1R－7

分散共有メモリシステムの複製管理方式*

川口 昇、中村 健二、佐藤 文明、水野 忠則†
静岡大学理工学研究科‡

1　はじめに

分散システムでは共有メモリを用いて複数のプロセスを並列に動作させることが可能である。しかし、共有メモリを管理するサーバに障害が生じるとそれを利用しているプロセス全てが停止してしまう問題がある。これを解決する方法の１つとして、ネットワーク上に共有メモリの複製を配置する方法が考えられる。本研究では共有メモリとしてLindaモデルのタプルスペース通信を対象とし、タプルスペース通信の特性を生かしたより効率的な複製管理方式を提案する。

2　分散共有メモリシステム

本研究では分散共有メモリとしてLindaモデルのタプルスペース通信を扱う。Lindaは、タプルスペース（以下TSと略記）と呼ばれる共有メモリ空間と、TSに対してデータ（タプルと呼ぶ）のやりとりを行うための４種の基本命令を提供する。基本命令はout()、eval()（タプルをTSに置く）、in()（タプルをTSから取り出す）、rd()（タプルをTSから読み込む）の４つで、プログラマは、この基本命令を使って並列実行の単位を生成し、プログラムを並列に実行できる。

このようなTS通信は、書き込み頻度が読み込み頻度より大きい、タプルはファイルのように更新されることが無いという２つの特性を持つ。

3　提案する複製管理方式

3.1　定数合意方式（Voting）

本研究ではTSの複製を配置したモデルを対象とし、複製管理に定数合意方式を採用する。定数合意方式（図1）は書き込み定数、読み込み定数を設け、それぞれの定数の数だけ読み書きを行う。この時、書き込み定数と読み込み定数の和が複製数より大きくなるように設定すると、読み書きが共に行なわれる複製が少なくとも１つは存在する。従って、読み込み時に最新のデータが得られ、一貫性が保証される。データが最新かどうかは複製ごとに付加されたバージョン番号で判断する。定数合意方式の利点は読み込み頻度、書き込み頻度に合わせて各定数を設定できる点である。

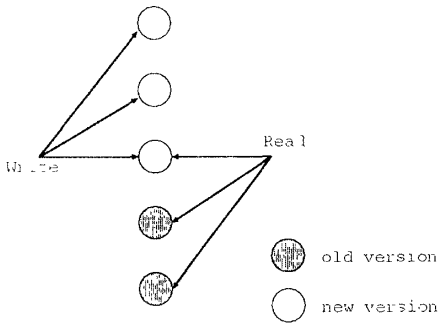

図 1: 定数合意方式

3.2　改良点

3.2.1　バージョン番号

2より、タプルは更新されることがないため、定数合意方式で用いられていたバージョン番号は不要になる。また、TS通信ではin()を実現するために削除操作が必要になる。定数合意方式の条件を満たすには、削除操作は書き込み操作の時と同一のタプルの組に対してなされる必要がある。そこで 書き込み操作の時に、「他の複製の位置を示す識別子」をタプルに付加することでこれを解決する。

3.2.2　非同期処理

分散システムで共有領域にアクセスする時は他からのアクセスがないように共有領域にロックを掛ける必要がある。しかし今回のように共有領域が複数

*A Method of Replication Control on Distributed Shared Memory System

†Noboru Kawaguchi, Kenji Nakamura, Fumiaki Sato, Tadanori Mizuno
‡Shizuoka University

ある場合でも、通常は全複製にロックを掛ける必要があり、ロックが解除されるまでの待ち時間がより長くなる。つまり、複数の複製間で同期をとる必要があり、処理効率の面からは望ましくない。そこで本研究ではこれを非同期に行う方法を提案する。

非同期方式は処理時間の面では優れているが、ロックを掛けないためメッセージの到着順序が複製ごとに異なる場合があり、一貫性が保証されなくなる。そこで、図 2 のように待ち行列中のメッセージを Birman のアルゴリズム [2] を用いて並び換え、到着するメッセージ順序が全ての複製で同一になるようにしてこれを解決している。

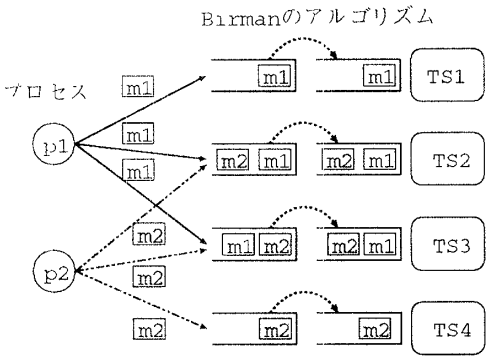

図 2: 非同期方式

4 シミュレーションによる評価

従来の同期方式と本方式の非同期方式の応答時間を比較するため、本研究ではシミュレータを作成し、評価を行った。図 3はプロセス数を 3、TS の数を 4、要求の処理時間をパラメータ 0.5 の指数分布とした時の応答時間をグラフにしたものである。グラフからも分かるように同じ要求発生率でも非同期方式のほうがより短い応答時間で処理を行なえることがわかる。

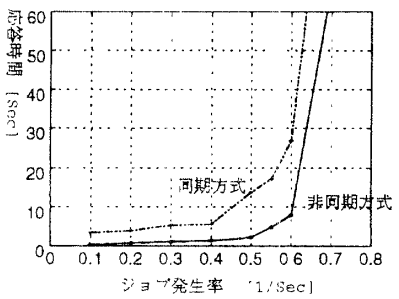

図 3: シミュレーション結果

5 複製付き TS 通信の実装

本研究の複製管理方式は現在 UNIX システム上に実装中である。ここでは実装の概要と現在の進行状況について述べる。

システム構成は TS を管理するための TS サーバ、複製管理プロトコルを実現するための通信ライブラリ (クライアント)、複製管理をグループ通信で実現するためのグループサーバからなる。このときのグループサーバと TS サーバはデーモンプロセスである。また、複製管理アルゴリズムは通信ライブラリに含めるため、ユーザプログラマは TS の複製の存在や一貫性に関して意識しなくてもよい。ただし、Birman のアルゴリズムは 2PC に基づいているため、複製管理アルゴリズムは TS サーバと通信ライブラリに分割して実行される。

現段階ではグループサーバのメンバ管理の部分と、TS サーバ、通信ライブラリ間のメッセージの並び換え部分が完成している。今後は残りの基本部分を完成し、性能評価ができる段階にもっていく予定である。

6 まとめ

本研究では複製付き TS 通信の複製管理方法を提案しシミュレーションで評価をした。ポイントは複製管理に定数合意方式を用いて TS 通信の特性に合わせて改良を加えたこと、非同期方式に Birman のアルゴリズムを組み合わせたことである。今後は本方式の実装を進めるとともに、複製の配置方法についても考察し改良を加えていきたい。

参考文献

[1] 木村 康則, 住元 真司, 細井 聡, 小沢 年弘, 服部 彰 : Linda 処理系の試作について, 信学研究報告, CPSY90-22, (1992)

[2] 滝沢 誠, 中村 章人 : 放送通信アルゴリズム, 情報処理, Vol.34, No.11, (1993.11)

[3] 吉田 紀彦, 樽崎 修二 : 場と一体化したプロセスの概念に基づく並列協調処理モデル Cellula, 情報処理学会論文誌, vol.31, No.7 (1990.7)

[4] K. Birman, A. Schiper, P. Stephenson : Lightweight Causal and Atomic Group Multicast, ACM Transactions on Computer Systems, vol.9, No.3 (1991.8)