勉強会参考資料
-アプリケーション、アルゴリズム-

田中 利彦

平成8年 5月 1日

1 並列と並行

並列とよく似た言葉に「並行」がある. 両者を同義で扱うこともあるが, ここでは区別する.

大まかにいうと, 両者がお互いに並行であるというのは, 論理的に同時に処理が出来るということである. 並列というのは物理的な資源が多数あるということでよって, 並列処理を考える上でソフトの並行性とハードの並列性がかみ合うことが重要である. というように用いる.

1.1 粒度

つぎに, 各並列処理方式で並列処理の対象とする基本単位 (タスク) の大きさ (grain size : 粒度とよぶ) で分類する. まず, 粒度の最も細かい命令間の並列生を利用するものとしては, スーパースカラ,VLIW, データフローマシン, プロセッサアレイがあり, 命令パイプライン, 演算パイプラインも若干並列性の使い方は異なるが, このグループに入れることが出来る.

このように 1 命令単位の並列処理を行う方式を細粒度並列処理とよぶ. またそれより少し粒度の粗い, 複数命令を並列処理単位とした方式としてはシストリックアレイが挙げられる. なお現在では, この程度の粒度を扱えるマルチプロセッサシステムが現れ始めているとともに, 細粒度タスク並列処理のためのオーバーヘッドを軽減するために, このような粗い粒度を取り扱えるようにしたマクロデータフローマシンも提案されている.

さらにタスク粒度の粗い Do ループの繰り返し間の並列性の利用は, マルチプロセッサで従来より行われている. この粒度の並列処理を中粒度並列処理方式とよぶ.

さらにタスク粒度が粗い Do ループ, サブルーチン, 基本ブロック (複数の代入分からなる部分) 間の並列性の利用もマルチプロセッサシステムでは行うことが出来る. この粒度の並列処理を粗粒度並列処理方式とよぶ.

1.2 駆動方式

計算機アーキテクチャーは, 機械命令の起動形態で分類すると, 制御駆動, データ駆動, 要求駆動に大分類できる.

1.2.1 制御駆動方式

制御駆動方式では機械命令プログラムの実行が, 一つの制御装置によって制御される. プログラム・カウンタで指された機械命令が主記憶装置から読み出され (命令フェッチ), 命令のデコード, オペランドフェッチ, 実行, ストアのように, 処理時間がほぼ等しい複数のステージ (別々に動作する専用のハードウェア) に分割される, いわゆるノイマン型計算機であり機械命令の実行が逐次的になされる.

しかし並列処理を考えるとデータ駆動の考え方が必要になり両者を組み合わせて計算機を構成しなければならない

1.2.2 データ駆動方式

データフローマシンがそうである.

データ駆動方式にはプログラム・カウンタがなく, 各機械命令はそのオペランド・データがそろうと実行される. したがって, 原理的には, プログラムに内在する全ての並列性を実行時に同時に引き出すことが出来る. プログラムはデータの依存関係を示すデータ・フロー・グラフで表現される. ベクトル化が容易でないような科学技術計算分野や人工知能用言語 PROLOG の処理系などの非定型的処理の高速処理が期待されている. しかし, 現状ではメモリの構成などを中心に解決すべき問題点が山積みしている.

1.2.3　要求駆動方式

要求駆動方式では, 機械命令はその結果が要求されるとき, 起動される. 要求された命令が出した結果は要求した命令に換えされる. 社会モデルでいえば, 階層型の会社組織や受注生産システムに対応する.

例えば, 会社組織では, 課長が処理内容を見て係長に要求を出し, 係長はさらに平社員に要求を出す. 平社員の処理結果は係長に返される. 係長は結果を更に整理して課長に返す. ここで, 処理の段取りがあらかじめ定まっているような定型業務は, 最初から平社員に渡されるようになっていることに注意する. 定型的な業務では要求を伝える過程はオーバーヘッドとなるからである. 結果を返す過程はデータ駆動的になされる. 要求駆動方式では, 実行過程が処理してみないと分からない動的なもので, あらかじめデータ・フロー・グラフにコンパイルできない場合, 要求を伝える過程で解釈実行しながらデータ・フロー・グラフを生成する. 従って, データ駆動方式がコンパイラ向きなのに対して, 要求駆動方式はインタプリタ向きなものとなっている. 要求駆動方式は関数型言語の項書き換えモデルや論理型言語 PROLOG の実行制御過程との親和性がよいとされている.

1.3　時間並列と空間並列

並列処理をする場合, 空間方向の並列化すなわち空間分割型並列と, 時間方向の並列化すなわち時分割型並列が考えられる.

空間並列の方は簡単で, 多数の処理装置を空いている場所に置くことである.

時間並列は, 一命令の実行過程をパイプラインと呼ぶ複数ステージに分け, ステージごとに独立実行できるハードウェアを装備し, これらの複数ステージをオーバーラップして稼働させるパイプライン処理方式のことである.

実世界に照らし合わせてみると, 時間並列が, 自動車などの生産ラインのことであり, 空間並列が生産ラインの増加にあたる.

1.4　通信モデル

離れたある両者間でデータのやり取りを行う場合, 二つの方法が考えられる. 一つが両者に共有のメモリにアクセスすることによりほしいデータをもらう方法. もう一つは手紙のように相手がほしいデータをメッセージにして送る方法.

1.4.1　共有メモリモデル

黒板みたいなものといえばいかにも便利そうだが, 実際は共有メモリに今何が書かれているかというのは外部からはわからない. それゆえ, 両者の間で同期をとる必要がある. すなわち制御がどこまで進んでいるか分かっている必要がある. よってこの方法はコントロール駆動の考え方であると言える.

1.4.2　メッセージパシングモデル

この方法はメッセージとともに, ほしいデータが必ず来ることが分かっており, そのデータにより先の処理を進めていくことができる. これは, データ駆動の考え方である.

2 データ並列とコントロール並列

データ並列とは,データが多数あるときデータの各要素に対してにたような処理をするときデータ並列という.

一方コントロール並列というのは, 例えばスーパースカラの演算部のように非均質な処理をするときをいう.

3 アムダールの法則

単一プロセッサで最良の直列アルゴリズムによる実行時間を T_1,N個のプロセッサで並列アルゴリズムによる実行時間を T_Nとすると,$S_N = T_1/T_N$で定義される速度向上比は,直列アルゴリズムの中でどうしても並列化できない割合をα とすれば

$$S_N = \frac{N}{1+(N-1)\alpha}$$

というようにプロセッサの関数で表されるで与えられる. これがアムダール (E.Amdahl) の法則である.

参考文献

[1] 富田 眞治. 並列計算機構成論. 昭晃堂

[2] Kevin Dowd. High Performance Computing. トムソン ・パブリッシング

[3] 笠原 博徳. 並列処理技術. コロナ社

[4] 情報科学事典

[5] 宮村 勲. 並列処理言語. マグロウヒル

[6] 奥村 晴彦. LaTeX 美文書作成入門. 技術評論社, 1991.