

教育向け並列プログラミング言語における ステップ実行を用いた理解支援ツール

岸本 優斗^{*1}, 近藤 一輝^{*1}, 中川 翔太^{*1}, 水谷 泰治^{*1}

Yuto KISHIMOTO^{*1}, Kazuki KONDO^{*1}, Shota NAKAGAWA^{*1}, Yasuharu MIZUTANI^{*1}

^{*1}大阪工業大学 情報科学部

^{*1}Faculty of Information Science and Technology, Osaka Institute of Technology

Email: e1b14035@st.oit.ac.jp

あらまし：本研究では教育用並列プログラミング言語で記述したプログラムの理解を支援するツールを提案する。本研究の目的は並列プログラミングの初学者に対して並列処理の理解を支援することである。この目的を達成するために、並列プログラムをステップ実行しながら通信状況を図形の描画によって可視化するツールを作成した。本発表では、ツールの内容、サンプルコードを用いた実験の結果、及び結果から得られた考察について報告する。

キーワード：並列処理, ステップ実行, 可視化, JavaScript

1. はじめに

計算処理を複数のプロセッサに分担させ、処理速度を向上させる技術を並列処理という。並列処理を行うプログラミング言語やライブラリとして MPI⁽¹⁾ や CUDA⁽²⁾等が挙げられる。著者らが所属している研究室では、これらの言語の独特な記法を抽象化した教育向け並列プログラミング言語(以降、T言語と表記)が考案されている。しかしながら、並列プログラミングはプロセス間の通信を考慮する必要があり、T言語を利用しても依然として学習の難易度が高い。

そこで本研究では、通信部分の可視化及びステップ実行によってT言語による並列プログラムの実行状況の理解を支援するツール(以降、本ツールと表記)を開発した。

2. T言語

T言語は、既存の並列プログラミング言語の独特な部分を排除し、各言語に共通した操作を抽出し、その共通操作のみで並列プログラムを記述・実行できる教育用並列プログラミング言語^(3,4,5)である。T言語はC言語をベースにしているため、C言語の殆どの機能を利用することができる。それに加え、並列処理に必要な関数が追加されている。

T言語では並列処理を行うため、T_Run という関数を使用する。T_Run 関数を呼び出すと、指定した関数が並列に呼び出される。図1に、T_Run 関数による並列呼び出しの様子を示す。

図1の左部はT言語のソースコードを表しており、右図は処理の流れを表している。T_Run 関数の第1引数は並列呼び出しする関数の数を表しており、第2引数は並列呼び出しする関数名を表している。図1ではmain関数でT_Run関数を呼び出し、2つの関数fが並列呼び出しされる。ここで、並列プログラムにおいて、並列呼び出しされる関数1つをプロセスと呼ぶ。

並列処理ではデータのやり取りや同期処理を行うことがあるため、T言語には通信や同期処理を行う関数が追加されている。表1はT言語において通信

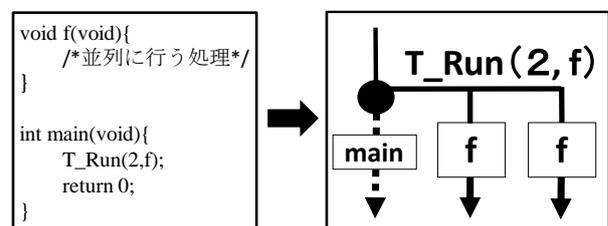


図1 T言語による並列処理の様子

表1 描画対象の関数

関数名	関数の説明
T_Send	特定のプロセスに値を送信する。
T_Recv	特定のプロセスから値を受信する。
T_Bcast	指定したプロセスが持つ値を他のプロセスに配る。
T_Barrier	各プロセスがBarrier処理を行うまで待機し、各プロセスの足並みを揃える。

と同期処理を行う関数を表している。

表1のT_Send, T_Recv, T_Bcastはプロセス間でデータの通信を行う関数である。T_Barrierは、プロセスの同期処理を行い、プロセスを実行するタイミングを揃える関数である。

3. 理解支援ツール

本ツールはT言語を用いて並列プログラミングを初めて学習する者を対象としている。本ツールは、T言語で記述されたプログラムの実行状況の可視化を実現するため、Webブラウザ上で動作させる方法を採用している。このため、本ツールではJavaScriptを使用している。

3.1 ツールの外観

図2に本ツールの外観を示す。図2の上部は左から順に、プログラム入力部、通信履歴部、標準出力部を表しており、下部は描画領域を表している。

プログラム入力部では、T言語のソースプログラムを入力してからボタンによって操作を行う。「Run」ボタンを押すと、入力したT言語のソースコードをJavaScriptのコードに変換する。プログラムに誤りが

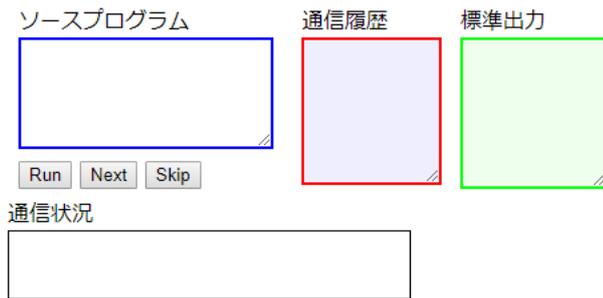


図2 ツールの外観

なければ、プログラムは実行可能状態となり、「Next」ボタンを押すと描画が発生する文までプログラムを実行する。「Skip」ボタンはスキップ機能となっており、プログラムの最後まで実行する。

通信履歴部では実行された通信内容を表示する。送受信の結果や、待機状態にあるプロセスの情報をこの部分に出力する。標準出力部では、printfなどのプログラムの実行結果を表示する。

3.2 ステップ実行

ステップ実行は「Next」ボタンを押すことによって実行される。ステップ実行を実現するために、本研究では JavaScript にあるジェネレータ関数という機能に着目した。これは、ジェネレータ関数内部で演算子『yield』を読み取るとその時点で実行を一時的に停止し、その関数の next()メソッドが実行されると停止した箇所から実行を再開するという機能である。この機能を用いて通信命令の度にプログラムを一時停止するステップ実行機能を実装し、停止する度に通信命令の詳細を表示する機能を実装した。

一般的なステップ実行は、コードを一文ずつ実行していくが、本ツールでは通信部分の可視化に着目しているため、T_Send 関数、T_Recv 関数、T_Bcast 関数及び T_Barrier 関数のいずれかの部分まで読み進めるとプログラムを一時中断する。

3.3 並列処理のシミュレート

本ツールでは、並列呼び出しされる関数を JavaScript で実行できる形に変換し、ブラウザ上で実行することで並列処理のシミュレートを行っている。

並列呼び出しされる関数はジェネレータ関数に変換される。そして、そのジェネレータ関数は、実行を制御するためのオブジェクトに格納する。このオブジェクトは並列呼び出しする関数の数だけ作成する。この状態で「Next」ボタンを1回押すと、それぞれのオブジェクトに格納された関数の『yield』部分まで実行される。これにより、複数の関数が並行して実行される様子を再現した。

3.4 描画領域

本ツールでは、T_Send 関数、T_Recv 関数、T_Bcast 関数及び T_Barrier 関数の内容を描画している。図3に、実行中の描画領域を示す。

描画領域の左部に表示している数字は、プロセスを識別する番号(以下、プロセス番号)を表している。

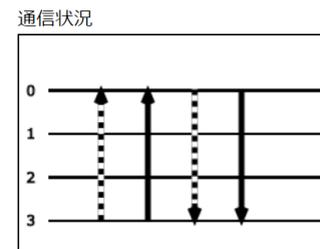


図3 実行中の描画領域

図3は、プロセス0とプロセス3において送受信を行う様子を表している。描画領域に表示している矢印は、Next ボタンを押すことによって表示する。破線で表示している矢印は T_Send 関数を表しており、実線で表示している矢印は T_Recv 関数を表している。

4. 実験結果と考察

本研究では、図3で示した様な送受信を繰り返す行うプログラムを対象に動作確認を行った。実験の結果、並列処理の動きをツールでシミュレートすることができ、T 言語プログラムで行う通信と同期処理の様子を描画で表すことができた。しかし、通信回数が818回を超えると領域内に描画を行うことが不可能となり、送受信回数が約8000以上となるとツールが停止するエラーが発生した。

また、スキップ機能に要する時間は通信履歴を非表示にすると大きく短縮することが分かった。この結果から、テキストエリアに文字を出力するという処理は高負荷であると考えられる。

5. まとめ

本研究では、並列処理の様子をステップ実行と可視化によって理解を支援するツールを開発した。今後は、利用者にアンケート調査を行い、可視化機能の改善等を行う。

謝辞

本研究は JSPS 科研費 JP15K21511, JP15K12008, JP17K00500 の助成を受けたものである。

参考文献

- (1) <http://www.mpi-forum.org/>
- (2) <https://developer.nvidia.com/cuda-zone/>
- (3) 田中寛章,藤井健太,磯淵郁也,水谷泰治."複数のハードウェアでの共通操作に着目した教育用並列プログラミング言語の提案".第78回情報処理学会全国大会,5ZC-02,(2016-03).
- (4) 田中寛章,水谷泰治."教育用並列プログラミング言語におけるCPU制御による見かけ上の速度向上率の増加".第79回情報処理学会全国大会,3ZC-08,(2017-03).
- (5) 水谷泰治,西口敏司,橋本 渉."教育用並列プログラミング言語におけるCPU制御による見かけ上の速度向上率の増加".第80回情報処理学会全国大会,(2018-03 予定).