

巻頭言

自動並列化への道程 — 計算機ソフトウェア最大の課題 —

昭和 32 年卒 名誉教授 津田 孝夫



スーパーコンピューティング（超高速大規模数値計算）は時代とともに変遷を遂げてきた。筆者が情報工学科の教授を勤めていた頃の当初は、ベクトル演算機能を備えたベクトル計算機が全盛で、自動ベクトル化コンパイラが著しい成功を収めた。線形代数（行列）をもととする数値計算には最適で、確実にスピードアップが実現できた。自動と名がつくように、ユーザは通常の Fortran プログラムを書くだけで、機械命令にベクトル演算命令が含まれる目的コードが容易に生成された。そのうちに大量生産で安価なマイクロプロセッサが出現し、ハードウェアコストを下げた多様な並列計算機が出現するようになった。モデル的には CPU とその直属のメモリを 1 単位として、それら多数をネットワーク結合したもので、いわゆる分散メモリ並列計算機である。NEC 社製の地球シミュレータでは複数個の CPU が主記憶共有のベクトルプロセッサ（1 ノードという）となっており、それらが多数ネットワーク結合されている。

問題は計算機を使う側のプログラミングである。人間の思考はもともと逐次的 (sequential) であって、順を追って 1 次元的にものを考える。いままでの高級プログラミング言語は Fortran, C をはじめ、すべて逐次記述である。一方上述のように演算をしてくれるハードウェアは分散並列である。同時に多数のプロセッサで演算実行して、百台なら百倍の速さで計算したい。このような「並列化」では、逐次プログラムが意図したのと同様の演算系列の順序依存関係が保証され、それに必要なノード上メモリのデータ参照や移動が行われないといけない。ユーザが書いた逐次プログラムからコンパイラがこれらの並列化を自動的に行い、それが実行できる命令列を生成するのが自動並列化コンパイラである。これが現在できないのである。

1994 年 MPI (Message Passing Interface) が国際標準として策定され、これが現在に到るまでも並列プログラミングの実質上の標準になっている。プログラミング言語としては Fortran または C、C++ を用い、プログラムに記述された演算系列の並列分割も、プロセッサ間のデータ参照・移動に必要なプロセッサ間通信命令の記述はこの MPI で定義された通信関数を用いて、プログラマ自身がすべて書かないといけない。これは昔のアセンブリ言語のプログラミングと同程度（あるいはそれ以上）に複雑で誤りやすく、デバッグも容易でない。

それで自動ベクトル化同様に期待されて出現したのが HPF (High Performance Fortran) である。これには筆者も多少かかわってきた。ベースとなる言語は Fortran でデータ並列言語である。各プロセッサにデータの分割配置の仕方だけをユーザは指定する。各プロセッサの計算処理の分割と必要なプロセッサ間通信・同期はコンパイラが自動的に生成してくれる。ユーザは逐次的な Fortran プログラムにコメント行の形で HPF 指示行を挿入するだけでよい。フリーで配布されている HPF 処理系もあり、Fortran と MPI をサポートするシステムならほぼ利用可能である。

HPF を使うと規則的な問題は比較的容易にシステムに見合った並列性能を実現できるが、不規則問題は並列化がむつかしく、並列性能が十分に得られないことが多い。MPI プログラミングは並列機に習熟していない一般ユーザには複雑かつ困難なものである。それで HPF に期待が寄せられたのではある

が、十分に普及していない。

以上が現状である。分散並列計算機では並列実行により flops で表される浮動小数点演算の実行速度を競う。しかし並列計算の粒度（並列実行のため分割された命令系列）はある程度大きいとプログラムの並列化がより効率的に行えることが多い。そのためにはプロセッサ・ノードのメモリサイズはある程度大きい方が都合がいいのであるが、実際世にある並列機ではそうになっていない。分散並列機の最大並列性能を出すには現状では MPI プログラミングだけであるが、これが出来るのは一部の能力に恵まれた余裕のある科学者・技術者だけであって、一般ユーザにはやはり自動並列化コンパイラの出現が強く望まれる。