電力系統過渡安定度計算の階層的並列処理手法

4L－9

西川 健†, 前川 仁孝†, 中野 恵一‡, 笠原 博徳†

早稲田大学理工学部電気工学科† オリンパス光学株式会社‡

1 はじめに

近年、電力需要の増大に伴い、電力系統は大規模化、複雑化しており、系統の運用、計画のために電力系統解析計算の高速化が求められている。その中でも、過渡安定度計算は電力系統の動的な安定度を解析するもので、リアルタイム計算負荷が大きく並列処理による高速化が求められている [1]。従来の電力系統解析計算の並列処理では、系統分割により粗粒度タスクを生成し、分割した系統ごとにプロセッサに割り当てる粗粒度並列処理に関する手法が研究されてきた [2][3]。しかし、この手法では、発電機数よりプロセッサ数の方が多い場合には並列効果を出すことが難しい。そこで本稿では、この系統分割による粗粒度レベルでの並列処理に加えて、分割した各系統内において、発電機を表す微分方程式と系統特性を表す代数方程式についてループフリーコードの生成を行い、このループフリーコードから近細粒度タスクを生成することにより並列処理を行う近細粒度レベルの並列処理を階層的に組み合わせる階層的並列処理手法の提案を行う。また、本手法の有効性を評価するためにマルチプロセッサシステム OSCAR シミュレータ上で性能評価を行った結果について述べる。

2 過渡安定度計算の階層的並列処理手法

本節では、電力系統過渡安定度計算の階層的並列処理手法について述べる。

一般に過渡安定度計算は、(1) 式の発電機の動特性を表す非線形微分方程式と、(2) 式の系統を表す線形代数方程式でモデル化される。

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}, \mathbf{V}) \quad (1)$$

$$\mathbf{I}(\mathbf{x}, \mathbf{V}) = Y\mathbf{V} \quad (2)$$

ここで、$\mathbf{x}$ は発電機の状態変数ベクトル、$\mathbf{V}$はノード電圧ベクトル、$\mathbf{I}$はノードへの注入電流ベクトル、Yはノードアドミタンス行列である。

この計算手法には、(1) 式と (2) 式を交互に解く分割解法と、(1) 式を代数方程式に変換して (2) 式と同時に解く同時解法がある。本研究では、系統の状態が変化しない限りノードアドミタンス行列を定数として扱え、計算時間の点で有利である分割解法を採用する。

発電機モデルは、シミュレーションの時間と精度の点で、X'_d モデル、過渡モデル、詳細モデル（Park モデル）の３つに分類される。本稿では、３つの中で最も精度の良い詳細モデルを採用する [4]。

本研究では、このような電力系統過渡安定度計算の並列処理を行なえる過渡安定度計算専用目的コンパイラの開発を行なった。本専用目的コンパイラは、ユーザがデータを入力することにより、自動的にタスク生成、タスクスケジューリングを行ない、並列マシンコードを生成する。以下では、タスク生成、タスクスケジューリングについて述べる。

Hierarchal Parallel Processing of Transient Power System Stability Calculation Method

Takeshi NISHIKAWA, Yoshitaka MAEKAWA, Keiichi NAKANO, Hironori KASAHARA

† Dept. of Electrical Engineering, Waseda University

‡ Olympus Optical Co., Ltd.

2.1 粗粒度及び近細粒度タスクの生成

本節では、系統分割を適用した際のシミュレーション計算手法と、粗粒度並列処理の単位となる粗粒度タスクの生成手法について述べる。

以下に、系統分割を適用した場合の計算手法について説明する。

発電機を表す (1) 式については、計算において各発電機間にデータ依存はなく、それぞれの所属する分割系統を計算するプロセッサに割り当てることにより、データ転送を削減することが可能である。しかし、プロセッサ数と発電機数の関係によっては、発電機タスクが均等に各プロセッサに割り当てられず、プロセッサ間に負荷の不均衡をおこすことが考えられる。

そこで、本手法では必ずしもプロセッサ数で系統を分割するのではなく、分割した系統内でステートメントレベルの近細粒度タスクを生成し、分割系統間の粗粒度並列処理と分割系統内の近細粒度並列処理とを階層的に組合せて並列処理を行なう。

次に、電力系統の系統部を表す (2) 式については、これを解くための連立方程式を生成する際にノードで系統分割し、分割点を縁側に移動することにより、図1のような縁付ブロック対角行列を得る。

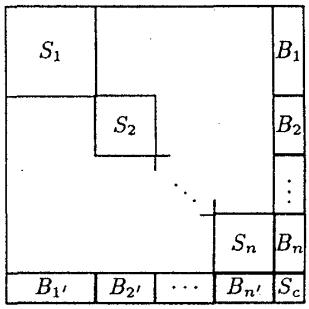

図 1: 縁付ブロック対角 (BBD) 行列

このような行列を LU 分解法を用いて解く場合、部分行列 $S_k(k = 1, 2, \cdots, n)$ は並列に LU 分解が可能で, 部分行列 S_k の計算の終了後、部分行列 B_k, 部分行列 $B_{k'}$ も並列に LU 分解が可能となる。部分行列 S_cの LU 分解は、部分行列 S_k, 部分行列 B_k, 部分行列 $B_{k'}$の LU 分解が全て終了した段階で計算可能となる。また、以上のように分割することにより、LU 分解において発生するフィルインは必ず図中の要素のある部分にのみ存在する。

また、前進代入部 (FS) における計算は、部分行列 S_k, 部分行列 B_k, 部分行列 $B_{k'}$を一つのグループとする部分行列を考えると、部分行列 S_cを除いて全て他の部分行列の計算結果は不必要であるから、部分行列 S_c以外は並列に計算可能である。部分行列 S_cについては、各部分行列の前進代入の計算が全て終り次第、計算可能となる。

最後に行なわれる後退代入部 (BS) の計算は、前進代入部とは逆に部分行列 S_cに属する解の計算を実行し、この値を他の部分行列に転送することにより、残りの各部分行列が並列に計算可能となる。

これらの各処理を粗粒度タスクとして定義すると、図２のようなタスクグラフを得る。これにより、部分行列 S_k, 部分

行列 B_k, 部分行列 $B_{k'}$の LU 分解と前進代入を n 個の粗粒度タスクで並列に処理し、その後部分行列 S_cの部分の LU 分解、前進代入、後退代入を計算し、最後に各部分行列の後退代入を n 個の粗粒度タスクで処理する。

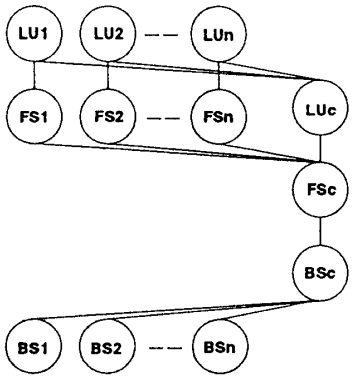

図 2: マクロタスクグラフ

図 2のようなタスクグラフにおいて、接続部分の計算は、他に並列実行可能なマクロタスクが存在しないため、すべてのプロセッサを使用した近細粒度の並列処理を行なう。実際には、先にも述べたように LU 分解は系統状態が変化した場合のみ計算を行う。従って、各時間ステップでは前進代入と後退代入のみが計算される。

2.2 階層的スケジューリング

本手法では、スケジューリング手法にデータ転送時間を考慮したヒューリスティックスケジューリングアルゴリズムである CP/DT/MISF 法を用いる [5]。

階層的なスケジューリングを行なう際には、まず系統分割より得られる各粗粒度タスク内を近細粒度タスクに分割して近細粒度タスクグラフを生成し、CP/DT/MISF 法を用いて近細粒度タスクのスケジューリングを行なう。次にスケジューリングを行なった結果得られる近細粒度タスク集合の推定実行時間を粗粒度タスクの推定実行時間とし、粗粒度タスク間のデータ依存関係より、近細粒度タスクグラフと同様な方法で粗粒度タスクグラフ (マクロタスクグラフ) を生成し、CP/DT/MISF 法を用いて粗粒度タスクレベルでスケジューリングを行なう。最後にこの粗粒度レベルでのスケジューリング結果と各粗粒度タスク内の近細粒度タスクのスケジューリング結果を組み合わせることにより最終的な階層的スケジュールを生成する。

3 過渡安定度計算の階層的並列処理手法の性能評価

本節では、電力系統過渡安定度計算の階層的並列処理手法の性能を OSCAR シミュレータ上での評価した結果について述べる。

3.1 OSCAR のアーキテクチャ

今回評価に用いたマルチプロセッサシステム OSCAR[6] は、16 台の PE と集中共有メモリ (CSM) が 3 本のバスで接続されている。各 PE は、5MFLOPS の 32bit カスタム RISC プロセッサ、ローカルメモリ、分散共有メモリ (DSM) などから成っている。このプロセッサは、すべての命令を 1 クロックで実行することができる。また、DSM を使ったデータ転送には、1PE 対 1PE の直接データ転送モード、1PE 対全 PE のブロードキャストデータ転送モードがあり、いずれも CSM を介するより高速にデータを転送できる。

3.2 性能評価

本節では、小規模系統の過渡安定度解析を例に、階層的並列処理手法を粗粒度あるいは近細粒度並列処理のみの場合と比較することにより性能評価を行なう。今回は、3 機 9 母線の系統 [4] について過渡安定度計算に要する時間を評価した。分割数はシミュレーションを行なう際の系統の分割数、PC 数は使用するプロセッサクラスタ数、PE 台数はプロセッサクラスタ内で使用するプロセッサ台数を表しており、全体で使用される総プロセッサ台数は、PC 数×PE 台数となる。

近細粒度並列処理 (PC 数を 1、PE 台数を 1,2,3,6,9 台と増やしていく) では 856.0ms, 506.2ms, 383.0ms, 300.8ms, 285.7ms となった。これに対して、粗粒度並列処理で PE 台数を 1 に固定し、PC 数を発電機数と同じ 3 台とした場合では 350.3ms となり、プロセッサ 3 台を使う限り粗粒度並列処理が近細粒度並列処理を上回っている。しかし、現在の系統分割技術を用いた粗粒度並列処理では、扱う系統に応じ並列性に限界があるために、4 台以上の場合では、発電機数より多く分割してもプロセッサ間の負荷の不均衡を起こす可能性がある。そのため、系統を 3 分割して、分割系統内をさらに 2 台のプロセッサを用いて近細粒度並列処理を行なう階層的並列処理を行なうと、計 6 台のプロセッサでの処理時間は 256.5ms となり、同じ PE 台数が 6 台の近細粒度のみの場合と比較して短い処理時間が得られた。PE 台数が 9 台の場合も同様であった。この結果より、系統の分割による粗粒度並列処理と分割系統内の近細粒度並列処理を階層的に行なう本手法の有効性が確認された。

4 まとめ

本稿では、系統分割を行ない分割系統をプロセッサクラスタに割り当てる粗粒度並列処理と、分割した系統内の計算をステートメントレベルで並列処理する近細粒度並列処理を組み合わせた階層的並列処理手法の提案を行なった。また、提案した手法の有効性を確認するために OSCAR シミュレータ上で評価を行なった結果、分割した系統に適用した粗粒度並列処理手法と分割系統内に適用した近細粒度並列処理手法を階層的に組合せることにより、従来の系統分割により分割系統間の並列性だけを抽出する粗粒度並列処理において並列処理効果の出るプロセッサ台数の限度を越えた場合でも、提案する階層的並列処理手法により、より高い並列効果が得られることが確認できた。

今後は、より大規模な系統に適用し提案する階層的並列化手法の性能を評価していく予定である。

参考文献

[1] Daniel J. Tylavsky : "Parallel Processing in Power Systems Computation", IEEE Transaction on Power Systems, Vol.7, No.2, May 1992

[2] 小田, 大山: "並列処理とパイプライン処理による電力系統解析計算の高速化", 電学論 B, Vol.115, No.5, pp.472-478, 1995

[3] J. Fong, C. Pottle : "PARALLEL PROCESSING OF POWER SYSTEM ANALYSIS PROBLEMS VIA SIMPLE PARALLEL MICROCOMPUTER STRUCTURES", IEEE Transaction on Power Apparatus and Systems, Vol PAS-97, No.5 Sep 1978

[4] 関根: "電力系統過渡解析論", オーム社, 1984

[5] H.Kasahara, H.Honda, S.Narita, "Parallel Processing of Near Fine Grain Tasks Using Scheduling on OSCAR(Optimally Scheduled Advanced Multiprocessor)", Proc. IEEE ACM. Conf. Supercomputing'90, pp.856-864, 1990

[6] 笠原, 本多, 橋本: "OSCAR のアーキテクチャ", 信学論 D, Vol.38, No.1, pp.62-81, Jan 1989