの開発にかかっている。この実現方法のひとつとして，例えば偏微分方程式求解用言語“DEQSOL”のようにユーザーインタフェースを高水準言語とし，高ベクトル化プログラムを自動生成する方式がある。また，高いベクトル化率，並列化率が実現できるアルゴリズムの開発が期待されるところである。

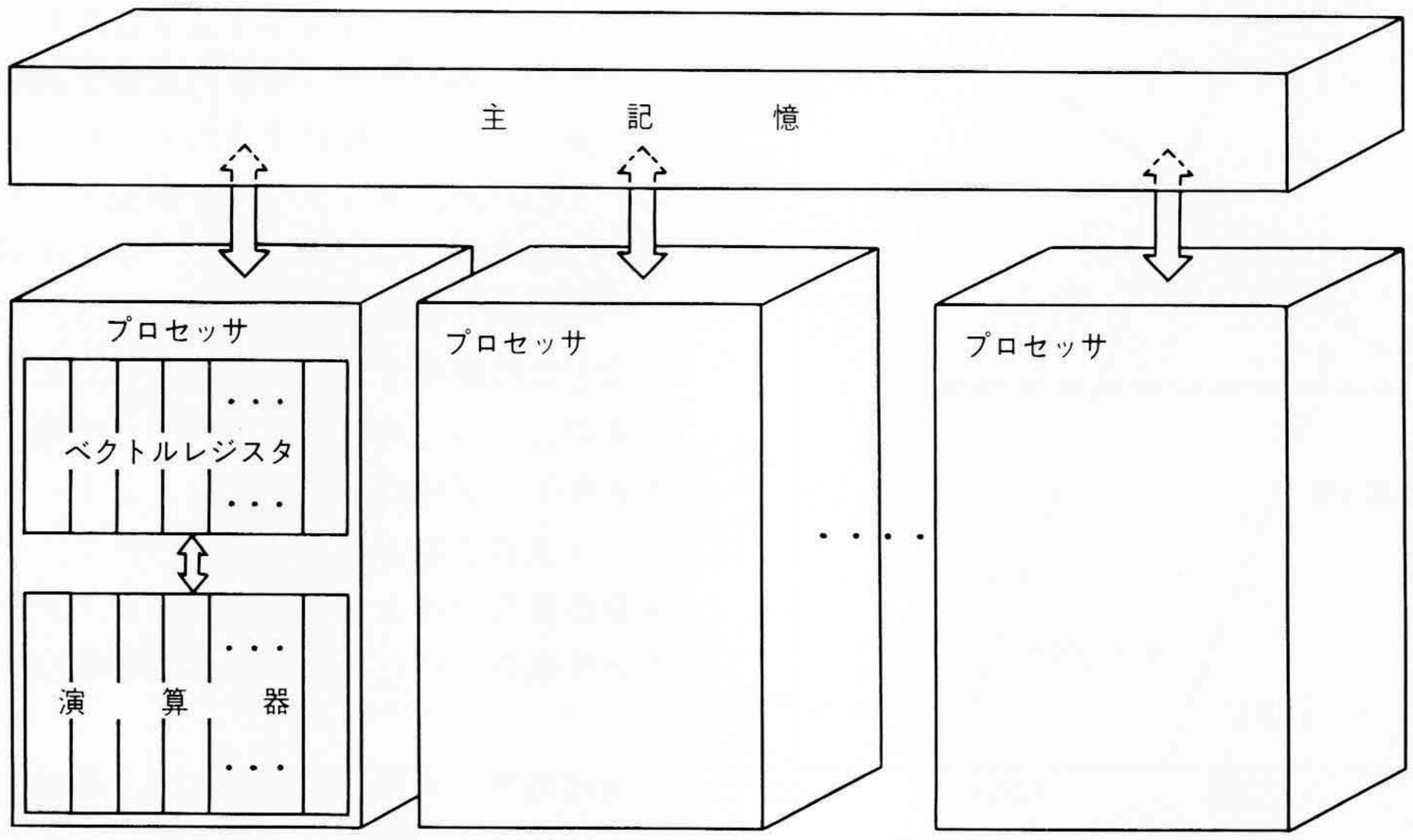
4.2　スーパーコンピュータの応用分野

現在のスーパーコンピュータの応用分野は，その大部分が科学技術計算分野である。これはこの分野の演算が，ベクトル演算に非常によく適しているからである。しかし，もう少しベクトル演算のスコープを広げてみると，多数のデータに同一処理を施す場合が多数あることが分かる。例えば，データベースの検索では多数のデータに対してあるデータを比較するという演算を行っている。HITAC M-68X　IDP(内蔵データベースプロセッサ)は，ベクトル処理によって関係データベース処理の性能向上を図った好例である[3]。コンピュータやLSIの開発に必すの論理シミュレーションも多数のゲートに対してAND，ORなどの論理演算を施すものであり，スーパーコンピュータ上での論理シミュレータが既に実現されている[4]。更には図形画像処理，知識処理分野でもベクトル演算・並列演算の研究開発が進められている。広範な分野に適用範囲を拡大することも，今後のスーパーコンピュータの課題である。

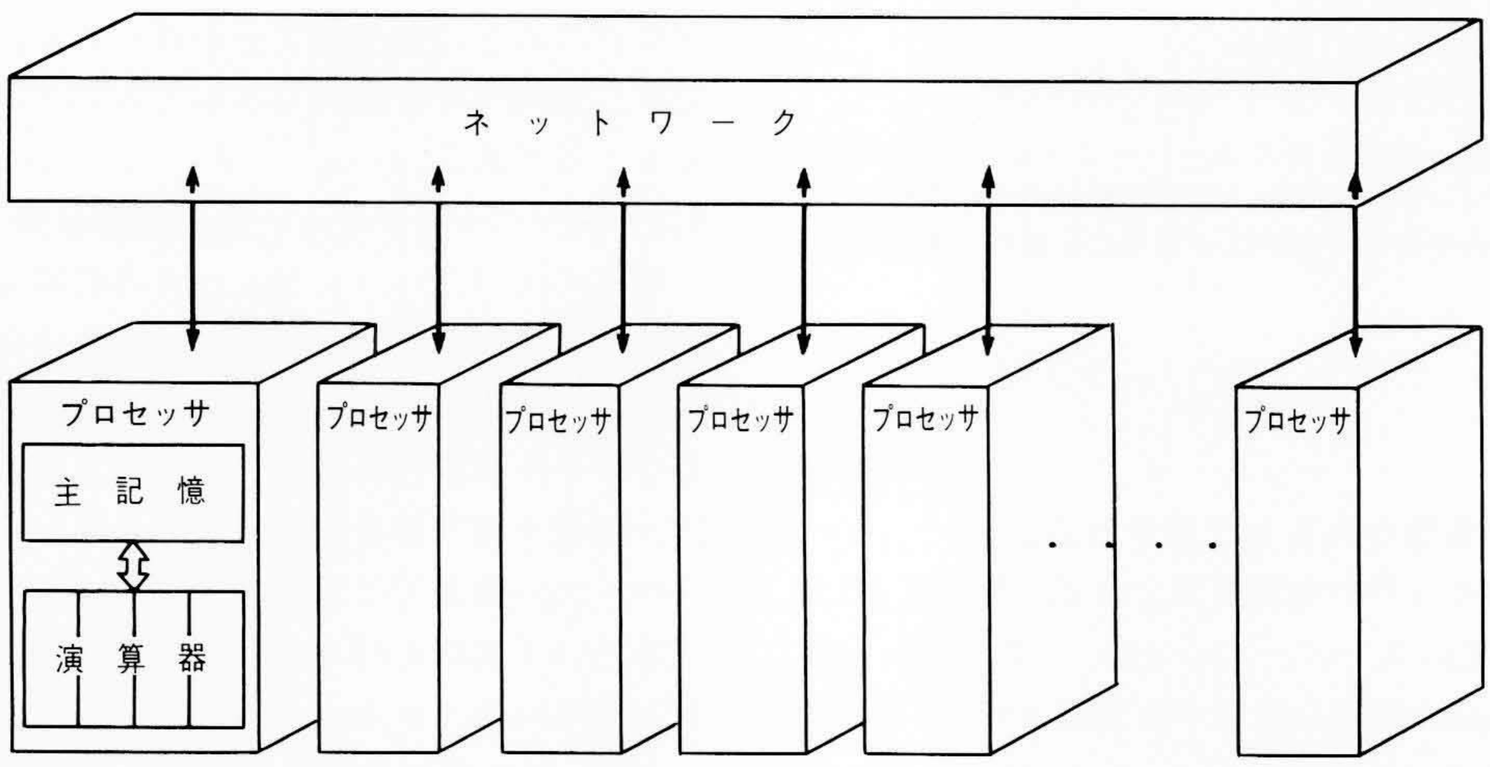
将来のスーパーコンピュータは，はん用ベクトル演算機能を持つことによって，科学技術計算分野だけにとどまらず，例えば知識処理を利用してプログラミングし，高速演算機能によってこれを解き，結果を画像処理によって高速に可視化し，グラフィックで表示することが当然の姿になると思われる。

5　結　　言

以上，スーパーコンピュータの発展過程をもとに将来のスーパーコンピュータを展望した。本論文ではほとんど触れることができなかったが，スーパーコンピュータの性能と使いやすさを決定するもうひとつの大きな要素はソフトウェアである。ユーザーが特に意識することなく高度なベクトル化や並列化を容易に行うことができるようにすることが必すであ



(a) マルチプロセッサ構成



(b) 並列プロセッサ構成

図7 将来のスーパーコンピュータの構成例 多数の演算器を実現する方式として，主記憶を共有したマルチプロセッサ構成方式と，多数のプロセッサをネットワークを経由して接続した並列プロセッサ構成方式がある。

る。また仮想記憶の実現や入出力の視覚化など，使いやすさの向上も要求されよう。

スーパーコンピュータを駆使することによって，新材料や新デバイスの開発，高度なシステムの実現，更には新分野の開拓を可能とする期待が一段と高まっていくであろう。不可能を可能とする高性能のスーパーコンピュータシステムの開発がいっそう切望されるところである。

参考文献

1) 堀越，外：コンピュータアーキテクチャ技術の動向とスーパーコンピュータ，日立評論，**65**，8，529～534(昭58-8)
2) 堀越：スーパーコンピュータ・アーキテクチャとその動向，情報処理学会「コンピュータ・システム」シンポジウム，11～18(昭60-12)
3) 鳥居，外：リレーショナル・データベースの処理速度向上を図るCPU内蔵型データベース・プロセッサ，日経エレクトロニクス 1987.2.19(No.414)，185～206(昭62-2)
4) S. Nagashima, et al.：Hardware Implementation of VELVET on the HITACHI S-810 Supercomputer, ICCAD '86, 390～394(1986-11)