

## 談話室



## データ・フロー計算機への別のコメント†

横井 俊夫\*\*

汎用のデータ・フロー計算機を目指す。すなわち、データ・フロー計算機を広義に解釈するという立場からコメントを行う。詳細は別の機会にし、ここでは要点のみを述べる。

データ・フロー計算機（機構）を次のようなものとする。 (1) 命令の実行は、オペランド（データ）が整い次第駆動される。従来の汎用計算機の命令は、プログラム・カウンタによって指示されると起動され、オペランドはそれ以前に整うようにプログラムするという機構とはちょうど裏返しの関係になる。これにより、計算の基盤が逐次から並列計算に置き換えられる。基盤が並列になるのであって逐次性を排除するのではない。逐次性は、少なくとも次の2つによって表現される。1つは、データを実行結果として出力する命令と、そのデータをオペランドとして入力する命令、すなわちデータの入出力関係によってである。もう1つは、成立の有無を判定する命令とその有無によって起動される命令、すなわち条件のガード関係（Guarded Commandより借用）によってである。計算の順序を規定する要因としては、マクロなものも考慮すると色々考えられるが、この2つが最も基本となる要因であることは明らかである。データ・フロー計算機は、データの入出力関係と条件のガード関係によって計算の逐次性を表現し、順序付けられない命令同志は並列に実行するということもできる。(2) この計算機構に基づいて許される逐次性の表現手段によって計算を記述するのであるから、従来のプログラミング言語のように、一次元の実行文の系列に、データの入出力関係や条件のガード関係を写像しなければならないプログラミング・スタイルではなく、(広義の)関数形式や論理式形式に基づく、本来必要な逐次性のみを表現すれば良いという自然なプログラミング・スタイルを取ることができるようになる。

(1),(2)に述べられている“逐次性の表現”の要請は、時間的に変化する内部状態を持つ対象物（データ）の処理は、その処理手順を、この逐次性の表現手段で記述すればよいということであり、並列度を高めるためには、不必要な副作用を極力排除するようにすべきであるということを含意する。“不必要な”副作用の除去は、単に並列度を高めるためだけではない。計算結果への名前付けや繰り返し計算など、従来は副作用を用いて行われていたものが副作用無しに表現され、その意味が論理的に簡明なものとなる。プログラム言語を論理的に透明なものとするためにも、副作用は極力排除されねばならない。

汎用の計算機構としては、簡便に実現される履歴依存性を持たねばならないことは自明である。データベースの操作の場合、交信し合う逐次プロセス(各プロセス内部では、さらにマイクロな並列処理が行われる)群として陽に並列処理を表現する場合、いずれも時間的に変化する内部状態を持つ対象物という考え方が必須である。しかし、広義のデータ・フロー計算機構にも、操作要求をデータ（メッセージ）として受け取って起動される対象物（抽象データ型、モニタを連想されたい）としてそのまま導入すれば良く、創生複製を繰り返す必要性は無い。この点は、理論的には、アクタ理論に適格に整理されている。また、データ・フロー計算機は、最もマイクロなレベルでの並列処理によって各種の並列処理を表現しようというものである。マイクロなレベルに至れば、一般的な並列処理の枠組みが取り出せるという指摘が、その功績の一つでもある。マイクロな並列処理であるから、あまり大袈裟な準備や支援がいるようでは、たちまちにその利得は帳消しになり、意味の無いことになる。trade-offを考慮してどの点に定めるかは、すなわち、複製をとることが容易ならそれで単純に並列度を高めるか、あるいはデータ構造とその操作の仕方を考慮して、できるだけ不必要な複製を作らず、特有の工夫により可能な限りの並列度を取り出すかは、プログラムする側の責任でもあ

† Another Comment on Dataflow Machines by Toshio YOKOI (Information Sciences Division, Electrotechnical Laboratory).

\*\* 電子技術総合研究所パターン情報部

る。また、ある範囲のものに対しては、構造メモリのなものも有効に働く。広義のデータ・フロー計算機機構は、従来の逐次計算をサブセットとして含むため、どのようなにも適応できるものである。データ・フロー計算機は、並列計算の問題の大半を一気に解決するものではない。むしろ、並列アルゴリズムの研究の開発を本格的に開始しうる有意義な土台をはじめて我々に提供してくれるものと考えべきである。

以上が、我々の広義のデータ・フロー計算機(機構)の定義である。“広義の”という形容からも明らかのように、アメリカで普通に理解されているデータ・フロー計算機の定義を逸脱したものである。細部はともかく、米澤氏の指摘(本誌, Vol. 21, No. 7, p. 797 の前半部)通りが普通の定義であろう。しかし、現在のところ我々は、“広義の”という形容なしに上記の定義を用いる。データ、要求、メッセージと色々なものの到着を契機に計算が駆動されるという駆動型の計算

機構の研究はさまざまあるが、ハードウェアからプログラム言語、応用に至るまで、我々が客観的に評価しうる形で最も多くの研究成果が蓄積されているのはデータ・フロー計算機の周辺にある。さらに、その未成熟な部分や欠点を克服する努力が精力的に続けられている。このような理由から、この分野の研究を一般的に指す普通名詞としてデータ・フロー計算機という言葉が、今のところ最も適切であると考えからである。

データ・フロー計算機は、言語・計算機構・アーキテクチャとあらゆる面から見て、特に今後のVLSIや新素子技術を加味すると、非常に期待の持たれる研究テーマである。しかし、いまだ未確定な部分が多い。色々な方面からの意見、批判の中で正しい方向に成長していくものと思う。その1つとして、米澤氏のコメントを良い機会として、我々の意見を披瀝させていただいた。ご批判を乞う。

(昭和55年9月8日受付)