

TR-702

負荷分散設計支援ツール ParaGraph の
動画機能の試作

久保 秀行、神子 真弓、松澤 史子、
相川 聖一（富士通）

October, 1991

© 1991, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~5
Telex ICOT J32964

Institute for New Generation Computer Technology

負荷分散設計支援ツール ParaGraph の動画機能の試作

久保 秀行 神子 真弓 松澤 史子 相川 聖一

富士通株式会社

1 はじめに

疎結合並列計算機上でプログラムを効率良く実行するためには、プロセッサ間の通信量を抑えながら計算負荷を分散することが肝要である。そこで、負荷分散戦略の設計を支援するため、プログラム実行時に収集したログ情報に基づいて、動作状況をグラフィカルに表示するツール ParaGraph [1] の開発を行なっている。これまでにプロセッサの稼働状況、プロセッサ間通信の発生状況及び並列プログラムの実行状況を種々のグラフで表示する機能を開発している。本稿では、新たに試作した並列プログラムの実行過程を詳細に表示する動画機能と簡単な例題プログラムに対する適用例について報告する。

2 負荷分散アルゴリズムの実現方法

負荷分散アルゴリズムを実現するには、問題を部分問題に分割する方法と、分割された部分問題を各プロセッサへ割り付ける方法について検討する必要がある。分割方法の検討では、計算負荷を分散させることによってプロセッサ間通信が増大しないように、ある程度独立に計算可能な単位に問題を分割することが重要である。また、割り付け方法の検討では、ある特定のプロセッサに計算負荷が集中しないように、負荷を均等に各プロセッサに割り付けることが重要である。これらの検討結果に基づいて負荷分散アルゴリズムは、並列推論マシンのプロトタイプであるマルチ PSI/V2 上では、並列論理型言語 KL1 [2] で実現される。KL1 は、暗黙の並列性と同期機構を持つ FGHC を基本部分とし、プログラムの実行単位であるゴール毎に並列に実行する機能を提供している。図 1 に KL1 プログラム例を示す。図中の述語 `foo` (PE1, PE2) のボディゴール `a`, `b` にはプラグマ '@node(PEi)' が付加されている。本プラグマはボディゴール `a`, `b` を PEi で指定したプロセッサ番号に割り付けるための負荷分散指示である。このため、負荷分散アルゴリズムを実現するには、各プロセッサ上でのゴールの振舞いを予め予測しながらプログラムを作成する必要がある。しかし、現状では並列に動作するプログラムの性能を予測することが困難なため、プログラマが思考錯誤を繰り返しながら負荷分散アルゴリズムの設計を行なっている。

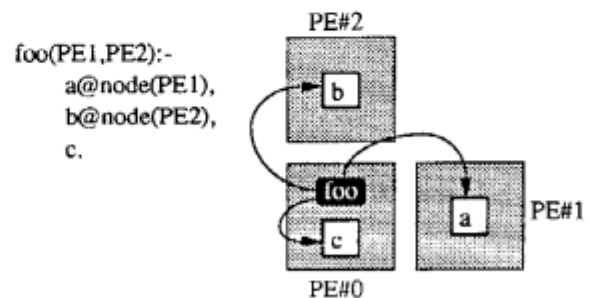


図 1: KL1 プログラム例

3 動画機能の試作

上記の問題を解決するために、各プロセッサ上でのゴールの振舞いを動画で表示する機能を検討し、試作した。本機能はプログラム実行時に収集したログ情報に基づいて、実行過程をグラフィック表示することにより実現している。なお試作はマルチ PSI/V2 上で行なった。

3.1 動作情報の抽出

並列プログラムの動作を把握するために、マルチ PSI/V2 上の並列オペレーティングシステム PIMOS[2] が提供する莊園を利用して、ユーザプログラムを解釈実行しながらログ情報を作成する。莊園とは、KL1 プログラムの実行や資源を管理するための基本機能であり、入力ゴールを莊園内で実行することで、述語が選択した節の系列（並列プログラムの実行木）やゴールを実行したプロセッサ番号及び述語名等のプログラムの動作情報を抽出することができる。また、無限ループやデッドロックのようにプログラムが正常終了しない場合においても、プログラムの実行を強制終了し、その時点までに収集された動作情報をログ情報として格納することができる。

3.2 実行過程の表示

実行過程の表示では、抽出された動作情報に基づいてプログラム中の構成要素と画面上の図形を対応付けて表示する。図 2 に N クイーン問題を解くプログラムの実行過程の表示画面例を示す。

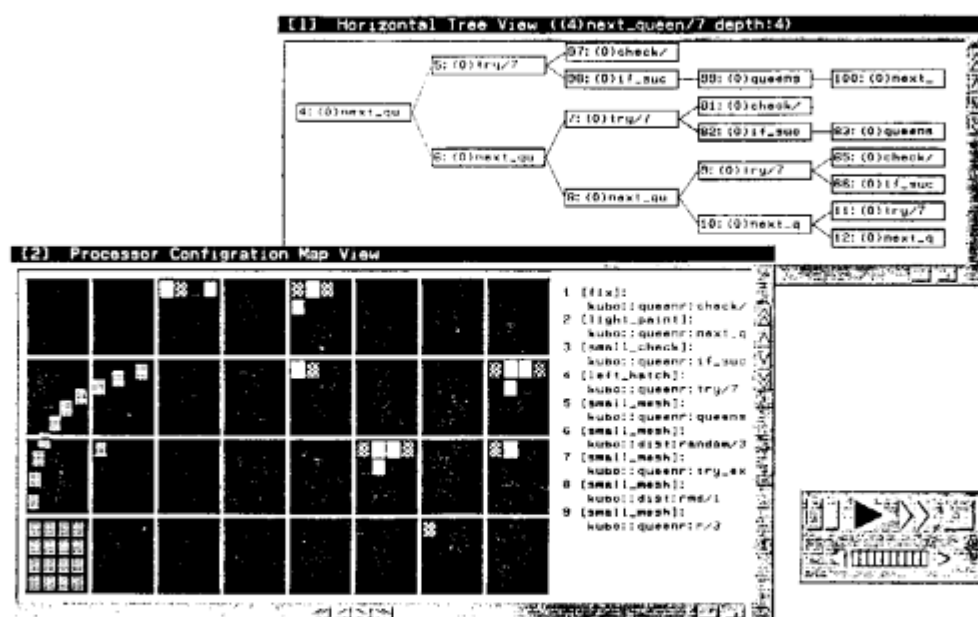


図 2: 表示画面例

(1) プロセッサ構成図上でのゴールの表示

図 2 の (a) はプロセッサ構成図 (32 台構成) 上でのゴールの実行過程の表示例である。図中の矩形はプロセッサを表し、プロセッサ内の小さな矩形はゴールを表す。ゴールは述語名別に色分けして表示される。実行過程の表示はゴールがリダクションすると対応する図形を消去し、新たに導出されたサブゴールに対応した図形を表示していくことによって実現している。この際、ゴールが分散実行される場合は、ゴールを表す矩形が送信先のプロセッサへ移動する (軌跡を描画する) ことで表現している。図では、ある特定のプロセッサ (プロセッサ 0 番) にゴールが集中している状態が表示されていることから、各プロセッサに対する負荷が均等でないことが分かる。また、正常終了しないプログラムの実行過程を本形式で表示することにより、無限ループやデッドロックを引き起こしているゴールを表示画面から特定することができる。なお、画面左下のパネルをマウスクリックすることにより、動画の停止、再開、ステップ実行等の制御が可能である。

(2) 並列プログラムの木構造による表示

図 2 の (b) はゴールの呼び出し関係を表す並列プログラムの実行木の表示例である。図中の矩形はゴールを表し、矩形の内部にはゴールの識別子、ゴールが実行されたプロセッサ番号及びゴールの述語名を表示している。木構造の表示では、表示するゴールと呼び出し関係の深さを指定することにより、並列プログラムの実行木中の任意の部分木を表示することができる。図より大部分のゴールがプロセッサ 0 番で実行されていることが分かる。

従って、プロセッサ 0 番で実行されたゴールの中で、ある程度独立に計算可能なゴールを他のプロセッサで実行するように負荷分散アルゴリズムを改良すれば良いことが分かる。

4 おわりに

並列プログラムの実行過程を動画で表示する機能を試作し、その適用例について述べた。小規模な (数万リダクション程度の) 例題に適用した結果、各プロセッサへのゴールの割り付け方法の改良や、正常終了しないプログラムのデバッグの際に各種の表示が有効であることが分かった。しかし、試作した動作情報の抽出処理では、各プロセッサで抽出した動作情報をプロセッサ 0 番で解析しているため、プログラム実行時にオーバーヘッドが発生している。このため、今後は本抽出処理の効率化を図ると共に実用規模のプログラムのデバッグも支援できるように動画機能の改良・拡張を進めていく予定である。

謝辞

本研究は、第五世代コンピュータプロジェクトの一環として行なわれた。ご支援頂きました近山 ICOT 第二研究室室長を初めとする方々に、深く感謝致します。

参考文献

- [1] 相川, 神子, 久保, 松澤, 近山, “負荷分散支援ツール ParaGraph”, 情報処理学会研究報告, 91-PRG-3, 1991
- [2] T. Chikayama, et al., “Overview of the parallel inference machine operating system”, In Proceedings of the International Conference of FGCS, 1988