# 3T－9　分散処理環境上の仮想並列計算機に対するスケジューリングアルゴリズムの設計とその検討

前富 博*　伊藤 聡†　島崎 眞昭‡
*九州大学工学部　†九州大学工学部(当時)　‡九州大学大型計算機センター

## 1. はじめに

ネットワークの進歩、計算機のダウンサイジングにより、分散処理は計算機処理形態の主流になりつつある。しかし、異種機間分散処理環境での大規模ジョブ処理においては、プロセッサの能力の違いから負荷のアンバランスが生じる為、タスクのサイズに応じたスケジューリングが必要とされる。

分散処理環境の1例であるPVM(Parallel Virtual Machine)上でも例外でなく、この負荷のアンバランス問題が生じている。そこで本研究では、異機種分散環境におけるタスクのスケジューリングに関し、プロセッサの能力の違いを考慮する新しいスケジューリングアルゴリズムを提案し、シミュレーションすることでPVM上での実現の検討を行なった。

## 2. PVM: Parallel Virtual Machine

### 2.1 PVMの概要

PVM(Parallel Virtual Machine)は、さまざまなネットワーク上のシリアル、パラレル、ベクトル計算機などの異機種の計算機の集合を1つの大きな分散メモリ型の計算機として使用することを可能にしたソフトウェアシステムである[1]。

その特徴として、

- C及びFortranのアプリケーションがサポートされており、プログラムが容易に作成できる。
- 1つのPVMで複数のアプリケーションを同時に実行できる
- コンパクトなシステムである

などが挙げられる。

PVMはアプリケーション、機種及びネットワークレベルで異種性をサポートしている。それにより、アプリケーションのタスクがその解決に最も適したアーキテクチャを利用することを可能にしている。

PVMは大きく分けてpvmdとlibpvmの2つの部分から構成されている。pvmd3とは、PVMのデーモンプロセスであり、Virtual Machineを構成するすべての計算機に常駐している。また、libpvmは、PVMのインタフェースルーチンのライブラリであり、メッセージの送受信、プロセスの実行、仮想計算機の再構成などに関するルーチン群を含んでいる。ユーザがPVMのアプリケーションを実行させるためには、ユーザが定義したVirtual Machineを構成する各計算機上でpvmdを動作させなければならなく、さらにlibpvmとリンクされなければならない。

なお、PVMの開発は1989年にOak Ridge National laboratory(ORNL)で開始され、その研究は今なお進行中である。プロジェクトの中心的な人物として、Al Geist、Adam Beguelin、Jack Dongarraらがいる。

### 2.2 PVMにおける問題点

PVMはさまざまなネットワーク上の異機種の計算機を複数台用いることにより、並列計算機に匹敵するほどの処理を行なうことができる環境を提供している。しかし、PVMはその構成によっては、使用している計算機の台数に比べて処理時間があまり短縮されない場合がある。その原因としては、

- 計算機の処理能力の違い
- プロセス間の通信時間

が挙げられる。

このことは、スケジューリング機能を組み込むことによりかなり解決できると思われるが、現在PVMにはそのような機能は組み込まれていない。

従って、PVMの環境下で、より効率の良い分散処理を行なうためには、上記のような問題点を考慮したプロセスの実行方法、すなわちスケジューリングの機能を組み込むことが必要であるといえる。

## 3. 異機種分散環境上でのスケジューリング

以上から、同機種のみならず異機種の分散処理環境で効率の良い分散処理を行なうためには、タスクのスケジューリング機能は必要であると思われる。

そこで、本章では、文献[2]で紹介されている同機種の分散処理環境におけるスケジューリングアルゴリズムから異機種の場合について提案し、そのシミュレーションについて述べる。

---

Task Scheduling Algorithm for Parallel Virtual Machine in Distributed Computing Environment
Hiroshi Maetomi, Akira Itoh, Masaaki Shimasaki*
Faculty of Engineering, Kyushu Univ.
Computer Center, Kyushu Univ.*

### 3.1 問題の定義

本研究のスケジューリング問題とは、処理能力の異なる m 台のプロセッサで n 個のタスクの処理を行ない、n 個のタスクすべての処理が終了するまでの時間を最小とする、タスクとその実行プロセッサの組合せ及び処理順序を決定する問題である。

プロセッサ間の通信は中継無しとし、各プロセッサは処理の実行と通信を同時に行なえるものとする。

### 3.2 スケジューリング・アルゴリズム

以下の手続きを用いてすべてのタスクを順次プロセッサに割り当て、処理を行なう。

- **プレプライオリティの決定 ( Calculating Prepriority )**

各タスクに対して、終了タスクまでの最長パスをそのタスクのプレプライオリティ *pp(i)* とする。*pp(i)* の値が大きいほどそのタスクの優先度が高いとする。

- **割当て可能条件の判定 ( Checking Schedulable Condition )**

割当て可能条件が成立しているとは、時刻 C において、直前タスクがすべて終了したタスク (割当て可能タスク) が存在して、かつどのタスクも割当てされていない空きプロセッサが存在することである。

- **プライオリティの決定 ( Calculating Priority )**

すべての割当て可能タスクとすべての空きプロセッサの組に対して以下の式で定義されるプライオリティ *p(i)* を求める。

$$p(i,j) = pp(i) \times power(j) + cost(i,j)$$

*pp(i)* : プレプライオリティ.
*power(j)* : プロセッサの能力 (単位時間当りの処理量)
*cost(i,j)* : タスク $i$ とプロセッサ $j$ の組を選択したときに不要となる通信コスト.

- **タスクの割当て ( Allocation )**

プライオリティの値が最大となるタスク $t_{max}$ とプロセッサ $p_{max}$ の組を求め、タスク $t_{max}$ をプロセッサ $p_{max}$ に割り当てる。プライオリティの値が最大となるタスクとプロセッサの組が 2 つ以上ある場合には、同一タスクに対する競合, 同一プロセッサに対する競合に注意して割り当てを決定する。

割り当てが決定すれば、プロセッサ $p_{max}$ がタスク $t_{max}$ を完了する時刻 (完了予定時刻) を、現在の時刻 C にタスク $t_{max}$ を実行するために必要な通信時間とタスク $t_{max}$ の処理時間を加えた値とする。

- **タスクの終了 ( Make a task completed )**

現在タスクが割り当てられているプロセッサのうち、完了予定時刻が最小のプロセッサを選び、タスクを完了状態、プロセッサを空き状態をする。また、現在の時刻 C を完了予定時刻に更新する。

## 4. シミュレーションと考察

### 4.1 シミュレーション

前章で提案したスケジューリング方式及び従来のスケジューリング方式についてシミュレーションを行ない、提案方式の有効性を評価する。シミュレーションの対象のタスク集合として、文献 [2] で用いられている逆動力学方程式に関するタスク集合を用いる。

シミュレーションにおけるプロセッサのモデルとして以下のようなものを考える。

- 通信のコストは一定
- プロセッサの能力をランダム (1 ～ 10) に変える
- プロセッサ数を 1 から 20 まで変化させる

### 4.2 結果及び考察

新しい方法において、プロセッサの能力をランダムに変化させた場合、全処理時間はプロセッサ数の増加に伴い減少するが、プロセッサ数がある程度以上になると全処理時間はある一定値に近付く。従来の方法に対して約 1/2 の全処理時間を実現できることがわかった。

しかし、タスクの依存関係が少なく、プロセッサの能力差が大きな場合にはこの提案方式では分散環境における効率の良いスケジューリングを行なうことができない。

## 5. おわりに

分散環境でスーパーコンピューティングを行うために必要となるシステムについて PVM とその問題点を検討し、またこのような環境で重要となるタスクスケジューリングについて述べ、その新しいアルゴリズムとシミュレーション実験の結果について述べた。

## 参考文献

[1] G.A.Geist, A.Beguelin, J.J.Dongarra, W.Jiang, R.Manchek and V.S.Sunderan : *"PVM3 USER'S GUIDE AND REFERENCE MANUAL"*, Technical Report ORNL/TM-12187 , Oak Ridge National Laboratory, May 1993.

[2] 小林真也, 木村春彦, 武部幹 : "マルチプロセッサシステムにおける通信時間を考慮したタスク割当て法", 電子情報通信学会論文誌 (D-1), **J76-D-1**, 12 , pp.689-694 (1993-12).