

ユーザ・インタフェース応用のための投機的並列計算システムの構想

5 T-3

館村純一

tatemura@sak.iis.u-tokyo.ac.jp

東京大学生産技術研究所*

1 はじめに

メモリ容量の増大やプロセッサの速度向上によって、計算機の使われ方は質的に変化してきた。現在のワークステーションなどでは、その計算資源の多くがユーザ・インタフェースに用いられており、人間の知的活動を支援するメディアとしてのコンピュータが注目されている。一方、並列計算機が現在実用化されつつあるが、その利用形態はまだ確定しておらず、利用技術の研究が並列計算機の普及にとって不可欠となっている。また、これらの並列機を含む計算機群と高速ネットワークによって構成される分散環境も豊富な並列計算資源となりうる。人間の数よりもプロセッサの数の方がはるかに多いという環境では、今までには考えられなかったことにもプロセッサを活用できる可能性があり、計算機の新しい利用技術が期待される。

本研究は、知的活動を支援するインタラクティブ・メディアとしての計算機への並列・分散処理の活用を目指し、「投機的並列計算」と呼ばれる概念を導入して、並列計算資源を有効に活用したユーザ・インタフェースを構築することを目的とする。

本稿では、研究の構想を概説し、投機的並列計算システム実現に向けての課題を明らかにするためにワークステーション・クラスタ上に試作した簡単な実験プログラムについて述べる。

2 投機的並列計算

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なっておくというものである。従来は、主に命令レベルの細粒度並列処理に活用されてきた [1] が、この考え方をヒューマン・インタラクションを持つプログラムに応用することで、以下の目的をめざす。

- 多数のプロセッサの有効利用: ネットワーク上のワークステーションや、実行並列度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。

3 期待される適用例

ユーザの希望の可能性を見越して、要求前に計算をしておくことにより、以下のような効果が期待できる。

- 提供する情報の質の向上
- インタラクティブティ（応答性能）の向上
- 冗長計算による信頼性・頑健性の向上

このようなユーザ・インタフェースの適用例としては、以下のものが考えられる。

- プログラミング支援（記述途中の不完全情報の解析によるエラー検出、最適化、バージョンアップ時の不整合性管理）
- 情報検索・ナビゲーション（要求情報の先行処理、情報リンクの動的構築）
- CAD・発想支援など（ユーザが逐次与える制約条件を満たすような解の探索・提示）

4 ワークステーション・クラスタ上の実験システム

投機的計算システムの性能決定要因を分析し、スケジューリング戦略、プログラミング技法、実装技術などを検討するため、予想される実例の典型的要素を含んだ簡単なプログラムを開発した。ハードウェアとしては、ネットワーク上の複数のワークステーション（ワークステーション・クラスタ）を対象とし、実装には並列プログラミングライブラリ PVM (Parallel Virtual Machine)[2] を利用した。

4.1 システムの概要

問題例として、あるデータについて (1) パラメータの設定 (2) 計算 (3) 結果の検討を繰り返しながら、ユーザにとって望ましいパラメータを得るという作業を考える。これは図 1 の破線枠内のような処理の繰り返しであるが、パラメータが変更された時点で次回以降の変更を予測して投機的に計算を先行実行することで、応答速度の向上を計る。

*Towards a Speculative Parallel Computing System Applied for User Interfaces
Junichi, TATEMURA,
Institute of Industrial Science, University of Tokyo

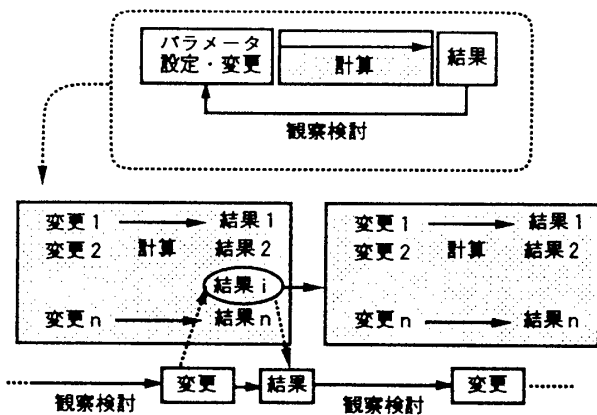


図1: パラメータ・チューニングの投機並列化
これは、デザイン支援、インタラクティブ・シミュレーションなどへの応用のほか、パラメータ空間上の移動を情報空間上の移動と捉えれば、ハイパーメディアや各種ビューアで要求前に次画面を計算して用意することなどにも応用できる。

4.2 システムの実装

システムは、実行管理をするマスタプロセスと投機計算をするスレーブプロセスからなり、マスタは、投機タスクを集中管理しながらパラメータの送信・結果の受信・タスクの中止を行なう。スレーブは、パラメータを受信し疑似計算を一定時間行いデータを返す。ユーザはパラメータ空間を1ステップで1単位ずつ移動するものとし、その近傍を投機処理する。測定のため、ユーザの入力はタイマーと乱数を用いて模擬した。現実装では、1ステップ、2ステップ先を投機対象とし、ステップ数(先行段数)に応じて優先順位実行管理がなされる。

4.3 性能測定

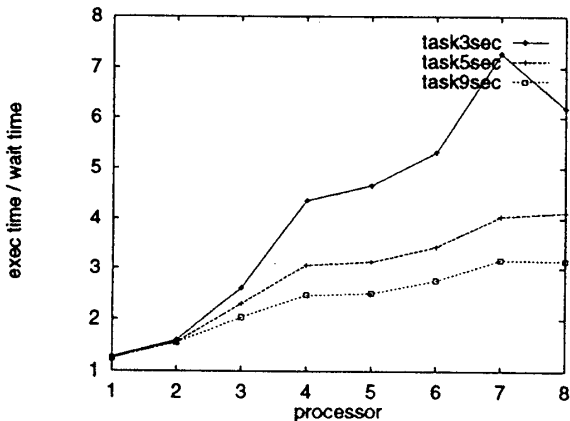


図2: 応答速度向上率

タスク処理時間を除いたマスタ・スレーブ間の応答時間は以下のようにになった。

転送データサイズ (byte)	400	4k	40k	400k
応答時間 (msec)	12	22	170	1500

3パラメータ・単調増加・先行実行段数2のシステム(すなわち、1段先行タスク数3、2段先行タスク数6)について、ユーザ入力間隔2(sec)、データサイズ400(byte)の場合の測定を行なった。タスク処理時間が3,5,9(sec)それぞれの場合に応答速度向上率(=タスク処理時間/応答時間)をプロセッサ台数1~8について測定した結果は図2のようになった。プロセッサ台数8のときのタスク処理時間(sec)と応答時間(sec)の関係は図3のようになった。

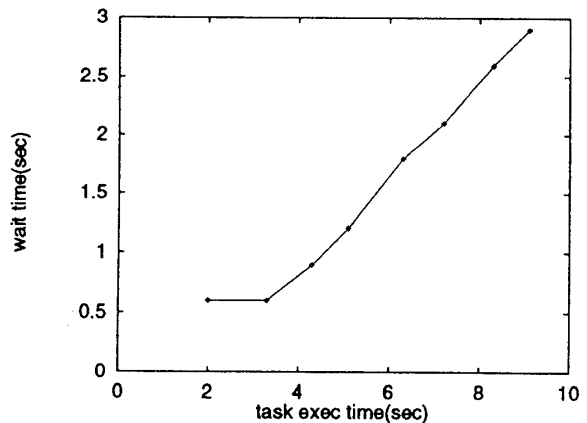


図3: 処理粒度-応答時間グラフ

5 おわりに

今後の課題としては、(1)その他のベンチマークの開発と測定結果の考察、(2)スケジューリング手法、投機タスク内並列処理、投機タスク間の計算・データの共有、効率的な分散管理方式などの実装技術、(3)計算量の爆発を避けるための、問題に応じた並列探索アルゴリズム、問題知識を用いた予測による投機効率の向上、(4)本格的アプリケーションの開発、(5)ライブラリの蓄積、並列言語・並列/分散OSの支援機能の検討、などがあげられる。

謝辞

東京大学の坂内正夫教授、田中英彦教授、および両研究室の方々には、議論に参加していただき、また計算機資源を提供していただきました。

参考文献

[1] 山名早人, 他: 投機的実行の現状と Unlimited Speculative Execution Scheme の提案, 情報処理学会研究報告 94-ARC-107 pp.105-112 (1994).
[2] Geist, A., et al.: PVM 3 user's guide and reference manual, ORNL/TM-12187, (1994).