

ワークステーションの特性を生かす 計算システム的设计について

戸川 隼 人

(日大・理工)

ワークステーションによる科学技術計算を、より快適に、より効率よいものにしていくための二三の提言。正味の計算時間を短縮することだけがHPCなのではなくて、データの準備、実行監視、異常事態への対応、計算の後処理などを含めたトータル・パフォーマンスを重視すべきであることを強調し、そのような視点から、インタラクティブ性の活用、標準データ形式を確立することの重要性、ウィンドウ環境の活用、データの蓄積管理などについて論じている。

DESIGN PHILOSOPHY OF SCIENTIFIC COMPUTING ENVIRONMENT CONSIDERING EFFICIENT USE OF ENGINEERING WORKSTATIONS

Hayato Togawa

(Nihon University)

This paper is a short essay discussing the way to improve
total performance of scientific computing.

はじめに

最近のワークステーションは性能が良い。東京大学大型計算機センターのかつての主力マシンを凌ぐ高速機を、個人が占有して使用できるようになった。上級機種のパフォーマンスは既に初期のスーパーコンに近い。

これを利用した大型計算(?)の事例とか利用技術などを集め、数値解析研究会で「ワークステーションによる数値計算」のシンポジウム(またはセミナー等)を開催してはどうか、と数年前から提案してきたが、今回、その第1歩ともいべき研究集会が実現したことは喜ばしい。

数値解析研究会がHPC研究会という形でリスタートしたばかり(第2回)でもあるので、今後の方向を探りながら、二三の話題を提供する。

ワークステーションの位置づけ

RISCワークステーションは、パソコンより100倍ぐらい速く、スーパーコンより100倍ぐらい遅い(どれとどれを、どのような条件で比較するかによって比率が違ってくるので、10~1000倍ぐらい、という方が正確であるが、だいたいの目安としては100倍と行ってよさそうである)。

スーパーコンより100倍遅いといってもひと昔前の大型機などと比べれば相当に速く「静的解析には十分」であり、非線形性があまり強くなければ連続体の時間依存問題にも十分使用できる。

100倍というのは大変な違いであるが、差分法や有限要素法による計算の場合には、分割をN倍細かくすると離散化した方程式の未知数の数が $N^2 \sim N^3$ 倍になり、計算時間は $N^3 \sim N^4$ 倍(またはそれ以上)になるから、スーパーコンより3~4倍粗いメッシュを使えば(それで済むような問題であれば)ワークステーションでも似たような計算ができるわけである。

現在、ワークステーションのパフォーマンスは過小評価されているように思う。多くの研究室において、高性能ワークステーションがもつばら「電子メール送受信機」兼「TEXやポストスクリプトの動くプリンタ制御装置」ないしは「UNIX練習機」として使われており、せっかくの計算性能が活用されていないのは残念である。

汎用ソフトの見直しが必要

わが家の(ということは個人のポケット・マネーで購入した)ワークステーションでも64MBの主記憶容量を持っており、パソコン(MAC II ci)にも20MBのRAMを実装させている。ワークステーションやパソコンのOSは軽いから(特にウィンドウ環境を使用しない場合には)上記の容量の大部分をユーザエリアに当てることができる。

しかし10年ぐらい前には、大型計算センターでも、一般利用者の使用できる主記憶容量は1MB程度であり、3MBを使用するには特別な手続きが必要であった。そのため当時のソフトはたいていユーザエリア1MBでも動くように作られていて、メモリ不足を補うために、たくさんのテンポラリ・ファイルを使っていた。

当時の(小さなメモリを前提とした)ソフトをそのままワークステーションに移植すると、ファイル入出力によるオーバーヘッドの多い、ローパフォーマンス・コンピューティングになってしまう。既に今日のワークステーションに合わせてチューニングされているソフトもあろうが、まだ十分に対応できていないソフトもあるようなので、見直しと改良が望まれる。

速いばかりがHPCではない

ところで、歴史的にふりかえってみると、ワークステーションの目的は、高速・大容量ではなくて「使い易い環境」を実現すること

であった。RISCチップが出現したおかげで、当初は予想もしなかったような高性能のマシンになってしまったが、これからは本来の目的である「使い易さ」をもっと追求すべきであろう。

HP（ハイ・パフォーマンス）という、CPUが動いている時間だけを短くすることのように受け取られる傾向があるが、ユーザーにとって重要なのは「問題を解くのに要する（着手から完了まで）の全時間」を短くすることであって、正味の計算時間はその一部にすぎない。

利用者にとってワークステーションがスーパーコンより「遅いけれども魅力的」なのは手元にあって

- 計算の前作業
 - データ収集
 - 点検と処置
 - 編集（変換を含む）

- 計算とのかかわり
 - 進捗状況の監視
 - 対話的利用
 - 異常事態への対応

- 計算の後作業
 - レポート作成
 - 視覚化
 - 結果の保存

などに便利だからであろう。ワークステーションのソフトは、その期待に応えるべく設計されることが望ましい。

計算の準備と後始末には、かなりの日数を要するのが普通である。その作業内容は数値計算とは全く異質のもので、必ずしも高性能のマシンでなければできない仕事でもないがコンピュータ・サイエンスの研究成果を活用すれば、かなり合理化できそうで、研究開発期間の短縮が期待できる。

前処理支援ツール

たとえば、自動計測システムで収集された

データの編集支援ツールの機能としては、次のようなものが考えられる。

- 計測データの管理
- キャリブレーション補正
- サンプリング
- 欠測データの発見と修復
- 各種のフィルタ

このようなプログラムが、現在無いわけではなく、LA（ラボラトリ・オートメーション）の一環として開発され、使用されている例も少なくない。しかし、その多くは個別の用途に応じて手作りされたものであって、中には技術レベルの低いものもある。

統計的データ解析の汎用プログラムには、上記のような機能を含む優れたものが見受けられる。しかし、できれば、特定のソフトの中だけでなく、科学技術計算用コンピュータシステムの基本ソフトの一部として組み込まれていくことが望ましい。

速やかな対応が能率を上げる

スーパーコンを使うと普通、課金されるがワークステーションはいくら使ってもタダ、という所が多いので、大規模な計算にワークステーションを使い、長時間かけて計算させることが多い。

長時間かかる計算を実行する際には、次のような機能があると便利である。

- ◆ 現在、計算がどこまで進んだか（どの段階を実行中か）を表示する。できれば更に詳しく、どこのループの何%ぐらいの所を実行していているを表示する。
- ◆ 中間結果の速報（オプションにより指定された変数の「現在の値」を、計算中、常時、画面に表示する）。プラント制御室の集中監視盤のようなもので、普通は正常に計算が進行していることを確認するだけであるが、異常が起きたときにはそれを速やかに発見し処理できるので、計算が徒労になるのを避けることができる。

◆ 「少しおかしい」と感じたとき、計算を中断して（もし異常がなければ計算を再開できるように、それまでに計算された結果を保持したまま）各変数の値を調べることができるようにする。

◆ 計算の節目の所で中間結果をダンプしておいて、停電などの不慮の事態に備える。これは暴走した場合の原因究明の手掛かりとしても役立つ。

◆ キーボード操作、マウス操作の全記録を残し、異常な結果が出たときの検討材料にする。

このような機能を個々のプログラムに組み込むことは既に広く行われており、そのためのプログラミング・テクニック（たとえばブレーク・ポイント・ダンプというようなもの）も知られている。しかし、できれば、これを（いちいち場当たりに考えてコーディングしてデバッグして……というのではなしに）自動的に（または簡単な指示だけで）組み込めるようにOSやコンパイラが支援してあげることが望ましい。

たとえば、進捗状況の自動表示には、次のような方法が考えられる（FORTRANの用語で説明するが、他の言語でも全く同様な処理が可能）。

- 1) プログラムの中のDO文を探す
- 2) そのループの実行時間を推定する
- 3) すぐ終わるものは放置。長時間かかりそうな場合は、DO文の次に、

$$P = (\text{制御変数の値}) / (\text{制御変数の最終値} - \text{出発値})$$

の値を実数型で計算して表示するためのWRITE文を挿入するか、バー表示のプログラムをCALLする。

そのループの計算内容を「プログラムの文面から判断して自動的に表示する」ことは一般には不可能であるが、プログラムにきちんとコメントが付けていると仮定すれば、DO文の直前の注釈行の内容を表示することにより

目的がほぼ達成されるのではないかと思う。

入力ミスを減らせば能率が上がる

長時間かけて計算したのに、入力時の操作ミスのためにムダになってしまうことはよく起こる。これを避けるためには、入力ミスを起こしにくい入力方式を工夫することが望ましい。入力ミスの原因としては、

- ① 単純な操作ミス（打ち間違え）
- ② 書式（順序）の誤り
- ③ 単位の取り違え

などが考えられる。

◆ 対話形式で入力する場合

親切なプロンプトを用意しておけば、上記②③に対処できる。しかし対話形式による入力は、数値を全部キーボードで入れなければならないため、オペレータの負担が大きく、上記①型のミスを起こし易い。その対策としては、他のウィンドウに表示されている数値をコピーしてきて「キーボード入力と見なして扱う」ことを許し、更に、入力待ちの状態からエスケープして、他のウィンドウで計算や検索を行なうことができるとよい。

◆ ファイル形式の場合

指定された書式のファイルを、汎用のエディタで作成、編集して読み込ませる方式は、②③のタイプのエラーを起こしやすい。これを避けるには「記入用紙」を模倣したファイルを用意しておいて、その所定欄に数値を「記入」させる方式にするとよい。

(例)

半径 (cm) :

質量 (kg) :

節点の個数 :

.....

このようなファイルをエディタの画面に読み込み、:印の右に数値を書き込む。アプリケーション・プログラムの方では

：印までを読みとばし、それより右の値だけを入力データとして扱う。利用者のイタズラで、エディタによる行の移動、入れ替え、削除、などを行なわれることが予想される場合には、タグ（上の例の「半径」など）を照合して、正確に対応させていけばよい。

◆ もっと理想的な方式

コンピュータが「与えられた資料」の中から必要なデータを自動的に探し出してくれると便利である。将来、いろいろなハンドブックが CD-ROMの形式で供給されるようになると、これが単なる夢ではなく、現実的な手段になるであろう。ネットワークにアクセスして必要なデータを集めてきてくれれば更に便利である。そのような利用形態を考慮して、CD-ROMやネットワーク上のデータベースには「コンピュータが読むためのタグ」を付けておくことを義務づけるべきであり、その標準化を早急に推進することが望まれる。

なお、必要なデータが見つからなかった場合の処置とか、文献によって値が少しずつ違っている場合にどれを採用すべきか、というような問題に関しては、利用者との対話によって解決していけばよい（利用者がいつでもマシンのそばに居る、という特質を生かす）。

時代のキーワードはOPEN

何から何まで一つのシステムの枠組みの中で行なう方式をCLOSEDといい、汎用的なモジュールの自由な組み合わせで対応していく方式をOPENという。これからの時代のキーワードはOPENである。

パソコンの業界では、DOS-V マシンのWINDOWSを軸に、急速にOPEN化が進行している。ワークステーションの業界ではSUNとDECとIBMの囲い込み戦略が

交錯して、いささか中途半端なOPENになっているが、UNIXとイーサネットによる共通の基盤が最初からある。

ワークステーションとパソコンの間、および、ワークステーションとスーパーコンの間は、まだあまりしっくりしていない。将来もっと円滑な協調システムの実現することが望まれる。

アプリケーション・ソフトの世界においてもOPEN化は重要な課題である。たとえば

◆ データ形式の標準化、および、代表的な既存のデータ形式の間の自動変換。これにより、いくものソフトをプラグイン的に組み合わせて利用することが可能になる。

◆ プログラムのコンポ化。たとえばプリプロセッサの部分だけを切り離して利用するとか、フィルタだけは利用者自作のものを組み込んで計算することなどを、自由にできるようにする。

このような動きは既にかかなり進んでいるが、更に広範囲に、徹底してOPEN化が推進されることが望まれる。

いくつかのお手本

これからの方向を示唆する、先進的なシステムの二三の例を紹介する。

◆ MATHEMATICA 数式処理と数値計算（それもブラックボックス的な処理と手続き的な処理の両方）を自由に混ぜて使用できる、という意味で注目すべきシステム。

◆ AVS ももとはグラフィック表示のプログラムであるが、ソフトウェアのOPEN環境のプラットフォームとしても注目すべきシステムである。

◆ VisualBASIC オブジェクト指向とGUIに関する強力な機能を持ったBASIC。先に論じた「快適な実行環境」の実現手段として注目される。