

並列プログラムの実行の可視化

7 J-6

下山朋彦, 早川栄一, 並木美太郎, 高橋延匡
(東京農工大学大学院 工学研究科 電子情報工学専攻)

1. はじめに

並列プログラムでは、どのタスクが同時に動作しているか、いつ同期命令が実行されたかなどの実行状況をつかみにくい。だが並列プログラムを作成する上で、プログラマが実行状況を把握することはデバッグ、評価、教育の観点から重要である。そこでタスク数が数十程度の並列プログラムを対象を絞り、実行を可視化するツールを作成した。

2. 可視化ツールの機能

本可視化ツールは、次に示すような可視化を行う。

(1) ダイアグラムによる時系列情報の図示

並列プログラムの時間軸に沿った動きを、横軸に時間軸、縦にタスクを並べたダイアグラムとして表示する。時間軸に沿った動きを示すことで、タスクの同期のタイミング、タスクの粒度を直感的に示すことができる。

(2) 相関図によるタスク等の相互関係の図示

タスク、セマフォ、データの参照関係を、それらの相関図として表示する。どのタスクがどのタスクを生成したか、タスク同士がどのセマフォを介して同期をとったか、どのタスクとどのタスクがデータを共有しているかなどの相関関係を示す。

(3) 棒グラフ等による実行の統計情報の図示

小規模な並列プログラムでも、同期命令が数百回も行われることがある。そのため、一つ一つの同期命令を追うことは現実的でない。そこで統計的なデータを取り、その結果を棒グラフ、ヒストグラム、折線グラフなどにする。

3. 可視化ツールの設計

3.1 構成

可視化ツールの構成上の特徴を次に示す。

(1) 並列計算機シミュレータの導入

現在、シングル CPU・マルチタスク環境で研究を行っている。だが並列計算機上の環境では、タスクの同期タイミング、実行時間が大きく異なる。そこでシミュレータを使用して、並列計算機上の実行状況を再現する。

(2) 実行記録の出力

並列シミュレータは、同期命令の実行時間など、並列プログラムの実行記録を出力する。

実行記録により可視化ツールを、実行記録をとる並列シミュレータと、実行記録を図示する図示ツールに分割することができ、モジュール性を向上することができる。

また実行記録と同じ順番で同期命令を実行することで、並列処理の非決定性を無くした再現性のある実行を行うことができる。

可視化ツールの全体構成を図1に示す。並列プログラムをシミュレータにかけると、プログラムの実行記録が得られ、その実行記録を図示ツールにかけると実行をビジュア

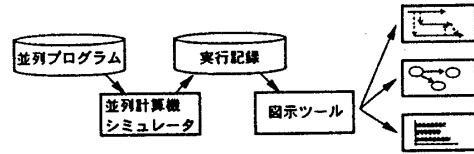


図1. 可視化ツールの構成

ライズすることができる。

3.2 シミュレータの設計

(1) シミュレータの目的

シミュレータを採用した目的は、次の二点である。

- (a) 並列計算機のアーキテクチャの実験を行える
- (b) メモリ競合など、細かい実行データがとれる

シミュレータを使用することで、様々なタイプの並列計算機をシミュレートすることができる。また後述するようにOSの機能をシミュレータ内で実現しているため、様々なOSを使用することができる。これらにより、並列システムに関しての実験を行うことができる。

またメモリ競合の様子など、実際の並列計算機ではとれない細かなデータをとることができる。

(2) シミュレータの概念構成

並列システムシミュレータの構成を、図2に示す。システムは、計算機シミュレータと、OSシミュレータに分かれている。これはOSシミュレータを交換し、いろいろなモデルのOSの実験を行うためである。

現在シミュレータの上で、モトローラのMC68000を使用した共有メモリ型計算機上で動くOMICRON V3をシミュレートし、実験を行っている。

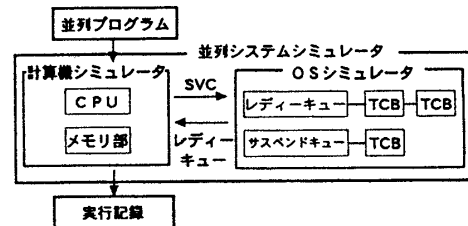


図2. 並列システムシミュレータの構成

(3) 計算機シミュレータ

計算機シミュレータは、図3のように並列計算機を仮想化する。計算機はいくつかのプロセッサエレメント (PE) とメモリユニット (MU) からなり、各 PE、MU がどう接続されているかはブラックボックスとなっている。このブラックボックス内で、バス競合などもシミュレートする。またこれを変更することで、様々な並列計算機のシミュレートを行うことができる。またバス競合の状態を記録したり、共有メモリのキャッシングの実験を行うことができる。

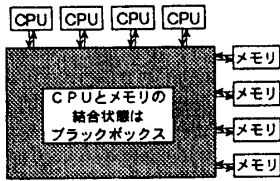


図3. 並列計算機の仮想化

(4) OSシミュレータ

OSシミュレータは、主にタスク管理を行う。内部では、普通のOSのタスクスケジューラと同様に、レディーキュー、サスペンドキューといったタスクキューによりタスク管理を行っている。

タスクへの操作は、計算機シミュレータから送られるSVC リクエストにより行う。またレディーキューを計算機シミュレータに渡し、レディーキュー内のタスクの並列実行を依頼する。

OSシミュレータは、同期命令の実行記録を出力する。この記録は図示ツールで使用するだけでなく、実行記録に沿って実行を行うことで並列処理の非決定性を無くした実行にも使用する。

3.3 図示ツールの設計

I. ダイアグラム

ダイアグラムの表示例を図4に示す。ダイアグラムにより、時系列データとしてタスクの実行時間を表示する。タスクがランニング状態になっている時間は、タスクを示す横線が実線になっている長さで、サスペンド状態になっている時間は点線の長さで示される。線の長さから、タスクの粒度や待ち時間によるロス、直感的につかむことが可能である。

また SVC 実行時間は、横線がハイライトで示される。タスクの粒度と SVC 実行時間の比を、明確に示すことができる。

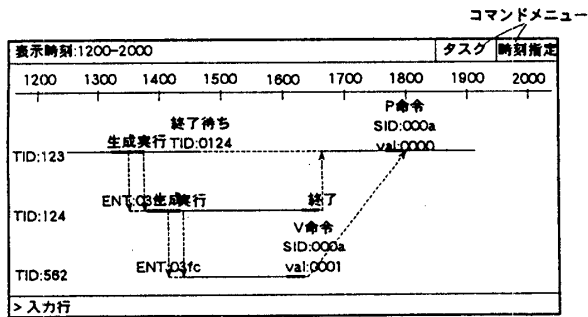


図4. ダイアグラムの表示例

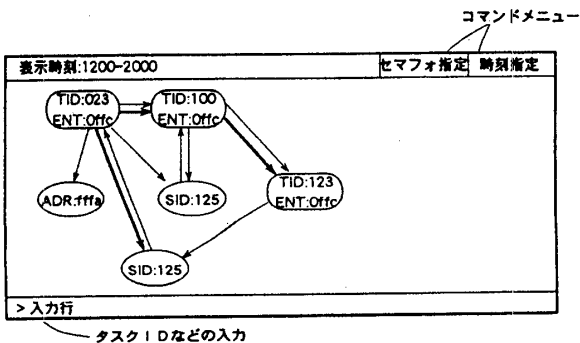


図5. 相関図の表示例

II. 相関図

相関図の表示例を、図5に示す。動的にタスクを生成するプログラムでは、並列に処理できる部分ができるタスクを生成してゆく。そこでタスクの生成関係を示すことで、プログラマに並列実行状況を示すことができる。

また、共有資源を管理するセマフォや、共有資源となっているデータの参照関係を調べることで、その資源にアクセスしているタスクを明らかにすることができる。

III. グラフ

統計値のグラフの例として、タスクの平均粒度の棒グラフの例を図6に示す。タスクの平均粒度や、各タスクの粒度の分布を知ることができる。タスクの粒度はプログラムの性能に大きな影響を与えるため、タスクが適切な粒度で動作しているか確認することは有効である。

また平均粒度の棒グラフを、粒度の大きさによりソートすることができるため、粒度が細かくて並列化のオーバーヘッドの方が大きいタスクをピックアップすることにも使用できる。

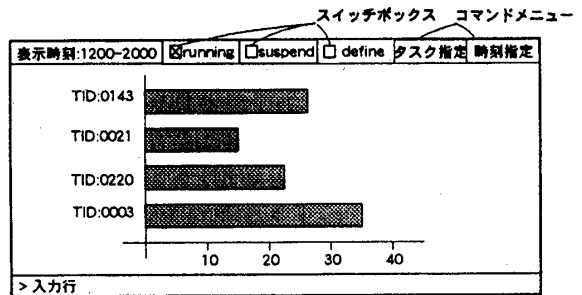


図6. 平均粒度の棒グラフの表示例

5. おわりに

(1) 成果

本可視化ツールにより、並列プログラムの実行状況を直感的に示すことができた。

ダイアグラムにより、タスク間の細かい同期の様子を、その時系列情報も含めて示すことができた。相関図によりタスクの生成状況や同期状況を示すことができた。統計情報を示す棒グラフやヒストグラムにより、全実行時間にわたった実行状況の様子を示すことができた。

(2) 問題点

本可視化ツールの実現の結果、つぎの点が問題になった。タスクの相関図を表示する際に、すべてのタスクの相関図を表示すると、すぐに画面が溢れてしまう。この問題は本ツールの設計時から認識し、指定したタスクと直接同期をとっているタスクのみを表示するなど工夫した。しかしまだまだ、改良の必要があることがわかった。

またソースリストとの対応をとるため、ソースコードデバッガと同様な機能が必要であることは設計時から認識していたが、初版では省略した。しかしタスクに渡された引数の表示などの機能が、可視化ツールにとり予想以上に重要であるとわかった。

6. 参考文献

[1] 下山朋彦, 他: アーリー法による並列構文解析プログラムの実現, 情報処理学会第44回全国大会, 1992
 [2] 鈴木茂夫, 他: OS/omicron 第2版の実現と評価, オペレーティングシステム研究会, 42-1, 1989.2.