

インタラクティブシミュレーションのための投機計算システムの実装

岩永 翔太郎[†] 松井 祐太[†] 花房 秀光[†]
福間 慎治[†] 森 眞一郎[†]

1. はじめに

我々は次世代のシミュレーション技術として、高性能な計算サーバ上での数値シミュレーションにユーザが直接介入し、シミュレーションのシナリオを実時間かつ対話的に変更するインタラクティブシミュレーション技術の研究を行っている。次世代プロセッサのメモリアダプタ化は、非常に多くの並列処理環境を我々に提供する。しかしながら中・小規模のアプリケーションでは、これらの計算資源を十分に活用することが困難である。そこで、インタラクティブシミュレータの実用化に向けて、従来の並列処理手法のみでは実現不可能な精度のシミュレーションを、有り余るコンピュータ資源を用いた並列処理とユーザ介入の連続性を利用した投機計算を用いた高速化により実現する手法を提案する。

2. 研究背景

本研究では、多くのシミュレーション中に現れる連立一次方程式 $Ax = b$ の求解問題に着目し、シミュレーション中のユーザからのインタラクションによって係数行列 A の一部の要素のみが変化をするインタラクティブシミュレーションを想定し、係数行列が変化するタイムステップと係数行列が変化しない複数のタイムステップが交互に発生するモデルを仮定する。係数行列が変化するステップから次に係数行列が変化するステップまでの間をステージと呼び、 i ステージ目の係数行列 A_i が $A_i \approx A_{i+1}$ となる性質を持つものとする。ここで、連立一次方程式 $Ax = b$ の数値解法として前処理付き共役勾配法を採用する。

そこで、係数行列の変化のパターンを予測して前処理行列の計算を投機的に行うシミュレーションモデルを提案する。一般に、投機計算を行うモデルでは予測が外れた場合には最も実際のものに近い行列を用いて近似解とする¹⁾か変化した後の実際の行列を用いて再

計算を行う必要があり、計算精度か実時間性のどちらかを妥協しなければならない。今回提案する手法では、予測が全て外れていた場合でも、投機的に求めた前処理行列を本来必要とされる前処理行列の近似行列として代用することで再計算の発生を排除しつつ計算精度を保つことができると考えられる。

3. 投機システムの実装

投機計算システムを構成するハードウェアの実体は、外部からの入力を受け付ける操作端末、操作端末からの入出力を行う管理ノード、そして、前処理行列の投機計算を行う投機計算ノード群、解計算を行う解計算ノード群である。

操作端末では、 b を管理ノードに送信し、管理ノードから x を受信する。管理ノードでは、操作端末とのインタフェースならびに投機計算処理および解計算処理を管理する。あるステージにおける係数行列を A_i とすると、管理ノードは A_{i+s} を複数予測し投機計算ノード群に送信し、その結果として、予測した A_{i+s} に対する P_{i+s} を受信する。また、管理ノードはその時点で最新の A, P と b を解計算ノードに送信し x を受信する。投機計算ノード群では、受信した A に対する P を計算し、管理ノードに送信する。解計算ノード群では、選ばれた P と A, b をもとに x を計算し、管理ノードに送信する。

実時間性を実現するため、投機を行うためのオーバーヘッドは微小なものでなければならない。そのため管理ノードの負荷を軽減し、通信量を減らす必要がある。

図1は前処理行列の計算が1ステージ内に予測が終了するという仮定のもとで、 s が1の場合の処理フローである。図1では $i+1$ ステージ以降、投機状態のプロセスと試行計算状態のプロセスが平行して動作する。*pattern* は未来の係数行列を予測するために必要な情報で、この情報から投機プロセス側が未来の係数行列を生成しそれをもとに前処理行列を導出する。*complete* は前処理行列の導出が完了したことを示す信号で、*status* は各資源ノードの状態遷移先の

[†] 福井大学大学院工学研究科

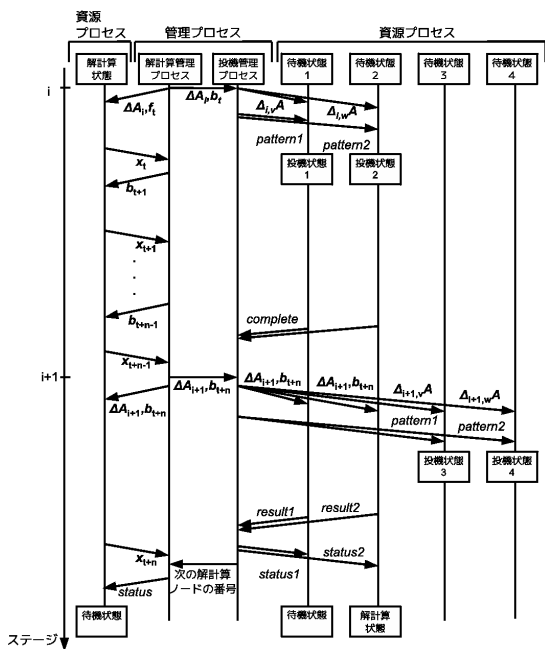


図 1 1 段階で 2 パターン投機予測した場合

情報である。

3.1 管理ノードの実装

管理ノードでは 1 ステージ内に処理を行えば良い投機計算と 1 ステップ毎に処理を行う解計算の両方の管理が必要である。これらの処理を単一プロセスで管理するとスループット重視の処理とリアルタイム重視の処理が混在し、管理ノードの処理が煩雑になる。そこで管理ノードを投機計算管理プロセスと解計算管理プロセスの 2 つに分割しそれぞれに担当させる。

3.2 資源ノードの実装

資源ノード群では、投機計算、試行計算、ならびに解計算の 3 種の処理を行うが、計算機資源に 3 つの処理を固定的に割当てていたのでは、密行列となる可能性がある前処理行列 P の通信が必要となり実時間性の確保は不可能である。そこで、 P を生成した投機計算プロセスを次ステージから解計算プロセスとして使用することでオーバーヘッドを軽減するシステム構成とした。その結果資源ノードは管理プロセスから情報を受けて図 2 のような状態遷移を行う。

4. 評価

計測には肝臓を有限要素モデル化したものを係数行列として用いた。行列のサイズは 3759×3759 で、非零要素数の割合は約 1.0[%] である。ある頂点 1 つに注目し、隣接した頂点との連結を強くするような変化 (肝組織の硬化に相当) を加え

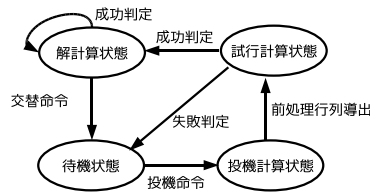


図 2 資源ノードの状態遷移

表 1 実験結果

	本システム	逐次処理
前処理行列導出 [ms]	211	203
反復計算 [ms]	137	135
応答時間 [ms]	143	342

実験を行った。実験の実行環境は CPU:Core2Duo 2.66[GHz],L2Cache:4[MB],Memory:2[GB], OS:Linux 2.6.19-1.2895.fc6 を 8 ノード用意し,Gigabit Ethernet で結合した。本システムを使用した場合と、本システムを使わず操作端末と解計算プロセスが 1:1 で通信を行う場合の、 ΔA_i の非零要素数が 169 の時のデータを表 1 に示す。本システムの方では 1 ステージに前処理行列導出が完了するとし、1 ステージ内で 2 パターンの投機予測を行ったものである。ただし、今回の実験では複雑な予測計算は行っておらず、*pattern* として A_i から前処理行列を導出するか、前処理行列として単位行列を用いるかの 2 パターンを用意し実験を行った。

表 1 の応答時間については、操作端末が b を送信終了した直後から x を受信終了直後までの時間で、操作端末側で計測を行った。

5. まとめ

本論文では、前処理行列を投機的に計算することで実時間性を保証する投機システムの実装について述べた。また、簡単な評価を行った結果、投機計算により前処理行列導出時間が隠蔽されていることが確認できた。また、反復計算時の管理ノードのオーバーヘッドも十分小さいことも確認できた。

参考文献

- 1) 小木哲郎, 渡辺浩志, 廣瀬通孝: 並列冗長予測法によるインタラクティブ・ビジュアライゼーション, 日本機械学会論文集 (C 編), 66-647, 2339-2346 (2000)
- 2) 依藤 逸, 他: 操作の連続性を考慮した投機計算を利用するインタラクティブシミュレータ, IPSJ Symposium Series, Vol.2009, No.5, pp. 229-238.