

# 画像処理における階層的な並列分散処理システムについて\*

1 K-8

天野 圭      稲吉 康史      宮村 勲†

新潟大学 工学部‡

## 1 まえがき

近年ワークステーションの発達に伴い、イーサネットを始めとする高速通信路で相互のワークステーション群をつなぐことにより、サーバ・クライアントモデルの分散処理環境が整いつつある。このような環境を画像処理に用いる際、画像処理の計算には並列性を持つものが多く、これらの処理を複数のワークステーション上に分散させることにより、処理速度の向上が期待される。我々は単純なサーバクライアント形式の分散形態ではなく、非同期的な協調作業を行うことが出来る階層型の並列分散処理のモデルとその構築方法を提案し、画像処理への応用について述べる。

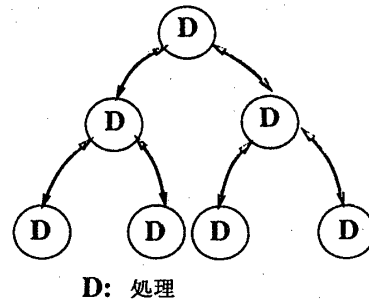


図 1: 階層的な分散処理形態

## 2 階層的な並列分散処理について

プロセス指向の分散処理はさまざまな形態が考えられる。分散された処理の一括管理をおこなうスター型の分散形態、またネットワークのトポロジーが有機的な形態などが考えられる。前者の場合、各分散処理の相互関係度、いわゆる協調度が低い場合は問題ない。しかし協調処理を必要とする場合、相互の分散処理の管理機構の問題、また物理的にはメッセージ通信のトラフィックの問題が起こる。後者は分散の形態が複雑になるので、アプリケーションを組む際問題が起こる。そこで、人間にとって分散処理の記述が容易な階層的な分散処理を行う形態をとる [1]。図 1 に階層的な分散処理の形態を示す。処理のコミュニケーションは非同期メッセージ通信形態 [2] で行なわれ、メッセージ通信の範囲は自分の親と子供に限定する。

でデッドロックが起こる危険性があり、単純に行うことは困難である。そこで、分散処理の単位を UOD (a Unit Of Distribution) と定義する。UOD 内には次の 3 つのオブジェクトより構成される。

- System management Object(SO): UOD 内、UOD 外とのメッセージ管理を行うオブジェクト
- Processing Object(PO): 計算処理を行うオブジェクト
- User interaction Object(UO): ユーザとのやりとりを行うオブジェクト

## 3 分散処理の具体的モデル

プロセス指向の分散処理を行う場合、各プロセスを異なるマシン上で起動し、基本的な通信ライブラリを用いてメッセージ通信を行う方法がある。しかし協調作業を行うような処理機構である場合、同期処理など

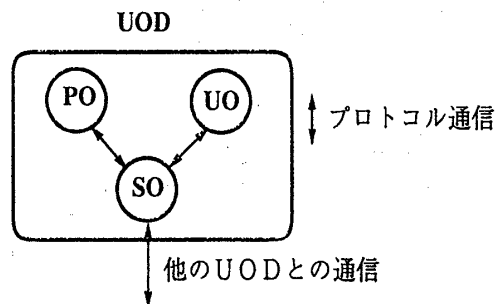


図 2: UOD の構成

図 2 に UOD の構成を示す。SO は本分散システムより供給されるオブジェクトであり、UOD 内の UO と PO 間、あるいは外部の UOD から送信されたメッセージを

\*Hierarchically distributed parallel processing for image processing

†Kei AMANO, Yasushi INAYOSHI, Isao MIYAMURA

‡Niigata Univ.

スプールする。UO は UOD のユーザインターフェイス部分を受け持つオブジェクトであり、対話的にメッセージを送信することが出来る。PO は UOD 内の演算処理を受け持つオブジェクトである。

ユーザが発する UO や PO からのメッセージは、各オブジェクト内部で通信プロトコルに変換される。オブジェクト間はプロトコルに従って通信が行われ、これを OCP(Object Control Protocol)と定義する。同様にプロトコルからメッセージへの逆変換も行われる。これらプロトコル変換はシステムとして供給される。プロトコルのインプリメントは BSD UNIX の通信ライブラリである socket を用いる。

PO や UO は内部に状態遷移テーブルを持ち、ある状態において受理するメッセージが決まっている。PO や UO がメッセージ受理を行なうには、メッセージがスプールされている SO に対してこれらのメッセージの送信要求を行う。要求されたメッセージが SO 内であれば送ってもらい、メッセージ受理を行なう。そして受理したメッセージに従って処理を行なう。状態遷移テーブルの例を図 3 に示す。これは状態 A の時は M1 と M2 と M3 が受理可能なメッセージで、M1 を受理すると状態 B へ遷移し、M2 を受理すれば状態 D、M3 を受理すれば状態 A に遷移することを示している。状態 A は M1, M2, M3 いずれかを受理したら状態遷移を起こすので OR 型の状態と定義する。また状態 B は M4 と M5 を両方受理しないと状態 C に遷移しないので、これを AND 型の状態と定義する。

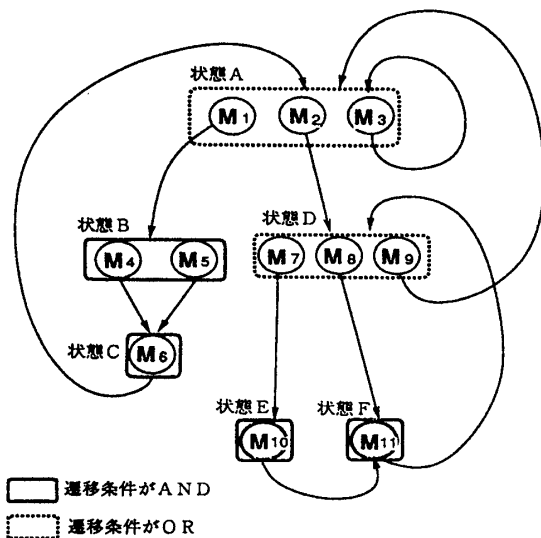


図 3: 状態遷移テーブル

## 4 分散指向の画像処理システム

これを彩画処理システム [3] に適応する。これは領域指向の彩画システムで、輪郭線入力システム、色の入力システム、彩色を行なうシステムで構成されており、ユーザが輪郭線を入力し、彩画に関する色情報を指定すると画像が生成されるものであるが、ここで各領域の彩画処理は並列処理可能となっている。つまり一つの領域に関する輪郭線と色の入力が終われば、この領域に関する演算を PO や子供の PO に依頼することができ、複数の領域の彩色演算をまとめて並列に行なうこともできる。ユーザは主に画像作成の際 GUI を用いて対話的に作業を行なう。この GUI の部分は UO が役割を果たし、また UO だけで即座に処理が行えないような計算部分は PO に処理を依頼する。また輪郭線入力、色入力、彩色演算処理をそれぞれ UOD として考えることにより、分散指向のアプリケーションが実現できる。ある領域の彩画の際には輪郭線入力の UOD が生成され、輪郭線入力を行なう。輪郭線に色付けを行なう際には色入力の UOD が生成される。色入力が終わったら彩色演算の UOD が生成される。このように UOD の動的生成を行なうことができる。

この分散システムが基本的に非同期形式であるので、各 UOD が独立で行なうことができる処理は各自で行ない、処理結果を必要とされる UOD に送信すればよい。送信された結果は各自の UOD 内の SO にスプールされるので、必要なリソースがあれば SO にメッセージ受信要求を行なうことにより得ることができる。

## 5 むすび

本稿で階層的な並列分散処理について述べ、この分散システムのワークステーション上での構築方法とこのシステム概要について述べた。また分散指向の画像処理システムの実現手法について述べた。課題としては本システムを用いたアプリケーションとして本格的な画像処理システムの構築があげられる。

## 参考文献

- [1] 天野 圭, 宮村 勲: "ネットワークにおける階層的な並列分散処理について", 電子情報通信学会信越支部大会, No182, p363-364(1992)
- [2] 宮村 勲: "並列処理言語", マグロウヒル, pp55-64(1986)
- [3] 天野 圭, 宮村 勲: "自然画彩色システムの開発について", 電子情報通信学会信越支部大会, No122(1991)