

マルチプロセス環境を利用したベクトル時間による 作業プロセスの順序関係の分析

石田 諭史[†], 宮崎 貴成[†], 坂下 善彦[†]

[†]湘南工科大学 工学部

1 はじめに

生産や建設工事などのプロジェクトにおいて、スケジュールの決定は作業プロセスの順序関係に影響され、作業計画を策定する場合には作業間の順序関係に則った計画をたてなければならない。しかし順序関係の発見や分析は人間の経験によって判断されている。現在は、大規模プロジェクトが増えており、作業の順序関係が複雑に入り組んだ場合が想定される。従って、机上で全ての順序関係の発見は困難であると考えられる。

本研究ではある作業プロセスをマルチプロセス環境上で実行させベクトル時間を生成する。一方で同じ作業プロセスを机上でシミュレーションを行ってベクトル時間を生成する。この2つのベクトル時間を、順序関係を解析するプログラムにより解析し、両方の結果を比較することによって開発した解析プログラムの有効性を示す。

2 ベクトル時間の解析方法

ベクトル時間とは分散処理システムにおいてイベントの順序付けが必要な部分のみに付くイベント発生時刻の記述方法である[1][2]。ベクトル時間を用いることにより各プロセスの局所時計を n 次元ベクトル $V_i = \langle V_i[1], \dots, V_i[n] \rangle$ で表すことができる。各プロセスの局所時刻にこの記述方法を使用し、順序関係を分析する。

ベクトル時間を作業プロセスの順序の解析に反映するために、分散システムにおける通信とプロセスの概念を用いる。通信によるメッセージの送受信を成果物の受け渡しとし、タイムスタンプを作業プロセスの開始時刻、または成果物の受け渡しなどのイベントのタイミングとする。これらを反映したベクトル時間の記述について、時空ダイアグラムを用いて図1に示す。

ベクトル時間で記述された順序の解析は成果物の送信時刻の比較を行う[2]。作業プロセス

P1 の送信イベント e_1 と、プロセス P2 の送信イベント e_2 のそれぞれのベクトル時間を V_i, V_j とする。 $V_i \leq V_j$ のとき、

- ① $V_i[x] \leq V_j[x]$ ならばイベント e_2 は e_1 に先行し、 e_1 と e_2 には順序関係がある。
- ② $V_i[x] \leq V_j[x]$ でも $V_i[y] \leq V_j[y]$ でもなければ、イベント e_1 と e_2 は順序関係がない。従って、並行に動作する。

以上の解析方法に基づき順序関係を発見する。

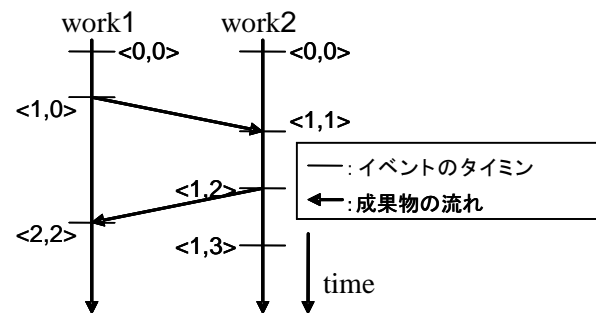


図1: 作業プロセスを反映したベクトル時間を用いた時空ダイアグラム

3 ベクトル時間の生成

ベクトル時間の生成には、仮想計算機環境 (VM) を使用して、各 VM に 1 つのプロセスを構築した。メッセージの受け渡しおよび履歴は API ライブラリを用いた。図2にベクトル時間の生成環境の全体図を示す。

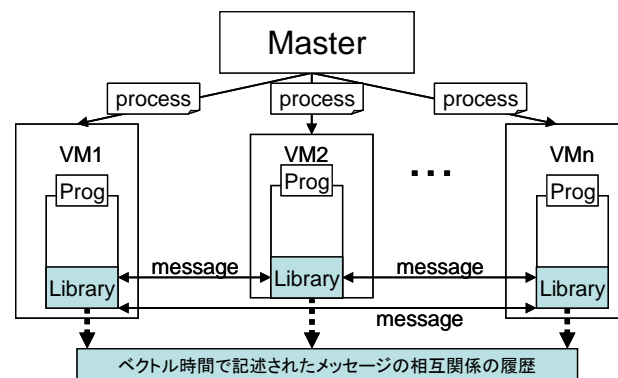


図2: ベクトル時間の生成環境の全体図

作業プロセスのプログラム製作には、MPI(Message Passing Interface)を用いた API ラ

Analysis of order relations for work process based on the vector time in multi process environments

[†] Satoshi Ishida, Takanori Miyazaki, Yoshihiko Sakashita
Faculty of Engineering, Shonan Institute of Technology

イブライを実装して製作した。この API ライブラリを経由してアプリケーション間におけるメッセージの送受信を実現し、ベクトル時間を生成する。API ライブラリの仕様は表 1、提供する関数を表 2 にそれぞれ示す。

表 1: API ライブラリの仕様

項目	仕様
モジュール名	sab.c
ライブラリ形式	スタティック
関連モジュール	mpi.lib

表 2: API ライブラリが提供する関数

関数名	機能
MPI_ready	MPI 関数の使用を可能にする
MPI_END	MPI 関数の使用終了
Send	メッセージ送信と VT 出力
Rcev	メッセージ受信と VT 更新
my_VT	内部処理による VT 更新

4 検証

検証方法として、サンプルの PERT 図から机上で生成したベクトル時間と本研究で生成したベクトル時間を、それぞれベクトル時間の解析プログラムで解析し、その結果を比較し同様の結果が得られているかを検証する。検証の流れを図 3 に示す。

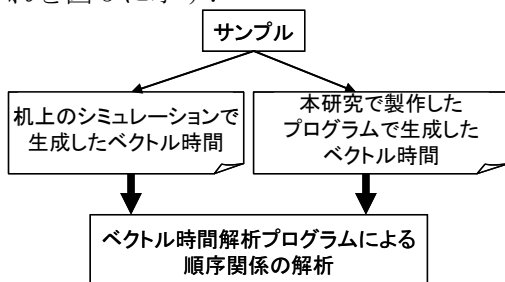


図 3: 検証の流れ

検証に用いたサンプルは資料に掲載された例題 [3] を参考にし、本研究でベクトル時間を生成するために業務分析を反映したサンプルを用いた。業務分析により作業プロセスがどのグループによって処理または実行されるのかを判定する。その分析結果を反映した PERT 図からベクトル時間を生成する。サンプルを図 4 に示す。

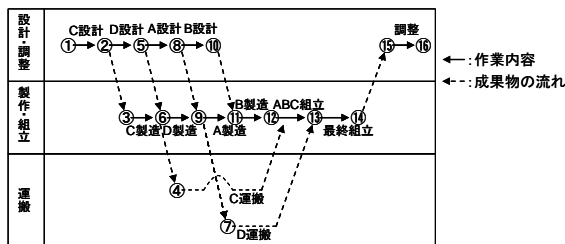


図 4: サンプルの PERT 図

検証の結果、製作したベクトル時間の生成プログラムによる解析結果と机上でのシミュレーションによるベクトル時間の解析結果は一致した。従って本手法が順序関係の解析に有効であることを示した。本研究によって生成されたベクトル時間を図 5 に、解析プログラムによる結果を表 3 に示す。

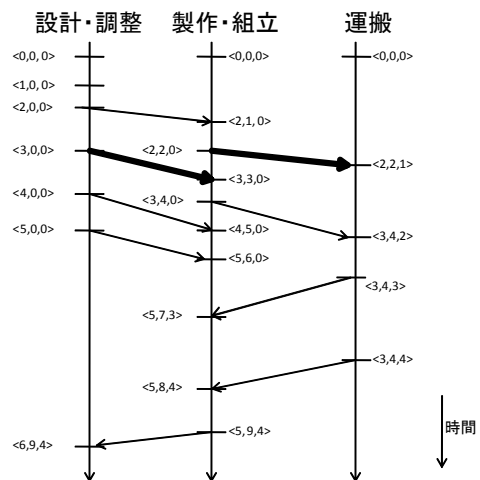


図 5: 本手法によって生成されたベクトル時間

表 3: 解析プログラムによる結果

ベクトル時間	比較結果
V1=<2, 0, 0>	
V2=<3, 0, 0>	V2 と V3 が並行に動作可能
V3=<4, 0, 0>	V4 と V5 が並行に動作可能
V4=<2, 2, 0>	...
V5=<3, 4, 3>	
...	

5 考察

本研究の手段と環境を用いることは順序関係の解析に有効であることがわかった。今後、机上で解析されたことによる順序関係の誤ったサンプルを用いて、見落とされた順序関係を指摘することができれば、複雑な作業プロセスでも順序が見つけられる可能性を得られる。

参考文献

[1] Friedemann Mattern. Virtual time and global states of distributed systems. In Michel Cosnard et. al. editor, Proceedings of the International Workshop on Parallel and Distributed Algorithms, pages 215-226. North-Holland, October 1989

[2] 谷口 秀夫 (編), 谷口 秀夫, 佐藤 一朗, 佐藤 文明, 柴田 義孝, 新城 靖, 横山 和俊 (著), 情報処理学会編集 IT Text 分散処理, オーム社, 2005.

[3] 刀根 薫 監修, PRET 講座 I 基礎編, 東洋経済新報社, 1966.