研究論文

934. 原子力プラント状態予測シミュレータへの
並列処理の適用佐々木和則*, 神余浩夫**,
笠原博徳**, 成田誠之助**

(1989年 7月7日 受理), (1989年 11月9日 再受理)

*Application of Parallel Processing to PWR Plant
Predictive Simulator*By Kazunori SASAKI, Hiroo KANAMARU,
Hironori KASAHARA and Seinosuke NARITA

A parallelism analysis integrated system(PARIS) with an MIMD multiprocessor scheme has been developed to analyze a simulation program which can execute anomaly effect prediction in nuclear power plants and to make the simulation program for parallel processing. The PARIS system analyzes task parallelism and processing time of each task when a user divides a program developed for a single processor into many tasks that are the elementary assignment units. Next, this system assigns tasks to processors using the static scheduling algorithm CP/MISF to minimize the overall processing time, and makes a parallel execution program which can be executed with a tightly coupled multiprocessor.

The PARIS system was assessed by nuclear power plant analyzer code (NUPAC-1) consisting of a reactor coolant system model, a steam generator model and so forth. The results showed that processing time of NUPAC-1 with 7 processors was 3.5 times as fast as with a single processor, and fast running capability was 4.8 times as fast as the real physical time even in accident analyses such as small LOCA. Furthermore, the results showed the PARIS system with the MIMD multiprocessor scheme on the tightly coupled multiprocessor to be efficient for realizing the predictive simulator.

KEYWORDS: *PWR type reactors, power plants, simulators, operator support system, parallel processing, tightly coupled multiprocessor, multi instruction multi data, scheduling algorithm, prediction, simulation codes, thermal hydraulic model*

1. 緒 言

原子力発電プラントに異常が発生した場合、運転員は多くのプロセス量を監視して異常原因を同定し、復旧操作を決定・遂行する必要があり、その時の運転員の負担は大きい。そのためプラントの特性や運転方法等の知識に基づいて、運転員の意志決定を支援する運転支援システムが必要であると指摘され⁽¹⁾、我が国においても知識工学を適用した原子力プラントマンマシンシステムの開発が実施されている⁽²⁾。

この原子力プラントマンマシンシステムでは、異常の進展状況、あるいは適切な運転操作量等の情報を提供するために、プラント状態を高速に推定・予測する状態予測シミュレータの開発が必要である。この状態

* 三菱電機(株)産業システム研究所(Industrial Electronics & Systems Development Lab., Mitsubishi Elec. Corp.)

**三菱電機(株)中央研究所(Central Res. Lab., Mitsubishi Elec. Corp.)

***早稲田大学理工学部電気工学科(Elec. Eng. Dept., Waseda Univ.)

予測シミュレータを実現するために、大型計算機を原子力プラントのマンマシンシステムに導入することはコストの面から非現実的であり、既存の原子力プラント動特性解析コードを高速処理向けにコード変換するシステムの開発が不可欠である。高速処理のためにベクトル処理向けにコード変換することが考えられるが、原子力プラント動特性解析コードに対するベクトル処理は効果が少ないといわれている⁽³⁾。

一方、原子力プラント動特性等の自然現象は、偏微分方程式や常微分方程式でモデル化でき、並列に計算できることが知られている⁽⁴⁾。そこで、原子力プラントを構成する機器の動特性モデルを並列に実行させることにより、原子力プラント動特性解析コードを高速処理することが期待できる。そのためには、マルチプロセッサが通信を行いながら異なった機器モデルの計算を実行できるように、単一プロセッサ用プログラムを並列処理向けプログラムに変換するシステムが必要である。

本研究では、マルチプロセッサが通信を行いながら異なった処理を行うMIMD(Multi-Instruction Multi-Data Stream)並列処理方式⁽⁵⁾に注目し、その処理方式向けにプログラムを変換する方法を中心に研究した。MIMD方式ではオリジナルプログラムを並列処理単位のタスクに分割して、オリジナルプログラムと同一計算結果が得られ、かつプログラム全体の実行時間が最小となるようにタスクをマルチプロセッサに対して割り当てなければならない。しかし、演算素子レベル、ステートメントレベルでの処理を、自動的に各プロセッサに割り当てて並列処理させるための並列コンパイラは未だ研究開発段階にある。また、FORTRANで数万行にも及ぶシミュレーションコードを並列処理可能のように手作業でコード変換するのは多大の労力を要する。

そこで、対象問題に精通したユーザー(オリジナルプログラムの開発者)がプログラムの構造・内容を考慮し、数10行単位でかつ1つの機能のブロックごとにタスク分割すると、並列に処理可能なタスク群と、逐次的に処理しなければならないタスク群とを解析する(並列性解析と呼ぶ)と同時に各タスクの処理時間を測定し、プログラム全体の処理時間が最小となるようにタスクのスケジューリングを行い、マルチプロセッサで処理可能な並列実行プログラムに変換するための並列処理統合システム(PARIS; Parallelism Analysis Integrated System)を開発した⁽⁶⁾⁽⁷⁾。

本報では、特にタスクの並列性解析、スケジューリ

ング、同期等のオーバーヘッド時間が小さい並列実行プログラムの生成が短時間で効率よくできる方式について検討した。次に、PARISを適用して既存のPWR動特性解析コードNUPAC-1(Nuclear Power Plant Analyzer Code)⁽⁸⁾を、密結型マルチプロセッサ計算機S27(Symmetry 27; Sequent社製)上で並列処理可能な並列実行プログラムに変換し、プロセッサ台数を变化させてその並列実行プログラムの処理速度を評価した。最後に、ベクトル計算機を用いた処理速度との比較も含め、PARISの状態予測シミュレータへの適用性を検討した。

Ⅱ. 並列処理統合システム(PARIS)の概要

対象問題に精通したユーザーがシングルプロセッサ向けに開発されたプログラムのタスク分割指示と、プログラム実行入力データの入力指示だけを行うと、タスクの並列性を解析し、プログラム全体の処理時間が最小となるようにタスクをプロセッサに割り当て、S27計算機上で実行可能な並列実行プログラムに変換するPARISを開発した。システム開発に当たっては、タスクのプロセッサへの割当てを決定するための処理時間が短く、生成された並列実行プログラムを実行する際の同期等のオーバーヘッドが小さくなるように設計した。また、性能評価支援ツールにより各種情報をわかり易く提供し、処理時間の短い並列処理向けプログラムを効率よく生成できるように設計した。そのシステム構成を第1図に示す。オリジナルプログラムのタスク分割指示と、分割されたタスクの負荷測定とを実施するために必要なプログラム実行入力データ指示に基づいて、PARISは以下の処理を実施する。

並列解析部において、分割されたタスク間のデータ依存、制御依存(GO TO文等の制御文)の先行制約を調べて、逐次的に処理しなければならないタスク群と、並列に処理できるタスク群を解析する。このとき並行して、タスク負荷測定部において各タスクの処理時間を測定する。次に、タスクの先行制約と各タスクの処理時間から、プログラム全体の処理時間が最小となるようにマルチプロセッサに対してタスクを割り当てる。最後に、プロセッサがこのスケジューリング結果通りにタスクを処理できるように、同期ルーチンなどの実行制御ルーチンをプログラムに挿入して並列実行プログラムを生成する。

得られた並列実行プログラムが十分な処理速度に達していなければ、ユーザーは並列処理による効果をさらに引き出すように、並列化プログラムのタスク分割

第1図 並列処理統合システム(PARIS)の構成

指示を修正する並列化チューニングを行う。並列化チューニングを効果的に行うには、タスク先行制約やスケジューリング結果など、並列性を評価する情報が必要となる。PARISはエンジニアリング・ワークステーション(EWS)のグラフィック機能と、マルチウィンドウ機能とを用いた性能評価支援ツールにより、これらの情報を図式化して表示し並列化チューニングを支援する。

Ⅲ. マルチプロセッサ・アーキテクチャ

各プロセッサの処理内容、通信データの種類およびデータ通信先が異なる場合、各プロセッサがデータ通信制御機構を持つ疎結合システムでは、データ通信制御機構が複雑になり、その処理に要する時間が大きくなる⁽⁸⁾。そこでPARISの開発では、各プロセッサが同一のデータに自由にアクセスできるように、プロセッサと共用メモリーがシステムバスを介して密に結合されたマルチプロセッサ計算機S27を採用した。そのハードウェアシステム構成を第2図に示す。S27はIntel社製の32 bitプロセッサ80386、浮動小数点演算装置80387を採用している。

第2図 並列計算機システムの構成

密結合方式においては、データに自由にアクセスすることはできるが、複数のプロセッサが同時にデータにアクセスしようとするとデータバスの奪い合い(バス競合)が生じ、1つのプロセッサのデータアクセスが終了するまで他のプロセッサは待たされる。そこで、データ転送を高速にすることによりバス競合の発生をできるだけ少なくするように、S27では80 Mbyte/sの高速バスを採用している。さらにバスの負担を軽減するため、各プロセッサは64 kbyteのキャッシュメモリーを装備している。また、ユーザーインタ

フェースとしてEWSのME1300をイーサネット経由で接続し、ユーザーがすべての操作をME1300から行える構成とした。

Ⅳ. 並列化方式

オリジナルプログラムから並列実行プログラムを生成する方法を第1図に従って詳細に述べる。

1. タスク分割

MIMD処理方式では、タスクグラニュラリティ(粒度)をどの程度の大きさに設定するかが重要な問題と

するため、分岐先はタスクの先頭であることが必要である。分岐先がタスクの途中に現れるような不正なタスク分割があれば、次の並列性解析部においてエラーとして検出する。タスク分割が実施されると、システムがプログラムの上流(実行順)から順にタスク番号を割り付ける。

2. 並列性解析

生成された並列実行プログラムがオリジナルプログラムと同一解を得るように、並列化プログラムを入力して、以下のデータ依存(データフロー)先行制約と制御依存(制御フロー)先行制約を調べてタスクの並列性を解析する。タスク相互間に制御フローとデータフローのいずれの先行制約もないタスク集合が並列に実行可能である。

(1) データフロー解析

ある変数に対して書き込み動作を行うと、その変数その後で参照する場合、書き込み以前の値と異なった値となっているため、書き込み変数を持つタスクは、それ以降の同一参照変数をもつタスクと並列に処理することはできない。そこでデータフロー解析では、各タスクに現れる共通データ領域の全変数(COMMON変数)に対して参照/書き込み動作を調べ、書き込みの変数を持つタスクに関してその変数を参照するタスク群を整理する。またデータフロー解析において、複数のタスクで定義・参照され、かつ共通データ領域に登録されていない変数については、PARISが自動的に変数を共通データ領域に追加し、COMMON変数の処理と同一処理を行う。

(2) 制御フロー解析

プログラム中にIF文、GO TO文等の分岐命令があると、分岐の判定によってその後の実行位置が異なる。すなわち、分岐命令を持つタスクと分岐先のタスクとの間には先行制約が存在し、分岐命令を持つタスクが終了しなければ、それ以降の分岐先タスクまでのタスクは実行できない。そこで、制御フロー解析ではプログラム制御文を調べ、分岐命令を持つタスクに関して分岐先タスクまでのタスク群を整理する。

データフロー解析と制御フロー解析が終了すると、2つの先行制約の和事象として、タスクをノード、先行制約を有向バスとしたタスクグラフに表わす。第3図に示す10個のタスクの並列性解析結果から得られるタスクグラフの例を第4図に示す。図中ノード内の数字はタスク番号を示す。矢印が先行制約を示し、→印の方向に沿ってタスクを処理しなければならないことを示す。

第4図 タスクグラフ例

このタスクグラフのバスの中には、タスク間の実行順序関係を規定する上で不必要な、すなわち存在しなくてもタスク実行順序関係が保存される冗長バスが存在する。冗長バスを含んだタスクグラフからタスクのスケジューリングを行うのは、処理時間がかかり非効率的である。あるノードから別のノードへの複数のバスを通過して到達するバスがあれば、2つのノード間の直接バスは冗長バスとして除去できる。そこで、スケジューリング以前に、[付録]に示すアルゴリズムでタスクグラフの冗長バスを除去する。第4図のタスクグラフから冗長バスを除去したタスクグラフを第5図に示す。図において、タスク番号④から⑦、⑧へのバス、タスク番号⑥から⑧、⑨へのバスが冗長バスとして除去されている。ノード横の数字は第IV-3節に述べる負荷測定結果であるタスクの処理時間を示し、()内の数字は第IV-4節のスケジューリングで述べるタスクレベルを示す。

第5図 冗長バス除去後のタスクグラフ

3. 負荷測定

プログラム実行前にタスクのスケジューリング(スタティック・スケジューリングと呼ぶ)を行うためには、各タスクの処理時間をあらかじめ測定しておく必要がある。そこで、並列性解析と並行して、並列化プログラムとそのプログラムの実行用入力データを用いて各タスクの負荷測定を行う。システムタイマーによる負荷測定では、最小測定単位が10msのため、タスクグラニュラリティの細かいタスクの負荷測定を、スケジューリングに必要なmsの単位で測定できない。そこで本システムでは、2台のプロセッサを用いて以下のようにタスク負荷を測定する。1台のプロセッサでプログラムを実行し、全タスクの処理時間を測定する。その間、もう1台のプロセッサは実行時間が明確な加算命令を実行し、各タスクの実行開始時と終了時のカウント数を記憶する。全タスクの処理時間を各タスクの処理に要したカウント数で比例配分して各タスクの負荷を求め、各タスクに対する処理時間を示すタスク負荷リストを作成する。

4. スケジューリング

タスクグラフとタスク負荷リストから、プログラム全体の実行時間が最小となるようにタスクをマルチプロセッサに割り当てる。プログラム実行時にタスクをマルチプロセッサに割り当てるダイナミック・スケジューリング方式では、スケジューリングの処理によるオーバーヘッドが大きくなるため、プログラム実行前にタスク割当てを決定するスタティック・スケジューリング方式を採用する。PARISのスケジューリング手法としては、スタティック・スケジューリング・アルゴリズムCP/MISF(Critical Path/Most Immediate Successors First)法⁽⁹⁾⁽¹⁰⁾を適用し、プログラム実行時のオーバーヘッドが小さくなるように、タスク・スケジューリング時のタスク優先順位決定法を変更した。

このCP/MISF法は、まず最初にタスクグラフの出口ノードからタスク*i*に対応するノード*N_i*までの最長のパス長*l_i*を以下の式で計算する。

$$l_i = \max_k \sum_{i \in \phi_k} t_k \quad (1)$$

ここで、 ϕ_k : 出口ノードから*N_i*へ至る*k*番目のパス

t_k: タスクの処理時間

この*l_i*をノード*N_i*のレベルと呼び、第5図の()内の数値がレベルを示す。次に、プロセッサへのタスク割当てを実施する際の優先順位付けを、このレベルの大きい順に実施する。同じレベルのノードがある場合

には、そのノードが終了しなければ実行できない後続タスク数の多い方から高い優先順位をつける。その後、先行制約を満たし、優先順位の高いタスクから順次待ち状態のプロセッサに割り当て、スケジューリンググラフを作成する。第5図のタスクグラフを例題として、2台のプロセッサでスケジューリングしたときのスケジューリンググラフ例を第6図に示す。

第6図 CP/MISF法によるスケジューリンググラフ

このスケジューリングでは、まず最もレベルの高いタスク1をプロセッサ1に割り当てる。他のタスクはタスク1が終了するまで実行できないためプロセッサ2は待機状態となる。タスク1が終了すると、先行制約を満たしレベルの高いタスク2をプロセッサ1に割り当て、次にレベルの高いタスク4をプロセッサ2に割り当てる。タスク2が終了すると、タスク6の先行制約が満たされ、レベルの高いタスク6をタスク3より先にプロセッサ1に割り当てる。以下同様に、タスクが終了した時点で先行制約を満たし、レベルの高いタスクからプロセッサに割り当てる。

しかしCP/MISF法の優先順位決定法では、1個のプロセッサが処理するタスク群を考えた場合、タスク番号の小さい順(プログラム上流からのタスク順)に処理するスケジューリング結果にならない場合が生じる。このことは、次節で述べる並列実行プログラム生成法を採用した場合に、実行制御によるオーバーヘッドが大きくなることを意味している。これは次に実行すべきタスクを探索する際、まず実行終了タスクの下流側タスク(タスク番号の大きいタスク)を順次探索していき、該当タスクがなければプログラムの先頭に戻り、さらに該当タスクを順次探索していくためである。

そこで、各プロセッサがタスク番号の小さなタスク順に実行するように、タスク割当てを行うたびに、先行制約を満足したタスク群の中で、待ち状態のプロセッサにすでに割り当てられている最後のタスクに近いタスク番号から高い優先順位を持たせる手法(NTF[Nearest Task First]法と呼ぶ)を導入して、スケジューリング手法を変更した。第5図のタスクグラフを例題として、2台のプロセッサでNTF法を導入してタ

スクをスケジューリングしたときの結果を第7図に示す。各プロセッサの処理するタスクを見た場合、タスク番号の小さいタスクから処理するようにスケジューリングされている。

第7図 NTF法を導入した場合のスケジューリンググラフ

5. 並列実行プログラム生成法

スケジューリング結果通りにタスクが実行されるように、スケジューリンググラフからマルチプロセッサで並列処理可能な並列実行プログラムを生成する方式として、以下の3つの方法が考えられる¹⁰⁾。

- (1) 各タスクをそれぞれ1個のプロセス(並列実行プログラム)として管理する。
- (2) タスクをスケジューリング結果順に統合して、各プロセッサごとに並列実行プログラムを生成する。
- (3) 並列ディレクティブを実行制御ルーチンに置換して、プロセッサ共通の並列実行プログラムを生成する。

(1)の方式では、スケジューリング結果に従ってプロセスを動的に発生・消滅させるため、その1回の動作に数10msのオーバーヘッドがかかり、トータルとしての処理時間が大きくなる。また(2)の方式では、原子力プラント動特性解析コードによる冷却材喪失事故(LOCA; Loss of Coolant Accident)、蒸気発生器細管漏洩事故(SGTR; Steam Generator Tube Rupture)等の事故解析のように、プログラムの処理ルーチンあるいはタスクの処理時間が変化した場合でも最適スケジューリングを行おうとすると、あらかじめ事故解析に応じて並列実行プログラムを各プロセッサに対して生成しておく必要がある。そこで、各プロセッサがスケジューリング結果を参照し、各タスクの実行処理を制御することにより、1個の並列実行プログラムで各種事故解析にも対応できる(3)の並列実行プログラム生成方式を採用した。

並列実行プログラム生成において、タスクがスケジューリング通りに並列に実行され、シングルプロセッサでの解析と同一の結果を得るためには、プロセッサ間でタスクの処理状態について通信を行い協調動作を

行う必要がある。この通信は共用メモリー上のタスク実行管理テーブルTEMT(Task Execution Management Table)を介して行う。TEMTは各プロセッサが次に実行すべきタスクのタスク番号とそのタスクを実行するための先行制約が満足されているかの情報を持つ。

協調動作には、同期機構と条件分岐が発生したときの分岐先の整合がある。同期機構については、TEMTを参照して次に実行すべきタスクの先行制約が満足されていなければ、実行制御ルーチンがプロセッサをビジーウェイト状態にすることによりプロセッサ間の同期をとる。また分岐先の整合については、タスクで分岐が発生すれば、TEMTを操作して他のプロセッサが次に実行すべきタスクを変更する。

ユーザーがタスク分割指示命令として与えた並列ディレクティブを、このような処理を行う実行制御ルーチンに置換し、マルチプロセッサ計算機で処理可能な並列実行プログラムを生成する。

6. 並列化チューニング

PARISにより生成された並列実行プログラムが十分な処理速度に達していなければ、並列ディレクティブ指示を変更して並列実行プログラムを再生成できる。タスク分割指示から並列性評価の過程を繰り返す並列化チューニングには、並列率(プログラムの中で並列に処理できる部分が占める割合)、並列度(同時に並列処理できるタスク数)などの情報が必要となる。効率よく並列化チューニング作業を進めるには、タスクグラフやスケジューリンググラフ等の情報を図式化して1枚の画面に表示することが望ましい。

そこで、PARISはマルチウィンドウ形式のグラフィックで並列化チューニングに必要な情報を表示する。タスクグラフにより並列度を、スケジューリンググラフによりプロセッサの待機時間および並列率を評価し、プログラム全体の処理時間が短くなるように、ソースプログラム・エディタによりタスク分割を変更する際の表示画面例を第8図に示す。

V. 並列処理手法の評価

本章では、PWR動特性解析コードNUPAC-1を例題として、PARISを用いた場合の処理速度およびPARISのプラント状態予測シミュレータへの適用性を評価する。

1. PWR動特性解析コードNUPAC-1の概要

NUPAC-1は運転訓練シミュレータおよび運転支援システムのプラント状態予測シミュレータに組み込む

第 8 図 PARIS の並列化チューニング画面

ために開発された動特性解析コードで、小 LOCA 時のように 1 次冷却材系統にボイドが発生し二相状態になる事故を正しく模擬でき、小 LOCA, SGTR 等の事故解析結果が安全解析コード RELAP 5/MOD 2⁽¹¹⁾ の解析結果と一致することが確認されている⁽⁷⁾。このコードは、1 次冷却材系統 (RCS) モデル、蒸気発生器 (SG) モデル、炉心熱特性モデル等から構成されており、さらにこれらのモデルは配管部、伝熱部、ポンプ部等から構成されている。

NUPAC-1 は配管部、伝熱部、ポンプ部等のノードと、それらを結ぶジャンクションとからなるノードジャンクション法を採用している。NUPAC-1 のノードジャンクション図を第 9 図に示す。RCS の分割数が 24, SG の分割数が 4 である。基礎式としては、加圧器, SG ドーム部では熱非平衡モデルを採用しているため、それらのノードでは液相, 気相の質量およびエネルギー保存式を立てるが、その他のノードでは熱平衡モデルを採用しているため、液相, 気相混合として以下の式を立てる。

混合質量保存式

$$V \partial \rho / \partial t = -\nabla (A \rho u) \Delta x + W_0 \quad (2)$$

混合エネルギー保存式

$$V \partial (\rho e) / \partial t = -\nabla (A \rho u e) \Delta x + V (\partial P / \partial t) + W_0 e_0 + q \quad (3)$$

また、各ジャンクションにおいては均質二相流として以下の運動量保存式を立てる。

混合運動量保存式

$$V \rho / A (\partial u / \partial t) = -\rho / 2 \cdot \nabla u^2 \Delta x - \nabla P - K / 2 \cdot \rho u |u| - \rho g h + H \quad (4)$$

各ノード、ジャンクションで立てられた保存式を、空間差分としてコントロールボリュームジャンクション法⁽¹²⁾、時間差分としてセミインプリシット法⁽¹³⁾で以下のように離散化を行い、各ノードにおける圧力の連立方程式としスカイライン法⁽¹⁴⁾により各部の圧力、エンタルピー、流量を求める。

混合質量保存式

$$V_i / \Delta t (\rho_i^{n+1} - \rho_i^n) = \sum_{k=i \pm 1/2} (A_k \rho_k^n u_k^{n+1}) + W_0^n \quad (5)$$

混合エネルギー保存式

$$V_i / \Delta t \{ (\rho e)_i^{n+1} - (\rho e)_i^n \} = \sum \{ A_k (\rho e)_k^n u_k^{n+1} \} + V / \Delta t (P_i^{n+1} - P_i^n) + (W_0 e_0)^n + q^n \quad (6)$$

混合運動量保存式

$$I \rho_k^n / \Delta t (u_k^{n+1} - u_k^n) = \rho_k^n / 2 \cdot \{ (u_{i-1}^n)^2 - (u_i^n)^2 \} + P_{i-1}^{n+1} - P_i^{n+1} - k / 2 \cdot \rho_k^n u_k^{n+1} |u_k^n| - (\rho h)_k^n g + H^n \quad (7)$$

上式において、 Δt が時間きざみ幅で、 n が時間ステップを表わし、 $\sum$ はノードボリュームに接続されたジャンクション数の和を表わす。また、添字 i はノードボリュームを、 k はジャンクションボリュームを表わす。本プログラムは n ステップの状態量を用いて $n+1$ ステップの状態量を求めるもので、(5)~(7)式にお

第9図 NUPAC-1のノード分割構成図

ける各項のうち n ステップの状態量を用いて計算する部分(既知項と呼ぶ)が、第9図に示す各ノードあるいは各ジャンクションごとに並列に計算可能である。SGプログラムの概略のフローを、第10図に示すSGモデルのタスクグラフに示す。

2. 並列実行結果

まず、NUPAC-1の中のSGプログラムを用いて並列処理による高速性を評価する。SGプログラムはノード数が少ないため、物性値計算、熱伝達率計算、既知項計算等の機能ブロックを、さらに各ノードあるいはジャンクションごとの78タスクに分割し、PARISにより並列実行プログラムを生成した。その際のタスクグラフを第10図に示す。このタスクグラフから最大の並列度はドーム部熱伝達率計算部、運動量保存式の既知項計算部で5となるが、4分割のSGモデルは既知項計算を各ノード、ジャンクションで実施するため、並列度 N は4の占める割合の方が多い。

また、ノード圧力に関する連立方程式解法のスカイライン解法プログラムは並列化していないため、並列率 R は第10図におけるSGモデル全体の処理時間に対する既知項統合部までの処理時間の割合となり、タスク負荷測定結果から0.8であった。スカイライン解法

プログラムを並列化していないのは、タスクグラニュラリティを細かくすることにより並列性は高まるが、逆に通信・同期のオーバーヘッドの方が大きくなるためである。

ここで、SGプログラムを78タスクに分割した場合の並列処理性能の目安となる概略並列性能 P (オリジナルプログラムに対する計算速度比)を次式で定義する。

$$P = (1.0 - R) + (R/N) \quad (8)$$

上式において、右辺第1項は並列処理不可能な部分で、並列性能においてボトルネックとなる部分である。この項があることにより、並列度が十分大きくても並列性能は $(1-R)$ より改善できない。第2項は処理性能において並列処理可能な部分が並列度に依存する部分である。SGプログラムの場合、 $R=0.8$ 、 $N=4$ として評価すると、ボトルネックになる割合が0.2で、概略並列性能は0.4となる。

SGの並列化プログラムに対してプロセッサ数を変化させてスケジューリングを行い、その結果に基づき生成した並列実行プログラムをS27計算機で並列処理させた。S27計算機はプロセッサを8台装備するが、うち1台は実行時間測定とオペレーティングシステム

第12図 SGモデルのスケジューリングリスト例

ヘッドと、同期を含む並列実行制御ルーチン処理によるオーバーヘッドがある。スケジューリング結果による並列性能と実測並列性能との差0.05のうち、0.02相当がバス競合によるオーバーヘッドであることを確認しており、残りの0.03が実行制御ルーチン処理によるオーバーヘッドである。

並列実行プログラム生成方式としての第Ⅳ-5節の(3)方式を採用したが、この実行制御ルーチン処理によるオーバーヘッドを時間に直すと約3msと小さく、この実行制御のオーバーヘッドが、RARISをプラント状態予測シミュレータに適用するに当たって問題となる時間ではないと考える。

次にノード数24のRCSプログラムについて、各ノードジャンクションで立てる保存式の既知項計算部を並列処理可能なように、S27計算機に装備されているプロセッサ数8で空間的に分割し、トータル96タスクで並列処理した。その実測並列性能を第13図に示す。並列性能はプロセッサ4台で0.51、7台で0.42であった。RCSモデルはノード数が本試験で用いた最大プロセッサ数よりも多いため、プロセッサ数を増加させると並列性能は向上するが、プロセッサ5台以降はそれほど著しく向上しない。これはスカイライン解法部分を含むタスクの並列性能におけるボトルネックになる割合がほぼ0.30と大きいためである。

第13図 機器単体モデルおよびNUPAC-1の並列化効果

最後に、NUPAC-1全体の並列処理を行い並列性能を評価した。NUPAC-1は、RCS、SG、炉心熱特性、ポンプ、伝熱量計算部等のプログラムに分類され、第14図に示すようなプログラムフローになる。図中の→印はタスクグラフと同様の先行制約を表わす。この図からRCSとSGを並列処理できることがわかる。このプログラムの中で、炉心熱特性、ポンプ、伝熱量計算部の処理時間はNUPAC-1全体の処理時間の約2%と小さいため、これらに割り当てるプロセッサは1台として固定し、RCS、SGに割り当てるプロセッサ台数を変化さ

せて並列処理を行なった。その時の実測並列性能も第13図に示す。プロセッサ7台を使用した時の並列性能は0.29であった。この値は、RCS、SGの各単体モデルを同数のプロセッサで処理したときの並列性能を大きく上回っている。これは異なる計算モデルを並列処理できるMIMD処理方式が、原子力プラント動特性コードの高速化に適していることを示している。

第14図 NUPAC-1 のプログラムフロー

NUPAC-1は並列化により、定常解析およびタイムステップが一定値で解析できる手動トリップ等の過渡解析については、実現象の5.4倍の高速性が達成された。またRCS内が二相状態になる2in.破断小LOCA事故解

析においても、自動的にタイムステップを小さくするため処理時間がかかるが、RCS内の状態変化が最も厳しい部分の計算でも実現象の4.8倍の速度で解析できることを確認した。PARISを適用してNUPAC-1を並列処理することにより、原子力プラントに異常が発生した場合のプラントの内部状態が詳細に、かつ数10min後の事故進展状況が実現象より速く得られる。動特性解析コードの高速処理を可能とするPARISを運転支援システムの状態予測シミュレータに適用すれば、運転員はこれらの情報を用いて適切な復旧操作を決定することができると考える。

Ⅶ. ベクトル計算機を用いた処理速度の評価

ベクトル計算機を使用した場合、RELAP5/MOD2は数10人・月のベクトル化作業を行えば、3.0～3.5倍の処理性能が得られるという報告がある⁽¹⁵⁾。そこで、NUPAC-1を並列処理する場合とベクトル処理する場合の計算速度に与えるそれぞれの処理効果を評価するために、NUPAC-1をベクトル化の手作業なしでベクトル計算機に準備されているベクトルコンパイラ(以下V.C.)を使用して処理させた場合、どの程度の高速性が得られるか評価する。

ベクトル計算はFACOM M-780 VP-400 Eを用い、SGモデル単体、RCSモデル単体、およびNUPAC-1全体についてそれぞれ高速性を評価した。V.C.を使用しない場合の処理速度に対するV.C.使用時の速度向上率を第2表(1)に示す。

第2表 ベクトル計算機での処理速度向上率

(1) オリジナルノード数の場合 (SG: 4ノード, RCS: 24ノード)				(2) ノード数を増加した場合 (SG: 10ノード, RCS: 48ノード)		
	SGモデル	RCSモデル	NUPAC-1全体	SGモデル	RCSモデル	NUPAC-1全体
速度向上率	0.95	0.85	0.90	0.98	0.88	0.92

この結果からV.C.を使用すると、いずれの場合もV.C.を使用しない場合に比較して処理速度が遅くなっている。これはNUPAC-1プログラムのベクトル長(ノード数)が短いことに加え、以下の理由によりNUPAC-1コードがベクトル化できない部分が多いためと考える。

- (1) 機器・構造の差異、各ノードで使用する実験式・データの差異により、各ノードの計算が全く同一処理にならないこと。
- (2) 各種事故模擬を考慮しているため、IF文によ

る分岐が多いこと。

次に、ベクトル長を長くした場合の処理性能を評価するために、SG、RCSモデルのノード数をそれぞれ4から10、24から48に変更して同様の評価を行なった。その結果を第2表(2)に示す。ノード数を増加することにより処理速度向上率は若干改善されたが、V.C.を使用しない場合よりもまだ処理速度が遅く、RELAP5/MOD2ほどの効果は得られなかった。RELAP5/MOD2で3.0～3.5倍の処理性能が得られたのは、RELAP5/MOD2コードがベクトル処理向きで、かつ

ベクトル化作業の効果があったこと、RELAP 5/MOD 2が安全解析コードであることから解析で使ったモデルのノード数が多いことが考えられる。

一方、PARISを用いた並列処理方式では、DOループ内に上記(1),(2)のような部分が多く存在しても、loop body(モデルのノードに対応する)ごとの並列計算、DOループ内の機能ごとの並列計算が可能であり、処理の高速化ができる。第13図および第2表から、特にノード数の少ないNUPAC-1は並列処理による高速化が適しているといえる。

Ⅶ. 結 言

ユーザーがシングルプロセッサ向けに開発されたプログラムのタスク分割指示とプログラム実行用入力データの入力指示だけを行うと、タスクの並列性解析、負荷測定を行いプログラム全体の処理時間が最小となるように、タスク優先度決定法にNTF法を適用したCP/MISF法でタスクをプロセッサに割り当て、並列実行プログラムを生成する並列処理統合システムPARISを開発した。

このPARISを用いて、二相状態が模擬できる原子力プラント動特性解析コードNUPAC-1を例題として並列処理し、その並列処理性能を評価した。PARISを用いてNUPAC-1全体を7台のプロセッサで実行した時の処理速度は、1台のプロセッサで実行した時の3.5倍であり、定常解析、過渡解析においては実現象の5.4倍、事故解析においても実現象の4.8倍以上の速度で実行できることを確認した。事故時の原子力プラント内部状態推定シミュレータに適用できる高速性を得ることができたと考える。

異常の進展予測、運転操作量の決定を行うためには、さらに高速処理が要求されることが考えられるが、現在すでにプロセッサの高速化が実施されていること、並列向きにアルゴリズムを書き直すことにより並列率の向上が図れることを考えると、密結合型マルチプロセッサ計算機とPARISの並列処理方式の結合が、プラント状態予測シミュレータ実現にも有望な手段と考える。

また、ベクトル計算機を用いた場合の処理速度向上率を評価した結果、ノード数の少ないNUPAC-1では処理速度の向上が見られなかった。マルチプロセッサ計算機がベクトル計算機に比して廉価であることから、コストパフォーマンスの面からも予測シミュレータの実現には、密結合型マルチプロセッサ計算機とPARISの並列処理方式の結合が有望な手段と考える。

本研究では、プログラム実行前にタスクをプロセッサに割り当てるスタティック・スケジューリング方式について検討し、NUPAC-1を例題として高速性およびプラント状態予測用高速シミュレータへの適用性を評価した。今後は、PARISの他コードへの適用を実施して高速性を評価するとともに、事故解析のようにタスクの負荷が変動する場合にも、最適スケジューリングを行えるように、プログラム実行時にタスクをプロセッサに割り当てるダイナミック・スケジューリング方式の検討と、その方式のプラント状態予測シミュレータへの適用性を検討していく予定である。

〔記 号〕

A : 流路断面積, 伝熱面積	(m^2)
V : 体 積	(m^3)
g : 重力加速度	(m/s)
h : エレベーション, 熱伝達係数	($\text{m}, \text{J/kg} \cdot \text{m}^2 \cdot \text{K}$)
l : 流路長	(m)
q : 伝熱量	(J)
W_0 : 他系統からの流量	(kg/s)
e : エンタルピー	(J/kg)
e_0 : 他系統からの流入・流出エンタルピー	(J/kg)
P : 圧 力	(Pa)
u : 流 速	(m/s)
ρ : 密 度	(kg/m^3)
H : ポンプヘッド	(m)
Δt : タイムステップ	(s)

本研究は、通産省補助金「原子力発電支援システムの開発」の一環として行われた。研究を進めるに当たり、京都大学原子エネルギー研究所の若林二郎教授、吉川栄和助教授、五福明夫助手には、ベクトル計算機による計算実施に関して多大のご尽力を賜わり、また(株)アイセルの小林俊雄、三菱電機(株)の田中光雄、荻野敬迪の諸氏から貴重なご助言、ご協力を頂きました。記して、ここに感謝の意を表します。

— 参 考 文 献 —

- (1) Safety Data Integration Group: Functional criteria for emergency response facilities, NUREG-0696, (1981).
- (2) 梶田征也, 田中光雄, 荻野敬迪, 北村哲男, 増井隆雄: 原子力発電プラントマンマシンシステム, 三菱電機技報, 63 [7], 65~68 (1986).
- (3) ISHIGURO, M.: Japanese experiment in vectorizing thermohydraulic reactor codes, Trans. Am. Nucl. Soc., 55, 328 (1987).
- (4) 福井義成: 汎用マルチプロセッサシステムによる並列処理, 情報処理, 28[11], 1469~1475 (1987).
- (5) 神余浩夫, 佐々木和則, 笠原博徳, 成田誠之助: PWR予測シミュレーションシステムの開発, 原子力学会「昭63年会」要旨集, A6.

- (6) 佐々木和則, 神余浩夫, 笠原博徳, 成田誠之助: 原子力プラント状態予測シミュレータへの並列処理の適用, 並列処理シンポジウム JSPP' 89 論文集, pp. 313~320(1989).
- (7) SASAKI, K., TERASHITA, N., OGINO, T.: The development of the fast-running simulation pressurized water reactor plant analyzer code(NUPAC-1), *Nucl. Technol.*, 85, 259~273 (1989).
- (8) PAKER, Y. (渡辺豊英, 桶谷猪久夫, 小澤義明共訳): “マルチマイクロプロセッサ・システム”, (1984), 啓学出版.
- (9) 笠原博徳, 成田誠之助: マルチプロセッサスケジューリング問題に関する実用的な最適および近似アルゴリズム, 信学論, J 67-D(7), 792~799 (1984).
- (10) KASAHARA, H., NARITA, S.: Parallel processing of robot-arm control computation on a multimicroprocessor system, *IEEE J. Robotics Autom.*, RA-1(2), 104~113 (1985).
- (11) RANSOM, V.H., WABNER, R.J.: RELAP 5/MOD 2 code manual, EGG-SAAM-6377, (1984).
- (12) KATSUMA, K.R.: RELAP 4/MOD 5 User's manual, ANCR-NUREG-1335, (1976).
- (13) LILES, D.R., REED, W.H.: A semi-implicit method for two-phase fluid dynamics, *J. Comput. Phys.*, 26, 390~407 (1987).
- (14) 山田嘉昭, 横内康人: “有限要素法による弾塑性解析プログラミング: EPIC-5 解説”, (1981), 培風館.
- (15) 篠沢尚久, 近藤一也, 和田善之, 石黒美佐子: 軽水炉安全性解析コードRELAP 5/MOD 2/CYCLE 36 のベクトル化, *JAERI-M* 86-019, (1986).
- (16) BUSACKER, R.G., SAATY, T.L. (矢野健太郎, 伊理正夫共訳): “グラフ理論とネットワーク/基礎と応用”, (1970), 培風館.
- (17) PETERSON, J.L. (市川あつ信, 小林重信共訳): “ベトリネット入門. 情報システムのモデル化”, (1984), 共立出版.

〔付 録〕

タスクグラフの冗長パス除去アルゴリズム

タスクグラフの冗長パスを「あるノードから2本以上のパスを辿って到達できるノードに対し, 直接到達できるパス」と定義すると, 冗長パス除去問題は, 「すべての可到達性が保存されるパス最少のグラフを求める」問題となる⁽¹⁶⁾. したがって, 以下のアルゴリズムにより冗長パスを検出し除去できる.

- (1) あるノード i から1パスで到達できるノードを列挙し, これを可到達集合 R_{i1} とする(並列性解析結果が R_{i1} となる).
- (2) 同様にノード i から n パスで到達するノードを列挙し, これを可到達集合 R_{in} とする. ここで, タスクグラフはループを持たないので, n はタスク数以下である(可到達集合 R_{in} は R_{i1} の行列表現を用いて解析的に計算可能である⁽¹⁷⁾).
- (3) ノード i から2パス以上の可到達集合 $(R_{i2} \cup R_{i3} \cup \dots \cup R_{in})$ と1パスの可到達集合 R_{i1} の積集合を求める.

$$P_{ir} = (R_{i2} \cup R_{i3} \cup \dots \cup R_{in}) \cap R_{i1}$$

- (4) すべてのノードについて P_{ir} を求める.
- (5) ノード i と P_{ir} 要素ノード間のパスが冗長であるから, そのパスを除去する.

前述のように, 可到達集合 R_{in} は R_{i1} の行列表現を用いて解析的に計算可能であるから, 以上のアルゴリズムはグラフ探索的な冗長パス検出手法⁽¹⁷⁾に比べて処理時間の点で効率的である.

◇(第6回)伝熱を伴う二相流と◇ 関連機器システム

混相流レクチャーシリーズ「基礎から最新線まで」

主 催 日本混相流学会, 協 賛 本会ほか18学協会

会 期 12月12日(水), 13日(木)

会 場 大阪市立大学・文化交流センター
(大阪市北区梅田1-1-3-1700)

○題目・講師○

気液二相流の流動と熱伝達の基礎 (工学院大) 植田辰洋
蒸発器と凝縮器における冷媒の二相流と熱伝達

(九大) 吉田 駿
熱交換器における着層・伝熱問題 (関西大) 勝田勝太郎
圧縮器内部における二相流 (静岡大) 柳沢 正

ヒートポンプにおける二相流の課題と解決法
(三菱電機) 田中直樹

ヒートポンプの熱交換器における二相流と伝熱

(東芝) 橋詰健一
ビル用ヒートポンプシステム (ダイキン) 米本和生
都市ガスによるコージェネレーション (大阪ガス) 竹田隆一
参加費(テキスト1冊含む)

協賛学協会員 15,000円, 非会員 20,000円
学生・院生 4,000円

申込締切 12月1日(土), 定員 100名

申込方法 ハガキに「混相流学会講習会申込み」と題記し, 氏名, 勤務先, 連絡先, 所属学会名を明記して下記までお申し込み下さい。参加費は所定の口座へ郵便振替または銀行振込で。なお, テキストのみご希望の方は4,000円(会員), 6,000円(非会員)を添えて「混相流学会講習会テキスト希望」と題記してお申し込み下さい。

申込先 (〒659) 芦屋市公光町9-7-202 鶴学術出版印刷
日本混相流学会企画運営委員会

(0797-38-3390; FAX 0797-38-3351)

本講習会に関するお問合せは (〒558) 大阪市住吉区杉本3-3-138
大阪市立大学 工学部機械工学科 東 恒雄 (06-605-2666)