第６回　コンピュータ科学基礎論　資料　　2017-10-27 山内

今回は、アルゴリズムと並列について、議論したい。
今式のコンピュータの誕生以来、大量かつ多種多様なアルゴリズムが開発され、プログラムとして利用されてきた。しかし、その前提となるのはコンピュータの元々のモデル、つまり命令の逐次実行であったし、「アルゴリズム」がそもそも、「やりたいこと・解きたい問題を逐次実行でどうやって解くのか、その計算方法」であった。 MIMD 型の並列処理を考えるとき、今まで考えられてきたものではうまくゆかない、もしくは足りないことがあるに違いない。
どう足りないのか、何が足りないのかを踏まえて、MIMD 並列を前提とする時代のアルゴリズムの姿を考えてほしい。

(1) 足し算 Σ a[i] の計算法を考え直してみて欲しい。
　a) 面白い工夫を考えつくか？
　b) 仮に、今まで逐次実行で考えていた変数Sに加えてゆくやり方を、スライドでも紹介した2つずつ足していくやり方に「変換」できるだろうか？ その時、足し算がどういう条件を満たすからできるのだろうか？ つまり、自動変換をしたいと考えるときに、どういう条件を満たす演算子（関数）に対してこのやり方ができるのだろうか？

(2) ソートの工夫を考えてみて欲しい。
　ビデオではバブルソートの代わりになる方法や再帰的なクイックソートの代わりになる方法を考えてみた。 考えてみるとこれらはいずれも逐次実行でなるべく早くソートできるアルゴリズムを元にして、それを並列化することを考えていた。 その考え方を止めるとして、最初から並列実行を前提に考えるとすると、どういうことが考えられるか？ そもそもどう考えたらいいだろうか？

(3) トリーのトラバーサル（走査）による探索の工夫を考えてみて欲しい。
　トリーをトラバースして探索するのに、深さ優先探索とか、幅優先探索とかがあることは、学習済みであろう。 仮に、条件を満たすノードを（複数あったとしても）最初に検出したものを結果とするものとする。（1つだけ見つければよい）
　① 最初に検出するまでのステップ数（通過ノード数）で、探索法を評価するとする。 この評価法は妥当だろうか？
　② 妥当だとして、深さ優先法や幅優先法では、どのような評価になるだろうか？ 木の形（バランス木か否か）、や、順序性（ノードデータが順序通りに配置されているか否か）に依存するだろう。
　③ では、並列処理によって探索するとしたらどうするか？ その時の評価はどうなるだろうか？
　並列アルゴリズムの1つの可能性として、「無駄を許す」ということが考えられる。 何か面白いアイデアが出ないか、考えてみて欲しい。

(4) 山登り法による最適化と遺伝的アルゴリズムの問題を考えてみたい。
　関数f(x)があるときに、それの最大値（最適化問題の場合は最適値）を山登り法で求める。 初期値のxを決めて、そこでの傾きを求め、傾きが登る方向へxを少し増やす。 これを繰り返すことで、最大値に到達するはず、という原理である。 極大値につかまってしまうので、一旦極大に到達したら別の初期値でやり直す、ということを何回か繰り返すことが必要になる （遺伝的アルゴリズム、genetic algorithm）。
　並列アルゴリズムの1つの可能性として、「無駄を許す」ということが考えられる。 何か面白いアイデアが出ないか、考えてみて欲しい。