

されど並列計算

逐次処理と並列処理をめぐるハード屋の気持ち、ソフト屋の気持ち

この数年、コンピュータのプロセッサチップは、計算コアを複数内包するようになった。クロック周波数を上げてプロセッサを高速化するのは難しくなったので、複数のコアを集めてトータルで性能をだそうとしているのだ。

複数コアの性能を引き出すためには、プログラムをそれぞれのコアで動作させ、うまく連携させていくことになる。そしてその考え方は、たくさんのコンピュータをネットワークでつなぎあわせた分散・並列システムへとつながっていく。

このような並列計算の概念は、古くからあったが、プログラムの並列化は一部の分野に限られていた。面倒な並列処理を書かなくても、プロセッサの性能成長が速かったからだ。とはいえ、クロック周波数は伸び悩み、いまはコンピュータをたくさん集めるほか、妙案はないように見える。

そこで、立ち位置の異なる3人の研究者に尋ねた。「並列」って、どう？

**ハード屋さんの言い分：
本当は並列に動いている計算機**

現在のコンピュータの根底にある動作原理は、チューリングマシンである。並んでいく命令列を順に計算していくというモデルは、まさに逐次処理そのものだ。

ところが実は、ハードウェアの中身は並列に動作している。プロセッサ内では数億のトランジスタが独立に動き、クロックですべて同期しているように見せかけられている。メモリには複数コアが勝手に読み書きを要求するが、データの整合性は保たれる。このような多数の技術マジックで、ハードウェアはやっとなら逐次処理が可能な1つのシステムに見せかけられているのだ。ハード屋はまさに、このことのために額に汗している。

さらに、数万のプロセッサをつなぎあわせた場合はどうだろうか。隣のプロセッサにデータが届くまでの時間は1ナノ秒、いちばん遠いプロセッサは、「京」スパコンの場合で10マイクロ秒。両者には1万倍の時間差がある。近くは速く、遠くは遅い。その時間もあいまいだ。

それが自然の姿なら、遠くを操作するときは1万倍の時間、近くを操作するときは1万分の1でも動くというアルゴリズムがあるべきではないか。その自然形をどう使うかがわからないから、現在はプログラムの因果関係を保たないと計算ができない。

ハードウェアは並列動作が「天然」。そして、自然の現象も世の中の現象も、たいていは並列に起きて、並列に推移していく。それをどうやって並列に解くかが、問われている

のではないだろうか。

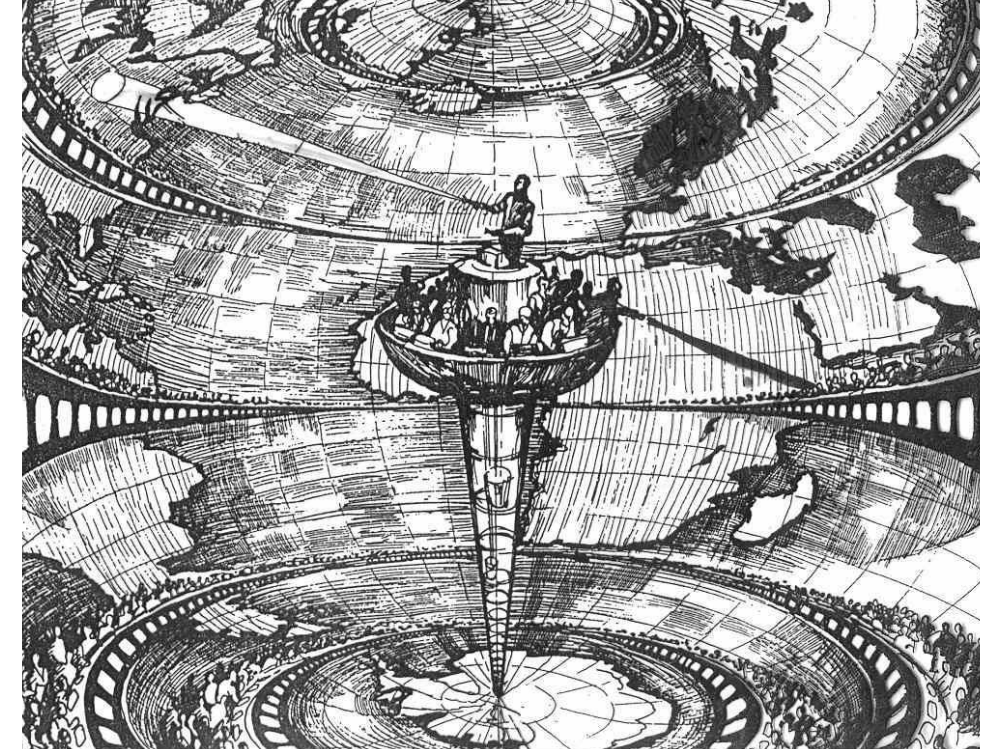
**実際に動かすOS屋さんの言い分：
データの移動、それが問題だ**

とはいえ、プログラムの動作は逐次処理が基本。並列計算のプログラム開発はやっぱり難しい。プロセッサとネットワークの性能差は大きいので、プログラム間の通信に時間がかかってしまうつくりでは、並列化の意味がない。しかしそれでも、性能を求めるためには並列化せざるを得ない。

基盤ソフトウェアの立場で重視するのは、結局のところデータの移動——プログラム間の通信とファイルシステムをどうするか。計算にはデータが必要だからだ。

ファイルシステムについていえば、数値計算は、テラバイト、ペタバイト級のファイルをはき出す。一方、最近ビッグデータといわれる分野では、Webやセンサーから集めた何万、何億もの小さなファイルを扱う。それをいかにして格納し、プロセッサに供給するか。

コンピュータは、速さも容量も大きく異なる部品をつなぎあわせてできている。何万ものディスクの中では、円盤が非同期に回転し、データがばらばらに記録され、それは次の瞬間に故障で消えるかもしれない。その不均衡と不安定の上で、データを1つの



「リチャードソンの夢」: Illustrated by Alf Lannerbäck



20世紀前半に活躍した英国の数学・気象学者、ルイス・フライ・リチャードソンは、いかにして数値計算による気象予測を実現できるかを考えていた。著書「Weather Prediction by Numerical Process」のなかでは、「予測工房」の構想を著している。巨大な劇場に6万4000人の計算手を集め、地球の表面を細分化して気象の変化を手分けして計算するというものだ。1922年、まだコンピュータが登場するまでにしばらく待たねばならないこのことである。この構想は実際には実現しなかったが、その後のコンピュータを使った数値計算による天気予測の原型といえる。(イラストは、1984年9月22日発行のスウェーデンのDagens Nyheter紙に掲載された)

安定したファイルに見せかけ、透過的に移動させるのは、とほうもないことである。チャレンジの甲斐がある難問だ。

**アルゴリズム屋さんの言い分：
速ければすべてよし**

数値計算は、並列計算でもっとも成功してきた分野だ。世の中の現象の多くは並列に推移するので、それらを並列計算で扱うのは自然なことである。しかし応用の立場では、実際に速くなければ意味がない。

たとえば気象シミュレーションでは、現在のある状態と過去の観測情報から、3日先、1年先を予測する。プログラムは時間に沿っ

て逐次に現象を計算する必要があるが、時間ステップで進む物理現象を時間方向に並列化して計算することは、現在のところうまく行っていない。

一方探索問題は、違う作戦で複数の経路を並列にたどることによって、高速化の望みがある。自然言語理解の分野でも、何万とある単語の組み合わせを評価するために、並列化が効果を上げている。

並列計算を実用的なものにするカギは、いかに並列性のあるアルゴリズムを編み出すか。そこはまさに、アルゴリズム分野の腕のふるいどころだ。そのうえで、並列計算の妨げとなるプログラム間のデータのやりとり

を、量、頻度ともに絞っていく。

最近よく使用されているGPU(汎用に使われる画像処理プロセッサ)や、SIMD(1回の命令で複数データを同時処理できるプロセッサの高度な機構)は、逐次処理の呪縛から一歩踏み出したハードウェアだといえる。逐次処理をあつさり捨ててアルゴリズムをがんばると、これまでできなかったことに手が届くようになるだろう。

未来への夢からいままある技術まで、思いおこいの並列話。平木 敬教授(ハード屋さん)、石川 裕教授(OS屋さん)、須田 礼仁教授(アルゴリズム屋さん)に語っていただいた。