研 究 速 報

# 並行プログラムのインタラクティブ視覚化への投機的計算の応用

Application of Speculative Computation to Interactive Visualization of Concurrent Program Execution

舘 村 純 一*
Junichi TATEMURA

## 1. はじめに

並行プログラムは，実行の流れが複数あるためにその実行の様子が理解しにくい．これを解決するために，デバッグなどにおいて実行視覚化技術が重要視されている．このようなシステムで大規模なプログラムを視覚化する場合には，大域的な情報を把握しながらユーザの必要とする情報を抽出する手法が大きな課題となり，実行データのフィルタリング・クラスタリングなどをユーザの視点に基づいて行う必要がある．この時，視点の移動に対してダイナミックに適応しながら情報を抽象化する機能が重要になると考えられる．本研究では，インタラクションに応じたダイナミックな視覚化処理を実現するために投機的並列実行を導入する手法を提案する．視覚化の対象としては，並行論理型言語KL1によって記述された並行プロセス指向のプログラムの実行情報をとりあげる．

## 2. 投機的計算を応用したインタラクティブ・プログラム

投機的並列計算とは，近年の並列処理研究の中で生まれてきた概念であり，プロセッサ資源に余裕のある時に，将来利用されると思われる計算をその要求が確定する前に行っておくというものである．

ヒューマン・インタラクションを中心とするプログラムは，ユーザの入力の逐次性のため従来の並列化の対象にはされてこなかった．しかし，ヒューマン・コンピュータ・インタラクションには複数のイベントや多数のデータが関わっているので，本質的にはイベント並行性やデータ並列性が存在すると考えられる．我々は，以下の目的のために，インタラクティブなプログラムに投機処理の考え方を応用する．

• 多数のプロセッサの有効利用
ネットワークで結合されたワークステーション群や，稼働率がピーク時以外の並列計算機などの余剰計算資源を活用する．

• インタラクティブ・サービスの充実
プログラム並列化による一定処理の高速実行ではなく，一定時間内の処理量の増大によりサービスの質を向上させるという発想で，並列処理応用を考える．

これにより，以下のような効果が期待できる．

（1）インタラクティビティ（応答性能）の向上
要求を受けてからでは時間のかかる処理のターンアラウンドの短縮が可能となる．

（2）提供する情報の質の向上
制限時間までにかけられる計算時間の増加に応じて，データの精度・最適化の度合などの向上をはかる．

（3）先行実行結果のフィードバック
エラーの事前回避や，ユーザへの提案・予報をアクティブに行う．

## 3. 並行プログラムのインタラクティブ視覚化

並行プログラムの実行を図的に表現してプログラムの並行性の理解を助けることを目指した例としては[1)~3)]などがあげられる．これらの手法の大きな課題は，並行プロセスが大量になった場合のスケーラビリティである．

大規模データの表示の問題を解決するには，画面上の表示データを削減するなどしてユーザの理解を助ける手法が重要となる．これは情報視覚化技術において共通した課題であり，グラフ構造や木構造の表示に魚眼表示(fisheye)[4)]，フラクタル[5)]などを応用した研究例がある．いずれの例も，データ構造の表示にユーザの「視点」を導入し，視点部分を詳しく表示しつつその他の領域も概観がつかめるように表示している．これらの方法は，グラフの構造的な情報を主に用いた汎用的な表示手法である．これ

*東京大学生産技術研究所　第3部

研　究　速　報

に対して，プログラム実行の視覚化には，どのような動作に着目するかなどのプログラムの意味上での「視点」が重要であり，上記の手法に加えて，データのクラスタリングなど，グラフの意味的な情報に基づいた抽象化が有効と考えられる．

一方，並行論理型言語のビジュアル・デバッガHyperDEBU[6]では，ユーザが何を見たいのかをブレークポイントとして指定することによって，視点に基づいた情報の抽象化を行っている．この機能は，ユーザの着目するプロセス構造や通信の相互関係の視覚化を可能とする．しかし，表示の一部の拡大や詳細な実行情報への視点移動は可能だが，視覚化の視点を変えるにはブレークポイントを変更して再実行する必要がある．すなわち，ブレークポイントに基づいた実行情報の視覚化部分に対してはダイナミックな視点の移動ができない．一般に，プログラムの構造や動作を良く知らなければ望ましい視覚化情報を得るような視点指示は難しく，試行錯誤を通じて適切な表示をインタラクティブに得る機能が望まれる．

以上のように，大規模並行プログラムの視覚化には，視点の移動に対してダイナミックに適応しながら情報を抽象化する機能が必要である．

## 4. 視覚化システムの投機的並列処理

### 4.1　並行論理型言語 KL1 の実行の視覚化

本研究で対象とするプログラムは並行論理型言語におけるプロセス指向のプログラミングで記述されたものとする．これは，プロセスネットワークを構成して計算を分散的に行うプログラムの記述法であり，再帰的なゴール実行で並行プロセスを表現し，単一代入変数を共有することで通信路（ストリーム）を表現する．プロセスネットワークは動的に構成されるので，ソースコードを観察しているだけではその実行の様子を把握することは難しい．そのために実行情報の視覚化を行うが，実行全体の情報は膨大すぎるため，ユーザがその時持っている疑問点，着目点に対応した実行情報のフィルタリング・クラスタリングを行うことが重要である．

プロセスネットワークのある状態を視覚化する場合，視点によって各部の抽象化の度合が違ってくる．プロセス一つ一つの動きに注目するのか，プロセス群としてまとめて抽象化するべきか，多くのデータフロー（通信の依存関係）のうちいずれに着目して実行を見せるか，といった要求に対応して視覚化を行う．

図1-(a) は，プログラムの実行の視覚化作業を表したものである．始めに，ユーザは着目点を表現する「視覚化指示」をシステムに与える．システムはこの指示に基づいてどこをどのように視覚化するかを解析する．この結果に基づき，ユーザの観点に応じた視覚化画面を提示する．この表示結果を観察検討して，ユーザは視覚化指示を調整する．本研究では，この繰り返し作業に投機処理を導入することでインタラクティブ性の向上を目指す．図1-(b) はその概念図である．

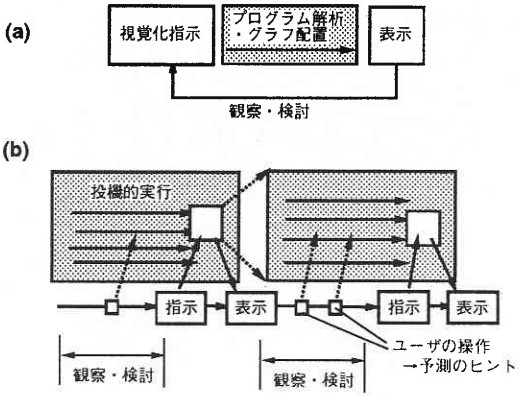


図1　プログラム視覚化作業の投機並列化

### 4.2　視覚化手法

ユーザの観点に適応した視覚化は，以下の処理によって行われる．

• 視覚化対象の選択

並行論理型言語のプロセスは動的に生成されるので，実行履歴は図2のような木構造を形成する．この実行履歴のなかからある時点で同時に存在しうるプロセスの集合（論理的スナップショット）を選択してプログラムの状態を表現する．ユーザが一つまたは複数のプロセスを指定することで，このプロセス（注目するプロセス）を含むスナップショットが選択される．このスナップショットは一般に複数存在するが，注目するプロセスを生成するのに必要最低限の実行を行った状態を示すように，木のルートに近い（時刻の早い）プロセスを選択する．

• グルーピング

プロセスの親子関係の木構造に基づいて階層的なクラスタ

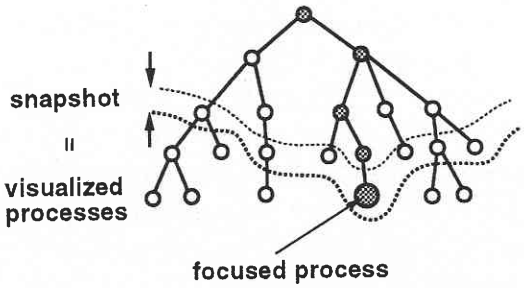


図2　視覚化されるプロセスの選択

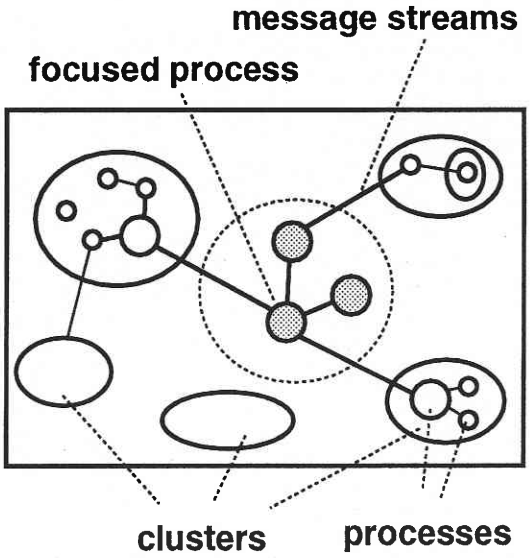

図3　視覚化の概念図

を構成することにより，プロセスをグループ化する．これは，注目点以外の部分を適度に抽象化する働きをもつ．

・重要度に基づく配置

注目するプロセスとのデータ依存関係に基づいて，各プロセスの重要度を計算する．この重要度に応じてプロセス（およびクラスタ）の画面上での大きさを調節する．図3は表示されたプロセスの様子を表す視覚化の概念図である．

### 4.3　ユーザの視覚化指示

ユーザは注目点をシステムに示すために以下のような視覚化指示をインタラクティブに行う．

・注目するプロセスの選択

表示されているプロセスから注目するプロセスを選択する．複数の視点を導入するために，複数のプロセスが選択できることが望ましい．また，画面上でのインタラクティブな選択の他に，ソースコードに基づいた検索をサポートすることが考えられる．

・プロセスのズーム

選択したプロセスを，その小プロセスに分解して再表示する．これは，計算履歴の木において，視点をあるノードから小ノードへと移動させることに対応する．これにより新しく現れるプロセスを含むようなスナップショットが新たに選択されるため，注目するプロセスのステップ実行の働きをもつ．

### 4.4　投機処理の導入

システムはユーザの視覚化指示を予測し，これに基づいて要求される処理を先行実行する．投機処理の対象としては，プログラム解析とグラフ配置計算があげられる．

視覚化されるプロセスの重要度を計算するためには，タイプ/モード解析，データ依存関係の抽出などのプログラム解析が有効である．計算資源を投入できればそれだけ詳しい解析が可能となる．ただし，これが実際に投機処理に値するかどうかは，解析のための計算量とその効果を具体的に評価する必要がある．

グラフを視認性よく配置するための自動描画のヒューリスティック・アルゴリズムとして配置問題をエネルギー最小化の問題に帰着させる手法などが研究されている（7)など)．このような手法では，最適化にかける計算時間に応じてグラフ配置の品質が向上すると考えられるので，投機的実行の対象になりうる．また，表示要求されるグラフ構造を完全に予測できなくても，先行実行により得られた部分解や類似解を初期値に適用することで，品質の良い配置を早く得られる可能性もある．

また，投機処理の効率向上のためには可能性のある視覚化指示の中から有効と思われるタスクを選択して実行しなければならない．視覚化指示の予測には，（1）部分的に指定された視覚化指示，（2）ユーザのブラウズ操作，（3）プログラムの解析結果，などの情報が有効と思われる．この予測に基づき，投機タスクの選択とその優先度の決定を行う．

### 4.5　投機的処理の効果

投機的処理導入で期待される最大の効果は視覚化指示と結果表示の繰り返しにおけるターンアラウンドの短縮である．この効果はダイナミックな視覚化のために重要な要素である．またこれに加えて，（1）アクティブな情報提供（2）計算資源に応じた品質の提供，の各効果が期待できる．（1）としては，視覚化すべき部分を積極的にユーザに提示したり，バグなどの検知結果を示すことなどが考えられる．（2）にはグラフの配置やプログラム解析の結果が該当する．

## 5. お　わ　り　に

現在，並行論理型言語KL1の処理系 KLIC[8] 上での実行視覚化システムを設計・実装中である．KLICは，逐次版およびワークステーション・クラスタ上の分散メモリ並列版に Tcl/Tk による GUI インタフェースを実装して利用している．課題としては，（1）プログラム解析結果の視覚化指示への反映手法，（2）ダイナミックに視覚化指示を与えるための GUI 設計，（3）投機的並列実行の効率的実装，（4）システムの評価手法などがあげられる．

(1995年12月18日受理)

## 参　考　文　献

1)　K. M. Kahn. Concurrent Constraint Programs to Parse

and Animate Pictures of Concurrent Constraint Programs. FGCS'92, pp. 943-950, 1992.
2) 高田，小池．Visualinda: 並列言語 Linda のプログラムの実行状態の3次元視覚化．WISS'94, pp. 215-223, 1994.
3) 田中．並列論理型言語 GHC のビジュアル化の試み．WISS'93, pp. 265-272, 1993.
4) M. Sarkar and M. H. Brown. Graphical Fisheye Views. CACM, Vol. 37. No. 12, pp. 73-84, 1994.
5) 小池，石井．フラクタルの概念に基づく提示情報量制御手法．情報処理学会論文誌，Vol. 33, No. 2, pp. 101-109, 1992.
6) 舘村，小池，田中．マルチウインドウデバッガ HyperDEBU における細粒度高並列プログラムの実行のデータフローの視覚化．情報処理学会論文誌，Vol. 34, No. 4, pp. 580-594, 1993.
7) 鈴木，鎌田，榎本．単純無向グラフ自動描画アルゴリズム．コンピュータソフトウェア，Vol. 12, No. 4, pp. 45-55, 1995.
8) 仲瀬，藤瀬，近山．ポータブル KL1 処理系 KLIC の概要．情報処理学会第47回全国大会，pp. 5-55-56, 1993.