

平成 30 年 6 月 20 日現在

機関番号：34406

研究種目：若手研究(B)

研究期間：2015～2017

課題番号：15K21511

研究課題名(和文)ハードウェア非依存な教育向け並列プログラム言語とその処理系の研究

研究課題名(英文)Hardware-independent Programming Language for Parallel Programming Education

研究代表者

水谷 泰治(Mizutani, Yasuharu)

大阪工業大学・情報科学部・准教授

研究者番号：10411414

交付決定額(研究期間全体)：(直接経費) 2,700,000円

研究成果の概要(和文)：本研究では、既存の代表的な複数の並列プログラミングからハードウェアに依存する煩雑な記述を取り除き、並列プログラミングに必要な共通的な操作だけで並列プログラムを記述できる教育用並列プログラミング言語、処理系、および可視化ツールを開発した。また、この処理系を用いて並列プログラミングの初学者に対して並列プログラミング演習を実施し、アンケート評価を実施した。

研究成果の概要(英文)：We proposed an educational parallel programming language, its processing system, and visualization tool. This language enables us to write parallel program without environment-dependent complex programming. We also applied the language and the processing system to a programming exercise for parallel programming beginners.

研究分野：並列処理

キーワード：教育工学 学習支援 並列プログラミング 言語処理系

## 1. 研究開始当初の背景

CPUのマルチコア化やGPUを用いた汎用計算など、並列処理の専門家ではない者の身近にまで並列処理可能なハードウェアが普及している。しかし、並列プログラムの開発はハードウェアの詳細を理解した上でその詳細を意識した独特な記述が必要であり難しい。さらに、近年の並列処理装置は複数のハードウェアが混在する傾向にある。このような環境を活用するには、同時に複数のハードウェアを意識して並列プログラムを記述せねばならない。そのため、現状では並列プログラミングを学び始めるには、予め多くのハードウェアの知識と並列化のコツを修得する必要がある。これは並列プログラミングを学ぶ際の大きな障害であり、並列処理技術の普及が依然として進んでいない大きな原因となっている。

一方、並列プログラムの開発に関する既存のツールでは、プログラムに注釈を挿入して並列化を指示する書き方と、並列処理させるプログラムや通信命令を細かく記述できる書き方がある。これら書き方は、並列計算装置の構造を把握しておく必要があるなど、並列プログラミングの経験者を想定しており、初学者にとっては敷居が高いものが多い。また、PC クラスタや GPU など特定の装置にのみしか対応していないものが多い。

## 2. 研究の目的

本研究では各並列計算装置における並列プログラムの書き方を抽象化し、並列プログラミングの本質を学ぶのに適した教育用並列プログラミング言語を開発する。それと同時に、その言語で記述したプログラムを実際の装置で実行できる処理系も開発する。特に、本研究では、特定のハードウェアに特化せず、多様なハードウェアに適用可能であるような統一的な記述で並列プログラムを開発できる仕組みの確立を図る。また、提案した教育用並列プログラミング言語とその処理系を用いて、初学者を対象としたプログラミング演習を実施し、その効果を評価する。

## 3. 研究の方法

本研究では以下の3点について取り組む。

### (1) 教育用並列プログラミング言語の提案

各装置から並列プログラムの記述に必要な共通的な記述スタイルを抽出し、そのスタイルのみで並列プログラムを記述する方法を検討する。特に、データ分散、プロセス・スレッドの識別と連係、排他制御、同期など、並列プログラミングの本質となる概念を整理し、各装置に適用できる記述方法を検討する。

### (2) 実機で動作する処理系の開発

(1)で検討した教育用並列プログラミング言語で記述したプログラムを実際の並列計算装置で実行できる処理系を開発する。これにより、実機を用いた本格的な並列処理を可能にする。この実現のためには、対象装置ごとに異なるコンパイラでコンパイル可能なプログラムに変換し、各装置上でプログラムを動作させるときの各々の連係手法について検討する必要がある。

### (3) 評価実験

提案した教育用並列プログラミング言語の処理系を用いて、並列プログラミングの初学者に対してプログラミング演習を実施し、アンケートによる評価を行う。ここで並列プログラミングの初学者とは、基本的な逐次プログラミングを学習しており、並列プログラミングを初めて学ぶ大学4年生とする。

## 4. 研究成果

### (1) 並列プログラミング言語の提案

単に並列プログラミングを行うだけなら、OpenMP や CilkPlus のようなコーディングを単純化するツールがある。しかし、ハードウェアの性能を最大限に引き出すには、ハードウェアに合わせたコーディングを行うことが有利である。そのため、並列プログラミングの学習者は将来的にハードウェアに合わせた並列プログラムをコーディングできるようになることが望ましい。そこで、本研究では教育用並列プログラミングを考案するにあたり、MPI, Pthreads, Cuda といった代表的な並列プログラミング言語やライブラリを対象とし、それらの独特な書き方を簡略化し、共通的な操作関数を用いて並列プログラムを記述する方式とした。

図1に本研究で考案した教育用並列プログラミング言語(以降、T言語と呼ぶ)で記述したプログラムとその実行のイメージを示す。図1左部のプログラムでは、main関数においてT\_Run関数を用いて関数fを4つ並列に呼び出している。T言語プログラムはtccrun コマンドによりコンパイルと実行を行う。図1右部は左部のプログラムの実行の

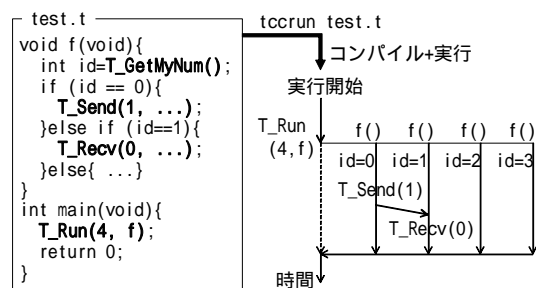


図1. T言語のプログラムと実行のイメージ

イメージを表す．並列に呼び出された関数の  
 中では T\_GetMyNum 関数によって関数呼び出  
 しの各実体に対して一意な識別番号を取得  
 できる．T\_Send 関数と T\_Recv 関数は第 1 引  
 数でそれぞれ送信先と送信元の識別番号を  
 指定し，第 2 引数以降に送信・受信バッファ  
 の先頭アドレスとサイズ(bytes)を指定する．  
 これらの操作関数は MPI や CUDA, Pthread など  
 の各環境から抽出・簡略化したものであり，  
 各環境におけるプログラミングの煩雑さの  
 軽減を意図している．

一般的に並列プログラムの記述方法とし  
 て分散メモリ型と共有メモリ型に分類でき，  
 それぞれプログラミングの仕方が異なる．T  
 言語ではこれらを 2 つの記述方法をそれぞれ  
 記述できるようにした．前述の T\_Send 関数  
 や T\_Recv 関数を用いた記述は分散メモリ型  
 である．一方，共有メモリ型の記述として，  
 T 言語は排他制御を行うための T\_Lock 関数と  
 T\_Unlock 関数を提供する．

## (2) 処理系の開発

T 言語で記述したプログラムを既存の並列  
 プログラミング言語やライブラリを用いた  
 プログラムに変換する処理系を開発した．

この処理系では，分散メモリ型で記述した  
 T 言語プログラムは MPI を用いた C 言語プロ  
 グラム (MPI プログラム) に，共有メモリ型で  
 記述した T 言語プログラムは Pthread を用い  
 た C 言語プログラムに変換する．

図 2 に変換の手順を示す．共有メモリ型の  
 場合は共有メモリ型用の処理系 (図 2 右上部)  
 を用い，処理系がもつライブラリと共にコン  
 パイルすることで，T 言語関数は Pthread 関

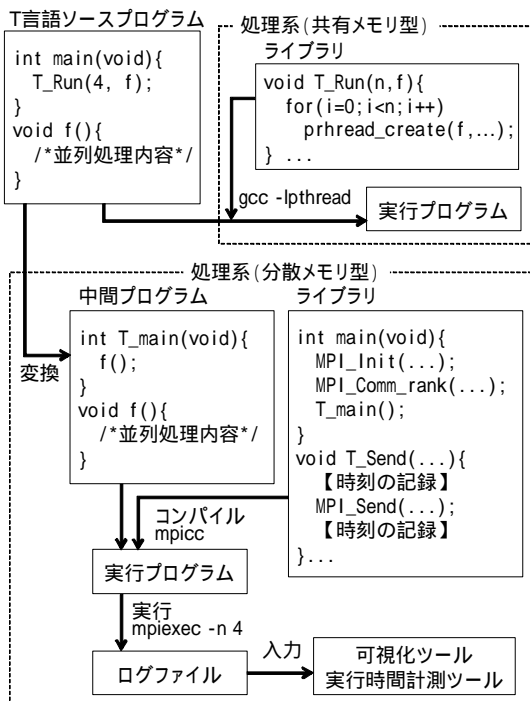


図 2. T 言語の処理系

数のラッパーとして機能する．

分散メモリ型の場合も同様に分散メモリ  
 型用の処理系 (図 2 下部) がもつライブラリ  
 とコンパイルする．このとき，ソースファ  
 イル内の T\_Run 関数呼び出しを抽出し，第 1 引  
 数に指定した並列呼出数を mpiexec コマンドで  
 並列実行するプロセス数に指定する．また，  
 T\_Run 関数は並列実行関数の呼び出しに置き  
 換える．さらに，main 関数を T\_main 関数へ  
 と名前を変更し，予めライブラリ内で定義済  
 の main 関数内で呼び出す．この main 関数で  
 は MPI の初期化処理などを含んでいる．この  
 ように MPI プログラムに変形することで並列  
 処理を実現した．

## 通信の可視化と性能測定の実現

分散メモリ型用の処理系では，並列プロ  
 グラムの実行における通信の可視化と，性能測  
 定を支援する機能を開発した．

並列プログラムの実行時に実行状況に関  
 するログを記録しており，そのログに基づい  
 て通信状況の可視化と指定箇所の実行時間  
 の測定を行う．図 3 に，通信状況の可視化の  
 例を示す．図 3 は，縦軸がプロセス番号，横  
 軸に時間の流れを表しており，各プロセスが  
 どのようなタイミングで通信を行ったかを  
 表現している．

また，図 4 のように T 言語のプログラムに  
 特殊コメントを挿入することで性能測定を  
 支援する．図 4 ではマクロ N の定義を特殊な  
 コメントで囲っている．このようなコメント  
 を挿入したプログラムを T 言語処理系で実行  
 すると，実行時にマクロ N を変動させる値の

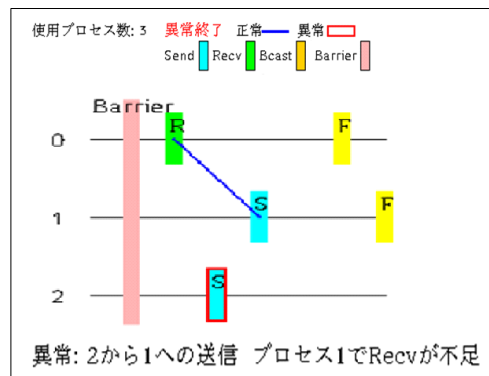


図 3. 通信関数の可視化

```

#!start_change
#define N 1024
#!end_change
int main(void){
  T_Run(4, f);
}
void f(){
  ...
}
  
```

図 4. 性能測定支援の特殊コメント

入力を促され、入力した N の値について再コンパイルと実行を行う。また、このときに並列処理に要した実行時間を自動的に計測し、標準出力に表示する。これらの機能により、並列プログラミングにおける性能測定に要する時間の短縮する。

この機能による性能測定に要する時間の变化について実験を行った。実験では、ガウスの消去法によって連立方程式を解く並列プログラムを作成し、並列実行したときの各プロセスの実行時間を取得した。この時、計測した実行時間を表にまとめる作業を、手動で行った時と本機能を使用した時に要した時間を比較した。

行列サイズを 256, 512, 1024, 2048 と変えて計測するとき、手動の場合、プログラムに実行時間を計測するコードを追加する必要がある。また、毎回行列サイズを手動で変更して、コンパイル・実行し、端末に表示された出力結果を手作業でまとめる必要がある。これら一連の作業に約 390 秒を要した。一方、本機能を使用すると、初めにプログラムに特殊コメントを記述して、端末上に変更する行列サイズを入力するだけでよく、約 24 秒で性能測定を完了した。これより、本機能により性能測定に要する手間を削減できることがわかった。

### ブラウザ上でステップ実行する機能

並列プログラムの動作のイメージを掴みやすくすることを目的に、T 言語のプログラムをステップ実行し、その実行に合わせて実行状況を可視化するツールを開発した。本ツールは実行状況の可視化を実現するために Web ブラウザ上で動作させる方法を採用した。

図 5 に Web ブラウザに表示される本ツールの外観を示す。図 5 の上段はプログラム入力部、中段は通信履歴部と標準出力部、下段は描画領域を表している。

図 6 に本ツールの処理の流れを示す。まず、プログラム入力部に T 言語のソースプログラムを入力する。その後、「Run」ボタンを押すとソースプログラムを Javascript のプログラムに変換する。プログラムに誤りがなければ



図 5. ツールの外観

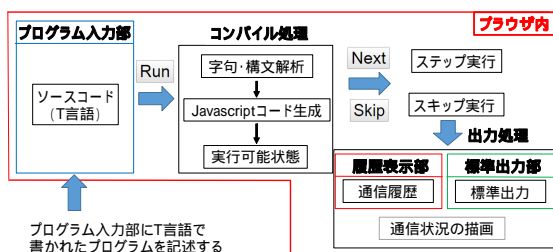


図 6. ツールの処理の流れ

ばプログラムは実行可能状態となる。「Next」ボタンを押すと実行状況を图示しながらステップ実行する。「Skip」ボタンはプログラムを最後まで実行する。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。標準出力部には printf などのプログラムの実行結果を表示する。

ステップ実行は Javascript のジェネレータ関数という機能を用いて実現した。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、その関数を表すオブジェクトの next() メソッドを実行す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。

### (3) 提案した言語および処理系の評価

2017/4/1(土)から 2017/6/17(土)までの期間に並列処理を専門分野とする研究室に配属された大学 4 年生 8 名を対象に並列プログラミング演習を実施した。この 8 名は C や Java によるプログラミングは学習しているが、並列プログラミングの学習は本演習が初めてである。

この演習は(2)で述べた分散メモリ型の処理系を用いて実施した。演習課題は表 1 に示した 5 つであり、最初の 4 つは簡単な通信プログラム、行列演算、画像処理などのプログラムを T 言語で開発する。最後の課題では T 言語で開発したプログラムを C 言語と MPI (高性能計算の分野において標準的に用いられているライブラリ) による再実装、および CUDA (GPU 上の並列処理における標準的な処理系) を用いた並列プログラムの作成である。

演習終了後に表 2 と表 3 に示すアンケートを実施した。表 4 にアンケート結果を示す。アンケート結果より、T 言語の単純さに対する肯定的な回答が多く得られた。具体的には、「シンプルであるため必要な処理を考えやすかった」、「MPI や CUDA に比べて関数等が直感的でわかりやすく学習のハードルが下がった」といった回答や、それに類する回答が多く得られた。また、最初に T 言語で並列プログラムを作成してから最後に実際に用いられている処理系を用いて作成する形式については、「T 言語によって基本的な考え方がある程度理解できているため MPI や CUDA の特有の処理も理解しやすかった」といった回

表 1. 課題の一覧

課題	期間	概要	言語
1	1週	簡単な通信プログラム	T言語
2	2週	行列積(データ分散無し)	T言語
3	2週	行列積(データ分散有り)	T言語
4	3週	(選択) フラクタル画像生成	T言語
		ガウスの消去法	T言語
5	2週	課題1-4をMPIで再実装	C + MPI
		簡単なGPUプログラム	CUDA

表 2. 質問項目の一覧

1	T言語における並列プログラミングにおいて、T_Run関数を用いた関数の実体が複数呼び出されることで並列処理を行うことの理解の難易度はどうでしたか？
2	T_Run関数で呼び出された各関数はそれぞれが独立に動作しており、それぞれ個別の識別番号が割り振られているという概念の理解の難易度はどうでしたか？
3	T_Send関数とT_Recv関数の理解の難易度はどうでしたか？
4	tccrunコマンドの使い方の難易度はどうでしたか？
5	T言語プログラムの可視化の機能は使用しましたか？
6	MPIやCUDAを用いるよりもT言語を用いることで並列プログラミングの学習のハードルは下がったと思いますか？
7	課題5でMPIとCUDAを使ったプログラムを作成する際、課題4までで用いたT言語で役に立ったことや役に立たなかったことがあれば教えてください。
8	もし、T言語を用いずに最初からMPIまたはCUDAを用いて同じ課題の並列プログラミング演習を行った場合、全体的な学習負荷はどうなると思いますか？
9	総合的に考えて、課題1から4までをT言語で学び、最後にMPI やCUDA を学ぶというスタイルは良かったと思いますか？

表 3. 各質問の回答の選択肢

1-4	(1)簡単だった (2)どちらとも言えない (3)難しかった
5	(1)よく(4回程度以上)使用した (2)少し(1-3回程度)使用した (3)全く使用しなかった
6	(1)思う (2)どちらとも言えない (3)思わない
7	選択肢は無し
8	(1)T言語よりも苦労すると思う (2)T言語とそれほど変わらないと思う (3)T言語よりも簡単に取組めると思う
9	(1)良かった (2)どちらとも言えない (3)良くなかった

表 4. アンケート結果

1-4	(1)簡単だった (2)どちらとも言えない (3)難しかった
5	(1)よく(4回程度以上)使用した (2)少し(1-3回程度)使用した (3)全く使用しなかった
6	(1)思う (2)どちらとも言えない (3)思わない
7	選択肢は無し
8	(1)T言語よりも苦労すると思う (2)T言語とそれほど変わらないと思う (3)T言語よりも簡単に取組めると思う
9	(1)良かった (2)どちらとも言えない (3)良くなかった

答が得られた。これより、T 言語を用いることで並列プログラミングの学習負荷の軽減に一定の効果はあったものと考えられる。

## 5. 主な発表論文等

(学会発表)(計7件)

水谷泰治, 西口敏司, 橋本渉. "並列プログラミングの学習における教育用並列プログラミング言語の適用". 第 80 回情報処理学会全国大会, 2018-03

岸本優斗, 近藤一輝, 中川翔太, 水谷泰治. "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". 教育システム情報学会学生研究発表会(関西地区), 2018-02

水谷泰治, 中本翔太, 安田耕記, 西口敏司, 橋本渉. "C 言語の初学者を対象としたプログラム実行状況可視化ツールの評価". 第 42 回教育システム情報学会全国大会, 2017-08

箕畑宏宣, 徳久三四郎, 西口敏司, 橋本渉, 水谷泰治. "並列プログラミングの学習において通信の可視化と性能測定を支援するツールの提案". 電子情報通信学会 2017 総合大会, 2017-03

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, 2017-03

吉本拓斗, 三村明, 辻賢紀, 古本顕吾, 水谷泰治. "C 言語の初学者を対象としたプログラム実行状況の理解支援ツール". 電子情報通信学会 2016 総合大会, 2016-03

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, 2016-03

## 6. 研究組織

### (1) 研究代表者

水谷 泰治 (MIZUTANI, Yasuharu)  
大阪工業大学・情報科学部・准教授  
研究者番号: 10411414

### (2) 研究協力者

西口 敏司 (NISHIGUCHI, Satoshi)  
大阪工業大学・情報科学部・准教授  
研究者番号: 80362565

橋本 渉 (HASHIMOTO, Wataru)  
大阪工業大学・情報科学部・准教授  
研究者番号: 80323278