

生産システムシミュレータ群の統合に関する研究
—同期メカニズムとその特性—

01501824	神戸大学	藤井 進
02301844	神戸大学院	*角田 晴久
02102464	神戸大学院	木谷 靖
	神戸大学院	萩田 篤史

1 はじめに

CIM環境下では新設備の導入に際して、その前後の工程との関連を十分に考慮して、その妥当性を検討することが必要である。また、ある工程における意志決定に際しても他の工程との関連を考慮することが重要である。このためには、従来のような工程や設備に関するシミュレーションだけでなく、より全体的なシミュレーションシステムを構築することが望まれる。ここでは、すでに開発されているシミュレータ群を統合することにより分散型シミュレーションシステムを実現することを考える。

これまでの報告[1]ではシミュレータ群を統合するために必要な機能について述べた。本報告では統合する上で最も重要な機能の一つである同期メカニズムの特性について実験的生産システムモデルへの適用と関連付けて述べる。

2 同期メカニズム

同期メカニズムとしては代表的なものとして以下の3つが挙げられる。

2.1 コンサバティブ法

コンサバティブ法ではあるプロセスへのメッセージの送信はあらかじめ決められたプロセスからしか行われぬ。またメッセージの通信路はプロセス内及び同一通信路においてはメッセージの追越しは起こらないことを仮定する。各プロセスは全ての入力通信路上にメッセージが到着した時点でそれらのメッセージの中で最小のタイムスタンプ値を持つメッセージの処理を行う。しかしこの方法ではプロセス間の通信路が閉路を構成する場合にはデッドロックという状態が生じる。デッドロックを防ぐ方法としてChandy・Misraのアルゴリズムが広く知られている。このアルゴリズムではヌルメッセージという実システムに存在しないメッセージを用いることによりデッドロックを防ぐ。このアルゴリズムではあるプロセスからメッセージが到着し、処理を行った後、そのプロセスから送信可能性のある全てのプロセスにヌルメッセージを送る。このメッセージはこのプロセスからこれ以下の最小のタイムスタンプを持ったメッセージが到着しないことを保証している。 — 52 —

2.2 タイムワーブ法

タイムワーブ法は各プロセスに設定したVirtual Timeの概念に基づいた同期法であり、JeffersonのAggressive cancellationに代表される方法である。各プロセスはシミュレーション時刻の減少も考慮したローカルクロックを持ち、これが正しい時刻を指していることを仮定してシミュレーションの処理を進める。

通常のメッセージ以外に、アンチメッセージと呼ばれるすでに送信したメッセージを取り消す役割を持つメッセージも準備する。受信されるべき時刻はreceive timeとして付与されているが、メッセージの発信順序と受信順序の関係については何の仮定もしておらず、一般にはメッセージの到着順序を決定することはできない。

ローカルクロックより小さなタイムスタンプを持つメッセージが到着したときには、メッセージの到着時間までクロックを巻戻し（ロールバックという）、その時点での状態変数を復帰させる。

誤って送信したことが判明したメッセージに関してはメッセージを取り消すために、アンチメッセージを送信する。アンチメッセージを受信した側は該当するメッセージを未実行のメッセージ列から削除する。そのメッセージがすでに実行されてしまっている場合には、さらにロールバックが行われる。

2.3 タイムバケット法

生産システムにおいては、ショップ内におけるワークの搬送に比べて、ショップ間の搬送は間欠的であり、また集中的であると考えられる。また、午前中に処理の完了したワークは、別のショップに搬送されたとしても午後になるまで必要となることは少ないといった場合も多い。生産活動の全てを各ショップの活動が比較的独立に実行できるタイムバケットと呼ばれる小区間に分割して、シミュレーションモデルを構築しようとするのがタイムバケット法である。タイムバケット法では、シミュレーション時間を固定長のタイムバケットに分割する。タイムバケット内の処理が完全に終了した後にシミュレーションを再開し次のタイムバケットの処理を開始する。

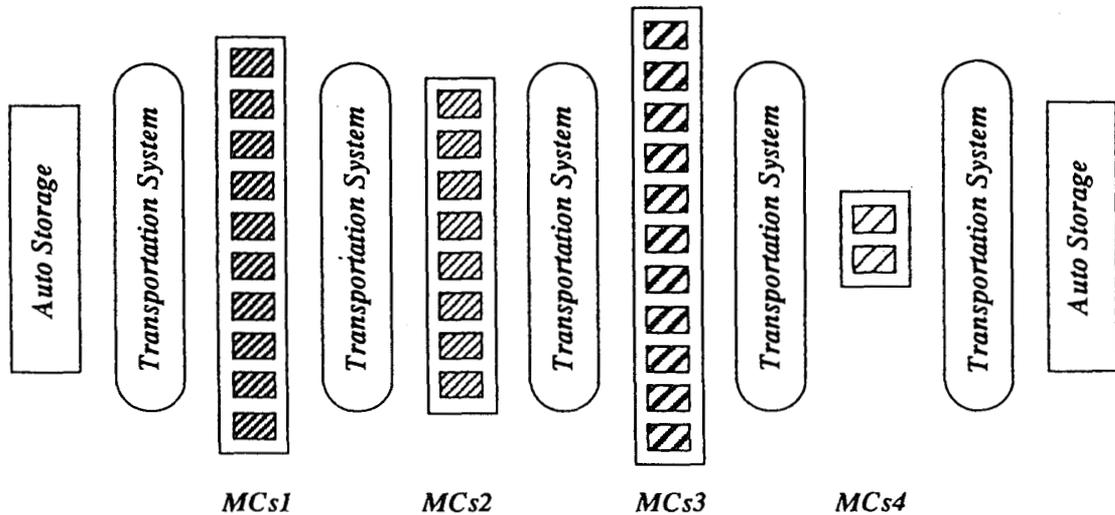


図1 シミュレーション用生産システムモデル

3 シミュレーションモデル

大量生産向きFMSの一つの試案として図1に示すようなシステム構成を考える[2]。これは図1に示すように同機能を持つマシニングセンタを並列に配置したセルを直列に接続する。このシステムにおいてはワークは頻繁に搬入出され、即時的に使用されるものとする。またセル間の搬送はランダムアクセス型であり、それら搬送システムは一元的に管理されている。

4 適用例

実験システム構築には図2に示すように汎用シミュレーション言語であるSLAM IIを用いた。各セルをそれぞれ1台のプロセッサに構築した。そして、セル間の搬送を全て1台のプロセッサ上に構築した。同期メカニズムにはコンサーバティブ法とタイムバケット法を用いた。またタイムワープ法に関してはC言語で記述されたシミュレーションライブラリであるsmplを用いて実験システムを構築した。コンサーバティブ法を用いた場合、ヌルメッセージ数の増加が実行速度向上を妨げるものと考えられる。そのためヌルメッセージ数と実メッセージ数を及び実行時間を測定した。この結果を図3に示す。タイムバケット法では、タイムバケットの大きさによって実行速度とシミュレーション結果に違いが生じる。またタイムワープ法では、ロールバックの増加が実行速度を低下させると考えられる。

5 おわりに

各同期メカニズムの特性結果の詳細に関しては、発表当日に示す。

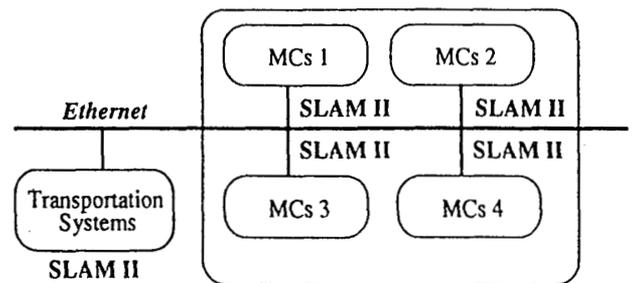


図2 実験システム

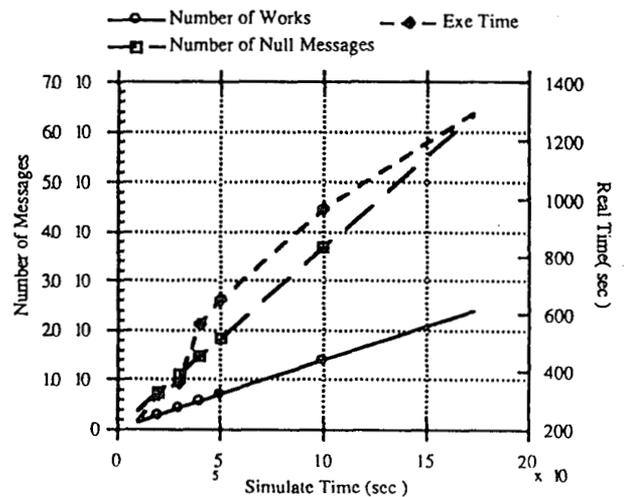


図3 コンサーバティブ法によるシミュレーション実行時間とメッセージ数

参考文献

- [1] 藤井ら, 分散型生産システムシミュレーションに関する研究, OR学会秋季アブストラクト集, 1994, 130-131
- [2] IMS/VI-2「機能変態可能な複合加工システム技術に関する研究開発」に関する研究成果報告書, 1993, IROFA, IMSセンター