

プログラム設計における並列処理概念の導入

三浦 康雄

日本ユニシス株式会社

、並列処理でのプログラム設計の考え方を紹介する。並列処理型の設計では、処理の機能細分化を図り、その細分化された機能の中から並列化できる単位に分類し、モジュール化する。これにより並列処理を行えるだけでなく構造化されたプログラムを作成することができる。このことは保守性、拡張性に富む。事例では以上の考え方を適用したレンダリング・アプリケーション・プログラムを紹介する。

1. はじめに

従来のプログラム設計では、シングルプロセッサを対象としていたため、同一アルゴリズムで処理できるものは、繰り返しなどの手法を使って逐次処理を行っている。

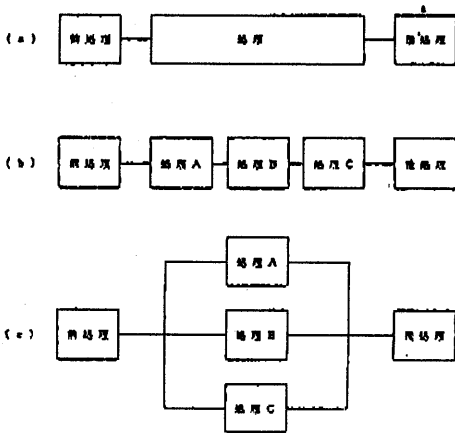
並列処理計算機の導入によって計算機の特徴をいかしたプログラム設計が必要となった。

本稿では、並列処理の考え方を取り入れたアプリケーション・プログラム（以下APと呼ぶ）の設計方法と、事例を紹介する。

2. 並列処理向きのAP設計の考え方

従来の逐次処理でのプログラム設計では、処理が機能的に細分化されていないものが多かった。あるいは、構造化プログラミングなどの手法を用いたものでも並列処理を意識したものは少ない。（図-1（a）（b））

並列処理でのプログラム設計では処理の細分化が非常に重要である。これは効率よくプロセッサに仕事を割り当てるためである。（負荷分散）（図-1（c））



(図-1)

並列処理での設計手順をまとめると以下の通りである。

- (1) 処理の機能別細分化
- (2) 並列化できる単位に分類。
- (3) (2) でまとめた単位にモジュール化（サブルーチン化）

また以下のことに注意して設計を行わなければならない。

- (1) 外部変数によるプログラム間のデータ渡しを避ける。
- (2) 共有変数によるプログラム間のデータ渡しを避ける。

3. 事例

曲面形状をレンダリングするアプリケーション・プログラムに並列処理設計の考え方を適用した。レンダリング技法は処理がピクセル単位に独立しさらに、処理がピクセル単位に同一であるという特徴を持っている。これらのことは並列処理を行うとき並列化しやすいという利点がある。

たとえばピクセルを構成するRGBデータの処理などは各データごとに独立して処理を行っている。

4. おわりに

以上の考え方を取り入れることにより並列処理での負荷分散の問題に対応することができた。また保守性に富み、拡張性の高いAPが実現できた。

また以上の考え方は、シングルプロセッサにも有効である。

参考文献

- 1) 瀧 和男：大規模汎用並列処理計算機の実現に向けて bit Vol. 23, No 2
- 2) Meiko社：CSToolsマニュアル (1990)