

## アルゴリズムアニメーションによる並列マルチタスク管理の解析と評価

6L-1

須崎 有康 田沼 均 平野 聡 一杉 裕志 塚本 享治

電子技術総合研究所

suzaki@etl.go.jp, <http://www.etl.go.jp/Organization/Bunsan/Fluid/TSS.html>

## 1 はじめに

筆者らは文献[1]においてパーティショニングアルゴリズムを使った時分割処理方式を提案した。時分割処理方式の性能はシミュレータにより、投入された全タスクの処理時間、タスクの応答性、プロセッサ利用率などで従来のパーティショニングアルゴリズムより効率が良いことを示した。しかし、ここでの解析は主に全体性能の解析であり、投入される個々のタスクの動的な振舞いの解析は不十分であった。また、個々のタスクの解析が不十分のため、タスクの優先度を導入するための指標が立たなかった。このため筆者らは作成したシミュレータにアルゴリズムアニメーションの手法を適用し、時分割処理方式の動的な振舞いを解析した。ここではアルゴリズムの動作と共にグラフ表示を連動させることにより、動的に変化する振舞いの解析を容易にした。

## 2 並列 OS のアルゴリズムアニメーション

## 2.1 動作理解

R.Sedgewick によって推進されたアルゴリズムアニメーションは、アルゴリズムの動作を直観的に理解するのに有効な手段であった。特に並列計算機では計算機構成が二次元以上になり、アルゴリズムの理解にはビジュアルな手法がより重要な位置を占めるようになった。更に並列 OS ではタスク配置管理やメモリ割り当て管理など、計算機構成ばかりでなく投入されるタスクの生存期間も問題になる。これらの管理アルゴリズムの理解には対象となる問題空間が時間軸を含めて三次元以上になるため、その動作の理解、解析にはアルゴリズムアニメーションが必須である。

また OS の解析には、計算機の効率的な利用方法ばかりでなく、ユーザが投入した個々のタスクがどのように処理されるのかを知ることが重要である。計算機が効率的に利用されるようになって、ある特定のタスクが不公平に処理されることは望ましくない。個々のタスクの振舞い、タスク同士の相互干渉、タスクの特徴と OS の

管理アルゴリズムの関係などを知るためにはアルゴリズムアニメーションのように動的にその動作を視覚表示する手法が有効である。更に他の管理アルゴリズムの候補がある場合、同時実行を複数表示することにより、アルゴリズムの能力を比べることができる。特に性質の異なる入力に対し、性能が逆転するようなアルゴリズムでは、視覚的にその原因を掴むことは重要である。

## 2.2 性能評価

アルゴリズムアニメーションは、アルゴリズムの動的な振舞いを理解するのに有効な手段であるが、アルゴリズムの性能評価を数値的に決定づけるまでには至らない。そこで筆者らは、アルゴリズムアニメーションに幾つかのグラフを連係表示し、性能評価を行なえるようにした。これは性能評価をするばかりでなく、グラフから性能が低くなる場所探し、アルゴリズムを解析するのに有効である。

また、OS では二つの側面から評価する必要がある。計算機を有効利用できるか示すこと（計算機側の性能）と個々のユーザに提供されるサービスの程度（個々のユーザに提供できる性能）の両面から解析しなければならない。この両者は必ずしも両立するものではない。両者をグラフ化してアルゴリズムアニメーションと同期して見ることにより、それぞれの性能の変化や関係をアルゴリズムの動作と共に確認することができる。

## 3 並列マルチタスク管理アルゴリズムへの適用

## 3.1 並列マルチタスク管理アルゴリズム

筆者らはメッシュ結合並列計算機のパーティショニングアルゴリズム 2D Buddy(2DB) または Adaptive Scan(AS) を時分割処理することで性能向上ができることを示した[1]。ここでは対象とする並列計算機の構造を表す仮想並列計算機(スライス)を複数提供する。各スライスへのタスクの割り当てはパーティショニングアルゴリズムで行なう。既存のスライスへ割り当てられなくなったら新しいスライスを作成し、そのスライスにタスクを割り当てる。スライスはラウンドロビンで一定時間実計算機に割り当てられ、スライスに割り当てられたタスクの処理を進める。また本方式では、あるスライスのタスクが占有している領域がもし別のスライスで未使用の場合、空き

The analysis and evaluation of multi-task system on parallel computers using algorithm animation

Kuniyasu SUZAKI Hitoshi TANUMA Satoshi HIRANO

Yuuji ICHISUGI Michiharu TUKAMOTO

Electrotechnical Laboratory

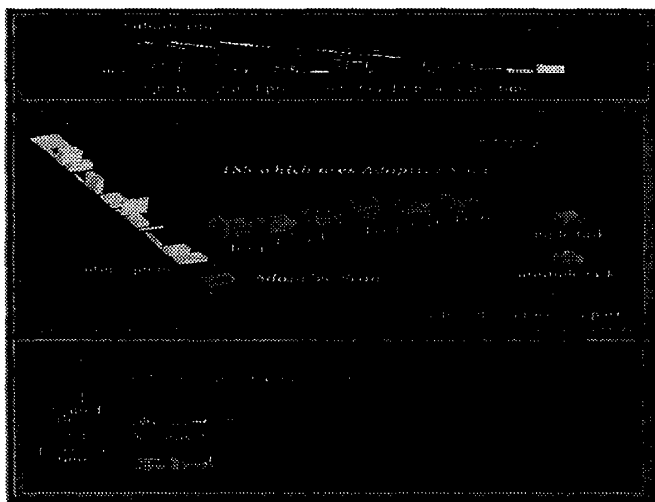


図 1: マルチタスク管理のアルゴリズムアニメーション

領域のあるスライスにそのタスクを割り当て、プロセッサの利用率を上げることができる。

### 3.2 アルゴリズムアニメーションの概要

2章の方針に従って並列マルチタスク管理を対象としたアルゴリズムアニメーションを作成した。この詳細と効果について述べる。

図 1 に今回作成したマルチタスク管理のアルゴリズムアニメーションを示した。この図の上段は投入されるタスク列を示している。タスクは底面積で必要とするプロセッサ数を表し、高さが処理にかかる時間を表している。中段は、時分割処理と通常のパーティショニングアルゴリズムによるタスク割り当て状況や個々のタスクの処理の進行を示している。下段では、終了したタスクを終了順に並べ、待ち時間と処理に必要とした時間の比を表している。このタスク列と上段のタスク列を比較することにより時分割処理と通常のパーティショニングアルゴリズムによる個々のタスク処理進行の違いを比較することができる。

図 2 には図 1 の処理と同期して表示されるグラフを示した。上段は時分割処理とパーティショニングアルゴリズムについて、それぞれのタスクの投入時間と終了時間の関係を表している。図 2 の中段と下段は時分割処理とパーティショニングアルゴリズムそれぞれのプロセッサ利用率の時間変化を表している。

### 3.3 アルゴリズムアニメーションの効果

マルチタスク管理のアルゴリズムアニメーションとそれに連動するグラフを表示することで、アルゴリズムの振舞いとその時の性能を同時に確かめられるようになった。図 2 の中段と下段のプロセッサ利用率とアルゴリズムアニメーションを同時にすることで、時分割処理化した場合の方が通常のアルゴリズムアニメーションより常時高いプロセッサ利用率を達成できることがわかった。また、通常のパーティショニングアルゴリズムでは、多数

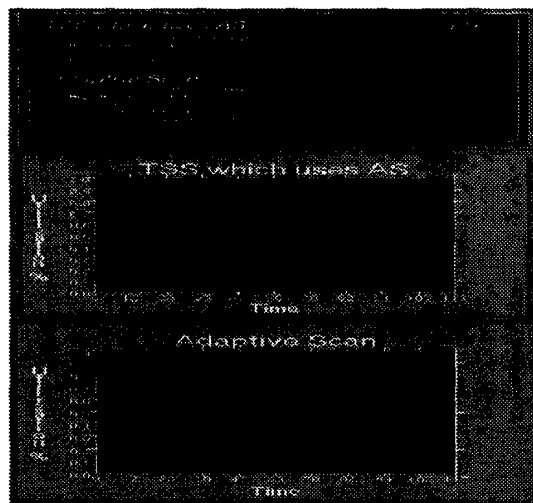


図 2: アルゴリズムアニメーションと連動したグラフ

のプロセッサを要求するタスクの前後で、タスクの終了待ちと割り当て待ちなどによりプロセッサ利用率が下がる閉塞状態を確認した。この状態は時分割処理で新たなスライスを提供することで回避できることがわかった。

更に図 1 の下段の終了した順のタスクと図 2 の上段のタスクの投入時間と終了時間の関係のグラフより、それぞれ 13 ぞれのタスク管理アルゴリズムが個々のタスクに及ぼす影響の違いが異なること明確にわかる。この結果は、現在は導入されていない優先順位の導入に示唆を与える。今後、タスクの必要とするプロセッサ数と処理時間および投入時間が、応答性にどのように影響するか相関を調べ、タスクの優先順位導入に役立てる予定である。また、プロセッサ利用率の変化も図 2 の中段より調べ、より高いプロセッサ利用率を達成できるようにしたい。

## 4 まとめ

本論文では、並列 OS のアルゴリズムアニメーションによる動的な振舞いの解析と評価方法を提案した。ここでは通常のアルゴリズムアニメーションのみでなく、個々の状態をグラフ表示と連動させることにより、性能評価および解析が容易になることを示した。この手法に基づいて並列マルチタスク管理のアルゴリズムを視覚化し、その効用について述べた。

なお現在、作成されたアルゴリズムアニメーションは <http://www.etl.go.jp/Organization/Bunsan/Fluid/TSS.html> で動画 (MPEG) として公開中である。

謝辞 本研究の一部は RWC 計画の一環として「超並列システムアーキテクチャに関する研究」で行なわれたものである。関係各位に感謝する。

## 参考文献

- [1] K. Suzaki, H. Tanuma, S. Hirano, and Y. Ichisugi. A time sharing system scheme that uses a partitioning algorithm for mesh-connected parallel computers. *SPDP-95*, 1995.