分散メモリシステム上での粗粒度並列処理における投機的なプリロード

那須 弘志† 本多 弘樹‡ 弓場 敏嗣‡
†電気通信大学電気通信学部 ‡電気通信大学 大学院情報システム学研究科

1 はじめに

PC クラスタなどの分散メモリシステム上で粗粒度並列処理を実現するために,「データ到達条件」を用いて異なるプロセッサに割り当てられたマクロタスク (Macro Task: MT) 間のデータ授受をプロセッサ間通信により明示的に行う方法が提案されている [2].

MT 実行とデータ通信とのオーバラップが可能なシステム上で粗粒度並列処理を行う場合，MT の実行開始以前に他の MT の処理と並行して必要なデータの通信を行う「プリロード」を用いることが望ましい．これによって，データ通信によるレイテンシを隠蔽し，MT の実行をより速いタイミングで行うことができるからである．

本稿では，プリロードを投機的に行う粗粒度並列処理を実現するための方法を示す．

2 粗粒度並列処理におけるプリロード

2.1 分散メモリシステム上での粗粒度並列処理

分散メモリシステム上での粗粒度並列処理の主な手順を以下に示す．

1. **コンパイル** … 逐次プログラムの MT への分割，MT 間の制御フロー・データ依存解析，実行開始条件 [1]・データ到達条件の解析．
2. **実行** … 実行開始条件の検査・MT スケジューリング，データ到達条件の検査・データ通信．

2.2 積極的・投機的なプリロード

分散メモリシステム上での粗粒度並列処理では，プロセッサ間のデータ通信オーバヘッドを削減するための手段として「プリロード」が用いられる．通常は，実行開始条件が成立した MT で使用されるデータのうち，データ到達条件が成立したもののみがプリロードの対象となる．ただし，実行開始条件やデータ到達条件の状況によっては，通常とは異なるプリロードも考えられる．

本稿では，通常よりも大きな性能向上を期待できる「積極的なプリロード」および「投機的なプリロード」を適用した粗粒度タスク並列実行方式を提案する．

- **積極的なプリロード：** 実行確定条件のみが成立した MT をプロセッサに割り当て (プリアサイン)，その MT に対してデータ到達条件が成立したデータをプリロードする．
- **投機的なプリロード：** プリアサインされた MT に対して，データ到達条件が未成立のデータをプリロードする．つまり，データ転送のみを投機的に行い，MT の実行はその実行開始条件が成立した後に行われる．このとき，データ到達条件が不成立となり，プリロードされたデータが不要となる場合があり得る．

投機的なプリロードは，積極的なプリロードに比べて，プリロードのより一層の効果が見込める．

3 積極的・投機的なプリロードの実装方式

3.1 実装システム

積極的・投機的なプリロードを適用した粗粒度並列処理の実装を，各ノードに 2 つのプロセッサを持つ SMP クラスタ上に，MPI と OpenMP を用いて行った．

スケジューリングには集中スケジューラ方式を採用し，ノードのうち 1 つをスケジューラノード (SN)，その他のノードを MT 実行ノード (EN) とする．各ノードで 1 つのプロセスを起動し，MPI を用いたノード間通信を行う．また，各プロセス毎に OpenMP を用いて 2 つのスレッド (マスタ，スレーブ) を生成し，処理と通信とを個別のスレッドで実行することにより，これらのオーバラップ実行を実現する．このとき，スレッド間通信には共有変数を用いる．

Speculative Preload in Coarse Grain Task Parallel Processing on Distributed Memory System
Hiroshi NASU†, Hiroki HONDA‡and Toshitsugu YUBA‡
†The University of Electro-Communications
‡Graduate School of Information Systems, The University of Electro-Communications

各ノードで生成される 2 つのスレッドの機能を以下に示す.

- **SN -マスタ** … 通信コードの実行
  -スレーブ … グローバルスケジューラコードの実行
- **EN -マスタ** … 通信・MT 処理管理コードの実行
  -スレーブ … MT 処理コードの実行

実装の概要図を図 1 に示す.

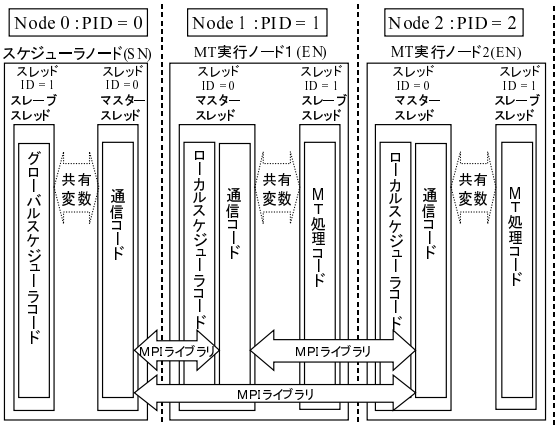

図 1: 実装の概要図

3.2 グローバルスケジューラの実装

実装した SN のグローバルスケジューラコードの主な機能を以下に示す．グローバルスケジューラは，最終 MT が終了するまで (1)〜(3) の実行を繰り返す.

(1) MT 終了と分岐方向決定の通知に伴う実行開始条件とデータ到達条件の更新

(2) 実行開始条件が成立した MT のプロセッサへの割り当て決定．MT 実行指示とデータ送信・受信指示の発行

(3) 実行確定条件のみが成立した MT のプリアサイン．積極的・投機的なプリロード指示の発行

積極的･投機的なプリロードによる通信が可能なデータは，下記の条件を満たすものとする.

1. データ生成元 MT が終了している
2. データ転送先 MT がプリアサインされている

グローバルスケジューラは 1.，2. の条件を満たすデータを随時検査し，プリロードの指示を発行する．その際，プリロードの対象となるデータの

$$\text{データ到達条件が} \begin{cases} True \text{ なら} & \text{積極的なプリロード} \\ False \text{ なら} & \text{投機的なプリロード} \end{cases}$$

の指示を発行する.

4 予備評価

積極的なプリロードの評価には SpecCFP95 の swim を C 言語で記述し直したプログラム (データサイズ 1024x1024) を，投機的なプリロードの評価にはプリロードの評価用に作成した例題プログラム (データサイズ 2048x2048, 2048x128) をそれぞれ用いた．MT 分割や粗粒度並列処理コードの生成は人手で行った．評価環境は各ノードに PentiumIII 866MHz を 2 つ搭載した SMP クラスタ (EN = 2 nodes) で，ノード間のネットワークには swim では Myrinet を，例題プログラムでは Ethernet を使用した.

swim と例題プログラムの実行結果をそれぞれ表 1 と表 2 に示す.

	実行時間	実行時間比	PL 効果
逐次実行	129.174 [sec]	1.00	—
PL 無し	114.990 [sec]	0.89	1.00
積極的な PL	99.951 [sec]	0.77	0.87

表 1: swim の実行結果 (EN = 2 nodes)

	実行時間	実行時間比	PL 効果
逐次実行	3.057 [sec]	1.00	—
PL 無し	2.514 [sec]	0.82	1.00
積極的な PL	2.162 [sec]	0.71	0.86
投機的な PL	1.837 [sec]	0.60	0.73

表 2: 例題プログラムの実行結果 (EN = 2 nodes)

表 1 から，swim では積極的なプリロードによって，通常よりも実行時間を 13 %削減できることがわかった．また，表 2 より，投機的なプリロードの機能が有効であることが確認できた.

5 おわりに

投機的なプリロードが有効な実アプリケーションでの評価や，EN 数が 2 ノードより多い場合のプリロード効果の検証をすることが今後の課題である.

参考文献

[1] 本多 弘樹，岩田 雅彦，笠原 博徳: “Fortran プログラム粗粒度タスク間の並列性検出手法”，電子情報通信学会論文誌 Vol.J75-D-I No.12，pp.951-960 (1990)

[2] 本多 弘樹，上田 哲平，深川 保，弓場 敏嗣: “分散メモリシステム上でのマクロデータフロー処理のためのデータ到達条件”，並列処理シンポジウム JSPP2002 論文集，pp.313-320 (2002)

[3] 深川 保，吉瀬 謙二，本多 弘樹，弓場 敏嗣: “分散メモリシステム上での OpenMP によるマクロデータフロー処理”，情報処理学会研究報告 Vol.2002(HPC-91-20)，pp.113-118 (2002)