

Linda のタプル管理プロトコルの提案*

3D-4

松田 哲史, 武並 佳則, 田口 哲也
システムエレクトロニクス研究開発センター
住友電気工業(株)†

1. はじめに

近年、多数の計算機が LAN でつながれているネットワーク環境の普及に伴い、ネットワーク環境上での並列・分散プログラム開発環境の研究が活発に進められている。Linda[1]はそうした研究の1つであり、タプル空間と呼ばれる一種の仮想的な共有メモリ空間内に存在する、タプルと呼ばれる共有データへのアクセス(Rd=読み出し, In=読み出し及び削除, Out=書き込み)としてプロセス間通信を実現する。Lindaで記述されたプログラムをネットワーク環境で実行するシステムで問題となるのは、タプルのアクセス時間をいかに短くするかということである。このために様々な手法[2]が提案されているが、本研究では[2]の手法を拡張し、1) タプルをアクセスするプロセスが存在する計算機, 2) タプルアクセスタイプ, の2つの情報をプログラマが annotation として記述することを許し、その情報に基づいてタプル管理プロトコルを使い分けることにより、タプルへのアクセス時間を短くしている[4]。本稿では、タプルアクセスタイプの分類と、タプル管理プロトコルについて述べる。

2. Linda におけるタプルアクセスタイプの分類

ここでタプルアクセスタイプと呼んでいるものは、1つのタプルが複数のプロセスによって共有されている場合に、この複数のプロセスがどのような形態でタプルをアクセスするかに関する情報である。本研究では、Linda プログラム中で用いられるタプルアクセスタイプを4通りに分類した。この分類は、[3]に述べられている共有データアクセスタイプの分類を参考にしており、以下に4つのアクセスタイプを説明する。

・ *general*: 1つのタプルをアクセスするプロセスがタプルに対して一意に決まるアクセスタイプ。

・ *producer-consumer*: マスター/ワーカー型のプログラムで、ワーカーがタスクキューの中から無差別にタスクを取ってくるような場合の、タスクを表すタプルに対応するアクセスタイプ。

・ *read-mostly*: 1つのタプルが複数のプロセスから読み出される場合に対応するアクセスタイプ。

・ *migratory*: 1つのタプルが、複数のプロセスにアクセスされるが、ある一定期間を区切ってみると1つのプロセスだけがアクセスしている場合に対応するアクセスタイプ。

3. 提案するタプル管理プロトコル

タプル管理プロトコルは、1対1, マルチキャスト(以下 R-multi と呼ぶ), ネットワーク全体で全ての受信者が受信するメッセージの順序の間に矛盾がないマルチキャスト(以下 RS-multi と呼ぶ)の3種類の通信プロトコルを用いる。全ての受信者が受信するメッセージの順序の間に矛盾がないとは、受信者 A と B が共にメッセージ 1 と 2 を受信したならば、A から見てメッセージ 1 と 2 を受信した順序が、B から見てメッセージ 1 と 2 を受信した順序と一致することをいう。通信プロトコルの処理にかかる時間は、1対1, R-multi, RS-multi の順で増加する。本稿では紙数の関係上アクセスタイプが read-mostly の場合のタプル管理プロトコルのみを簡単に説明する。Linda プログラムの実行に参加する計算機上では、タプルの管理を行なうタプルサーバープロセスが実行されている。アクセスタイプ read-mostly のタプルの場合、そのタプルをアクセスするプロセスを実行する全ての計算機上のタプルサーバーにタプルの複製が作られる。このタプルに対する In, Out 要求は、複製をもつ全てのタプルサーバーに対して RS-multi プロトコルで送信される。ここで、全てのタプルサーバー

* A proposal of Linda tuple management protocol

† Matsuda Tetsushi, Takenami Yoshinori, Taguchi Tetsuya
Systems Electronics R&D center, Sumitomo Electric.

がIn/Out要求を同じ順序で受信することが重要である。Rd要求は、このタプルの複製をもつタブルサーバーの内、任意の1つ(通常同一計算機上のもの)に対して1対1プロトコルで送信される。

4. serializability のための十分条件

Linda で書かれたプログラムを実行するシステムが満たすべき性質として、serializability(プログラムを実行する各プロセスが発行するタブルアクセス要求を、1つのサーバーが逐次的に処理したと仮定して問題が生じないこと)がある。serializability は Linda のタブル操作の定義から必ず満たされなければならない。本研究で用いるタブル管理プロトコル群を用いた場合、以下の条件を満たすように通信順序制御を行えば、serializability が満たされることを証明している。

- ・1つのプロセスが送信する要求は、1対1, R-multi, RS-multi を問わず、送信側が送信した順序に矛盾しない順序で受信側は受信する。
- ・あるプロセスがRd/Out要求を送信する場合、自プロセスが以前送信したR-multi またはRS-multi を用いるIn要求に対して、全てのタブルサーバーでの処理順序が確定するまでRd/Out要求を送信しない。証明には、Linda プログラムを実行するプロセスが行なうタブルアクセス(In,Rd,Out)を表すノードと、1つのプロセスが実行するタブルアクセス間の前後関係、タブルアクセス間のデータ依存関係(同じタブルをアクセスするOutはIn,Rdに先立つ、同じタブルをアクセスするRdはInに先立つ)を表す有向エッジから成る有向グラフを用いる。このグラフにループが存在しないことが、serializability を満たすための十分条件である。ループが存在しないことは、エッジの始点となるタブルアクセス要求のタブルサーバーでの処理順序が確定する時刻が、有向エッジに沿って増加することから導かれる。

5. 実測結果

今回導入したタブル管理プロトコルを用いることで、どの程度タブルアクセス時間を短縮できるかを、Sun SparcStation2 5台をEthernetでつなげた環境で実測した。表1に、複数の計算機が同時に1つのタブルをRdでアクセスする時に、read-mostly プロトコルを使用/不使用の場合を比較した結果を示す。この結果より、読み出すデータのサイズと計算機数が大きくなると、read-mostly タブル管理プロトコルが有

Rd する計算機数	2	3	4
データサイズ	Rd 1 回あたりの時間 (sec)		
read-mostly 使用			
1KB	0.0419	0.0438	0.0466
3KB	0.0539	0.0530	0.0557
5KB	0.0639	0.0627	0.0656
read-mostly 不使用			
1KB	0.0186	0.0256	0.0315
3KB	0.0362	0.0501	0.0713
5KB	0.0527	0.0771	0.1260

表 1: read-mostly の測定結果

効に働くことが判る。データサイズ、計算機数が小さい場合に、read-mostly 用プロトコルを使用した方がRd時間が長くなるのは、read-mostly 用プロトコルが利用するRS-multiプロトコルの方が、1対1プロトコルよりも通信コストが高いからである。

6. 結論

ネットワーク環境で、Linda で記述された並列・分散プログラムを実行する際に問題となる、タブルアクセス時間を減らすために、タブルのアクセスタイプを分類し、各アクセスタイプに適化したタブル管理プロトコルを提案し、実装して有効であることを確認した。今後の課題としては、

- ・今回提案したタブルアクセスタイプが、実際のプログラムで現れることの検証
- ・タブルをアクセスするプロセスが動的に変化する場合への対応が挙げられる。

参考文献

- [1] N. Carriero, D. Gelernter, *Linda in Context*, CACM, April 1989
- [2] G. Wilson, *Improving the Performance of Generative Communication Systems by Using Application-Specific Mapping Functions*, Edinburgh Parallel Computing Center TR91-13, p129-142, 1991
- [3] J. Bennet, J. Carter and W. Zwaenepoel, *Adaptive Software Cache Management for Distributed Shared Memory*, ISCA 1990, IEEE, p125-134
- [4] 松田 哲史, 武並 佳則, LAN 上での Linda タブル管理方式について, JSPP93 予稿集, p191-198