

超並列プログラミング環境の検討

4H-5

市吉 伸行 関田大吾 西岡利博 吉光 宏

技術研究組合 新情報処理開発機構 超並列 MRI 研究室

1 はじめに

超並列計算機実現に向けたアーキテクチャ技術等が発展しつつあるが、超並列計算機普及のためには、プログラム開発の容易化が大きな課題である。超並列プログラムにおいては、逐次プログラムと比べて、並列アルゴリズムや負荷分散方式といった要素が加わっており、多数の実行主体の存在はデバッグや性能チューニングを複雑な作業としている。超並列 MRI 研究室では、RWC プロジェクトにおける超並列ソフトウェア分野の研究の一環として超並列プログラミング環境の研究に着手しており、プログラミング支援ツールの観点からこの問題への対処を検討している。本稿ではこれまでの検討状況を報告する。

2 超並列プログラム開発の問題点

一般にプログラム開発では、設計、コーディング、デバッグ、性能改善といったステップを経る。逐次プログラムと比べた場合、並列プログラムでは、プログラムがより複雑であること、(MIMD 型の場合)プログラム実行に再現性が保証されないこと、負荷バランス、通信オーバーヘッドなどの性能決定要因の増加等の難しさが加わっている。また、超並列マシン向けのプログラムのデバッグ作業は、

1. 単一プロセッサでの初期デバッグ
(主に、基本アルゴリズムのデバッグ)
2. 少数プロセッサでの初期デバッグ
(主に、並列論理デバッグ)
3. 多数プロセッサでの本格的な初期デバッグ
(主に、並列性能デバッグ)

Toward a Massively Parallel Programming Environment

Nobuyuki ICHIYOSHI, Daigo SEKITA, Toshihiro NISHIOKA, and Hiroshi YOSHIMITSU

Massively Parallel MRI Lab., Real World Computing Partnership

2-3-6 Otemachi, Chiyoda-ku, Tokyo 100, JAPAN

のようにプロセッサ数を変えながら色々な側面を試験するようなものになるであろう。並列プログラム開発環境は、そのような事情に配慮し、各段階に合った適切な支援を行なえることが望ましい。

さらに、初期デバッグによって、バグの数の90%ないしそれ以上が発見・修正されたとしても、それ以降のテスト・ランやプロダクション・ランにおいて発生するであろう取り切れてないバグへの対策も考えなくてはならない。

超並列 MRI 研究室では、RWC プロジェクトで開発する超並列マシンおよびプログラミング言語をターゲットとして、プログラミング支援ツール(特に、デバッガ[4]、プロファイラ)による上記問題への対処を研究している。

3 デバッガの検討

逐次プログラムのバグの追求においては、バグが顕在化する箇所の手前の1つないし複数の適当な箇所にブレークポイントやトレースポイントを設定して、途中状態を検証することによってバグ発生候補箇所を狭めるという「繰り返しデバッグ(cyclic debugging)」がよく用いられる。

ところが、並列プログラムでは、ハードウェア構成要素間の動作の非同期性に起因するプログラム動作の非決定性があるために、バグ動作が必ずしも再現せず、上記の技法の適用が難しい。そこで、非決定的なプログラムの実行を再現させようという、再現実行(replay)機構の研究が従来からなされてきたが[1]、実際のプログラミング環境で実現されたものはまだ少ない。特に、研究途上にある細粒度超並列マシン向けの方式の研究事例は、まだ見受けられない。当研究室では細粒度超並列マシン RWC-1 [2] 向けの再現実行の実現方法を検討中である [3]。

非同期性に起因する別の問題として、ブレークポイントの意味の問題がある。プログラム実行中に指定した地点でプログラムを停止させてプログラムの状態を調べるブレークポイントの機能は非常に便利だが、非同期並列プログラムにおいては、ブレーク対象・条件の指定方法、ブレークポイントでの停止

方法として何が適当であるかが、それほど明らかでない。デバッグ容易性や実現可能性の観点から、これらを検討すべきであろう。

その他、デバッガからの情報の提示方法などにも配慮が必要である。例えば、トレースポイントからの出力が多数のプロセッサから非同期にユーザに提示されるのは好ましくないであろう。トレースに関しては、実行時にトレースファイルを生成し、ポストモーテムに解析するというイベントトレース機能も検討している [5]。

4 プロファイラの検討

超並列マシンではピーク性能と実効性能との乖離が一つの大きな問題である。期待した性能が得られない時、どこに問題点があるかつき止めるための性能プロファイリングは重要である。プロファイルデータ収集メカニズムの研究は多くなされてきているが、得られた性能データを具体的にどのように解析すべきかという方法論はまだ未熟である。

当研究室では、プロセッサ稼働率、通信頻度、ネットワーク混雑度といった基本的性能データ収集の機能を実現し、その上で、性能データ解析/提示ツールを開発して行くことを考えている。

具体的には、インストルメンテーション用コードをソースコード中に挿入することによって、スレッド起動回数やパケット送出回数を常時記録しておく、適当な時間間隔で RWC-1 のプロセススイッチ機能により、実行中プログラム全体に割り込みをかけて、各プロセッサにおいてカウンタ値やプログラムカウンタ値をログ領域に記録する方式を検討している。(RWC-1 では高速なプロセススイッチを可能とするために、全系(より正確にはパーティション)を一斉に止めて、ネットワーク中のパケットは通過中のプロセッサに吸い上げるドレイン機能を持っている。)十分に高速なドレインが実現される場合、上記割り込み時にネットワーク混雑状況もサンプリングできるであろう。

5 統合型デバッグ/プロファイル環境

前記のイベントトレースの解析で何か問題点が発見されると、さらに詳しいイベントトレースで追求したいかも知れない。また、性能プロファイリングにおいて発見された負荷分散の不具合をデバッガに

よって追求したいこともあろう。このように一つの現象を追求するために同一内容の実行を複数回繰り返したいのは通常のことだが、実行の度にプログラムの振舞いが異なると不便であろう。

すると、繰り返しデバッグ以外にも、イベントトレースやプロファイリングも再現実行下で行なえると望ましいと考えられる。すなわち、再現実行メカニズムを基礎として「統合型」のデバッグ/プロファイル環境を実現するということである。そのためには、記録実行の性質が通常の実行のそれとかけ離れなれていないことが必要である(デバッグ目的でもそうだが)。一方、一旦そのような再現実行メカニズムが実現されると、再現実行下のイベントトレースやプロファイリングは調べたい実行に影響を与えないというメリットもある。

6 おわりに

超並列 MRI 研究室では、平成 5 年度より 4 年間の計画で、RWC プロジェクトで開発する超並列マシン RWC-1 向けのプログラム開発支援ツールを研究開発し、その評価に基づいて、その後のプログラミング環境の研究開発を継続する予定である。4 年間の前期では、直面する問題の解決/緩和を目指したツールの研究開発を行ない、後期では、あるべきプログラミング環境の観点から、より直観的なプログラミングの支援、部品化、自動化など、さらに進んだテーマの研究を行ないたいと考えている。

参考文献

- [1] 山田剛. 並列処理システムにおけるプログラムデバッグ. 情報処理, Vol. 34, No. 9, 1993.
- [2] 坂井修一, 他. 超並列計算機 RWC-1 の基本構想. 並列処理シンポジウム JSPP'93, pp. 87-94, 1993.
- [3] 市吉伸行, 他. 超並列マシン RWC-1 における再現実行方式の検討. 第 48 回情処全大 4H-8, 1994.
- [4] 関田大吾, 他. 超並列プログラムでのデバギングの考察. 第 48 回情処全大 4H-7, 1994.
- [5] 西岡利博, 他. 並列プログラムのポストモーテム型デバッグ環境に関する考察. 第 48 回情処全大 4H-6, 1994.