6D-8

Ｃｏｐｒｏｓｓにおける並行プログラミング

真野 芳久　大石 東作　高山 文雄
電子技術総合研究所 ソフトウェア部

1．はじめに

我々は、多数のプロセッサからなる整構造並列アーキテクチャ上で実行される高度に並列化された科学技術計算用プログラム作成のために、データフロー図による並列プログラム設計支援から[1]、シミュレータによるデバッグ支援[2]、並列計算機による並列実行に至るまでの総合的な支援システムの開発を目指している。本稿ではシミュレータによるプログラミング支援機能について述べる。そこではプロセス間のメッセージ交換により直接プログラム中に並列性を記述する方式を採用し[3]、多数のプロセスが同時実行するプログラムの開発支援のために、すぐれたユーザインタフェースを持つ実行時支援機能を持たせている。

2．Copross システム概要

ここで想定している並列計算機は、放送機能を持つ整構造のプロセッサアレイである。即ち、各プロセッサは隣接するプロセッサと通信用の結合を一様な構成方法で持つ整構造であり、更に、すべてのプロセッサは共通のバスで結ばれている。図１に、放送機能付き２次元整構造プロセッサアレイの構成例を示す。

Copross は、このアーキテクチャを反映したメッセージ交換に基づく Copross/L 言語の仮想並列計算機コードへのコンパイラ、仮想コードのシミュレータを中心とした支援システムである。Copross/L は、CSP[4] をベースとし、放送機能、配列演算機能などを持つ言語である。シミュレータでは、多数のプロセスの並列実行の様子を的確に表現でき、中断時などにおいて種々の情報を比較対比できるすぐれたユーザインタフェースを実現するために、視覚的に表現された情報を持つ種々のウィンドウを活用して動作する。図２に Copross および関連システムの構成を示す。

3．並行プログラミング支援

多数のプロセスの複雑な実行状態を的確に把握できる環境とするために、Copross は、会話型で、視覚的表現を重視し必要な情報を独立かつ選択的に提示することで理解性を高め、柔軟な実行モードを持つこ

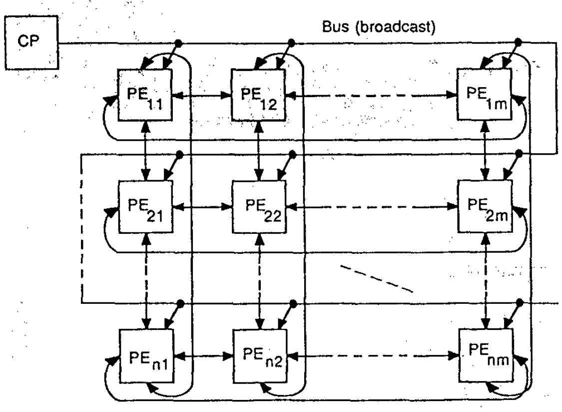

図１．放送機能を持つ２次元整構造プロセッサアレイの例

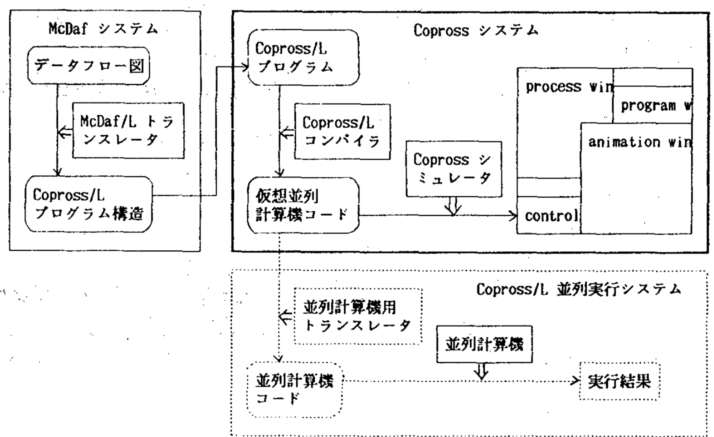

図２．Copross および関連システム構成図

Concurrent Programming on Copross System
Yoshihisa MANO, Tosaku OISHI and Fumio TAKAYAMA
Electrotechnical Laboratory

とで並列計算機上の実行を反映できるようにしている。

Copross における支援機能は大きく中断時の状態観察機能と実行時の動的観察機能に分けられる。プログラム実行時には、利用者は全体のプログラム及び特定のプロセスの動作を理解の容易な形態で画面上に動的に観察でき、また、その観察に基づいて希望の時点でプログラムを中断させることができる。シミュレータに柔軟な実行モードを設けることで、この動的観察をさらに支援する。プログラム実行時や中断時には、利用者からの要求に応じて、必要とされる情報が独立に提示される。この情報提示は、視覚的に表現できるものはその長所を生かしつつ、また、不必要に多くの情報を提示するのではなく利用者に必要なものを選択的に提示する方法でなされる。

これらの機能は、それぞれの役割を持つ種々のウィンドウに割り当てられている。ウィンドウには、制御ウィンドウ（プログラム実行・中断、モード設定の制御）、実行モード設定ウィンドウ（シミュレーションスピード、中断条件、各種トレースフラッグなどの実行モード設定）、動画ウィンドウ（プログラム実行のプロセス単位での動的描写）、プロセスウィンドウ（プロセス毎の状態・通信履歴の表示）、プロセス群ウィンドウ（同じプロセス定義から生成されたプロセス群の表形式の状態表示）、プロセス定義ウィンドウ（プロセス定義のソーステキスト及び実行地点等の情報の表示）、ソースプログラムウィンドウ（ソースプログラムテキストの表示。エディタの対象となる）などがある。

図3にシミュレーション実行中における画面表示例を示す。利用者は、動画ウィンドウによりプログラム全体の動作を観察しつつ、また、プロセスウィンドウやプロセス定義ウィンドウに表示される変数の現在値や実行箇所等により個々のプロセスの動作を観察しつつ、必要に応じて任意の時点で実行を中断させる。中断時には、任意のプロセスの状態を種々のウィンドウを利用して調べることができる。これらの表示を参考にして、ソースプログラムウィンドウ中のプログラムテキストを編集して誤りを正す。また、注目すべきプロセスのみの実行、あるいは中断条件やトレース変数の指定などの実行モードの変更を行なう。こうして、プロセスの集合である Coprоss/L プログラムの開発は、個々のプロセスに対する検討、プログラム全体の動作に対する検討を、自由に組み合わせることで、スムーズに進めることができる。

4．おわりに

今後は当所で開発された並列計算機上で実行するために、プロセスのプロセッサへの割当てなどの必要な処理を行なうトランスレータ、並列実行時環境の開発を行ない、McDaf システムも含め、総合的な並列プログラミング環境の構築へと進む予定である。

最後に、本研究の機会を与えられた柏木電子計算機部長、日頃御指導頂く棟上ソフトウェア部長、二木言語処理研究室長、藤村情報システム研究室長、植村プログラム研究室長に感謝する。

参考文献
[1]大石、等：並列計算用データフロー図の並行処理言語への変換、32回大会 3G-3。[2]真野、等：メッセージ交換に基づく並行プログラミングのためのユーザインタフェース、32回大会 3G-2。[3]高山、等：メッセージ交換に基づく１つの並行プログラミング言語、32回大会 3G-1。[4] C.A.R.Hoare：Communicating sequential processes, CACM 21,8 (Aug. 1978)。

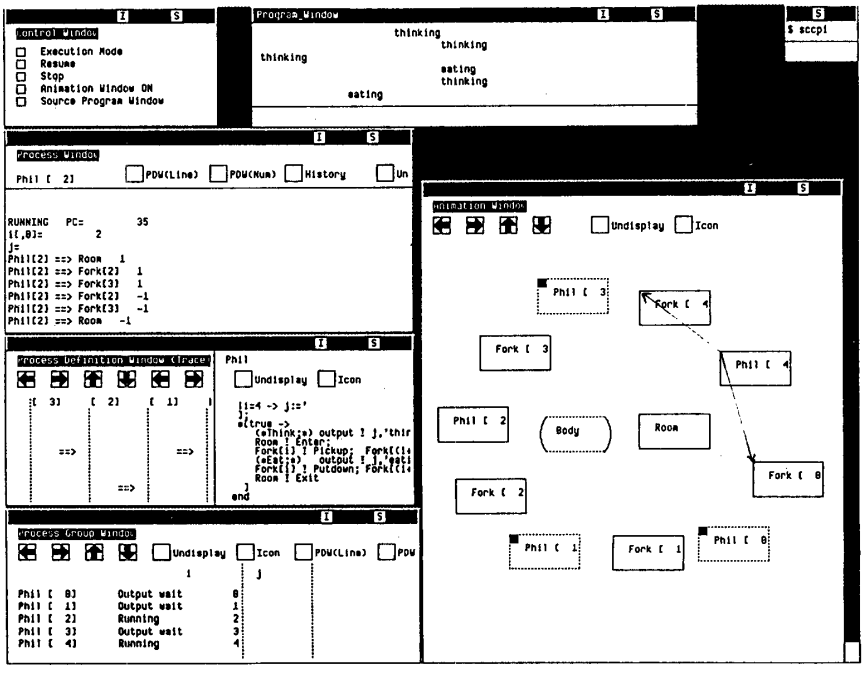

図3．シミュレーション中における画面表示例