

# 東京大学のスーパーコンピュータを用いた並列プログラミング教育

—工学部・工学系研究科共通科目「スパコンプログラミング1およびI」  
(2007年度夏学期)を通じて—

片桐 孝洋

東京大学情報基盤センター 特任准教授

## 1. はじめに

東京大学情報基盤センター（以降、センター）では、スーパーコンピュータの潜在的な新規ユーザである、東京大学工学部を主とする学部学生と大学院生に対して、センターのスーパーコンピュータの啓蒙と、ユーザの研究分野における高性能並列プログラム開発能力の長期的な支援を行なう目的で、工学部および工学系研究科の共通科目「スパコンプログラミング1およびI」を開講している。本科目は、工学部や工学系研究科以外の学生も受講可能である。

本講義の受講生に対し、半年間有効となるセンターのスーパーコンピュータ（HITACHI SR11000/J2）のアカウントを無料で発行して演習を行っている。これは、スーパーコンピューティング部門のサービス「教育利用（試行）」の一環である[1]。

本報告は、2007年度夏学期（2007年4月～2007年7月）に行った講義の報告である。

## 2. C言語によるMPIを用いた並列プログラミング教育

### (1) 講義の方針

本講義を開講するに当たり、以下の指針を示した。

*「高性能計算を学ぶためには、計算機アーキテクチャに始まり、コンパイラやOSといったシステムソフトウェア、さらに扱っているアプリケーションのアルゴリズムに至る広範な階層の知識が必要となる。講義でこれらすべてを扱うことはできない。そこで、厳選された実用的な課題について講義と演習を行う。本講義は、従来講義のように広い知識の獲得を目指すものではない。実際に高性能プログラムを基盤センターのスーパーコンピュータ上で開発できるという、実用的でかつ、研究者として生き残るために必須な技能の習得を目指すものである。この技能の習得により、受講者の研究を格段に進展させることを目標とする。」*

すなわち、MPI(Message Passing Interface)を用いた並列プログラミングにおいて、MPIの機能を網羅的に紹介する従来のテキスト[2]のような方針ではなく、センターユーザの多数を占めるであろう数値シミュレーション研究者に必要な最低限の実装知識と、センターのスーパーコンピュータを利用するための最低限の技術の習得を目的にした。この理由は、以下のとおりである。

- きめ細かな演習を行うには、講義時間が少ないこと（90分×13回程度）
- 網羅的な講義では知識は増えるが、実際の研究に使えない場合が多いこと

また、東京大学理学系研究科において、「教育利用（試行）」制度を利用した並列プログラミング教育[3]も既に開始されている。本講義との差異は、本講義が数値計算ライブラリ（行列一行

列積、LU 分解法)・コンピュータサイエンス寄りの構成であることに對し、[3]は数値シミュレーション解法(差分法、有限要素法、境界要素法)・計算科学寄りの構成となっている点である。

## (2) 並列プログラミング教育の方針

並列プログラミング教育における最も重要な概念は、SIMD (Single Instruction Multiple Data stream) の概念であるという仮定のもとに教材を作成した。SIMD とは、並列計算機の種類の一つである。以下のように、同一の命令(たとえば加算命令)がなされるが、加算する対象のデータは、並列計算機の構成要素であるプロセッサ・エレメント (Processor Element, PE) で異なるというモデルである。



図1 SIMD の概念

本演習では、高並列で高性能な並列プログラムを作成する際に、どの並列計算機環境でも使うことができる MPI ライブラリを用いる。また、高並列な計算機環境はすべて、分散メモリ型の構成となっている。MPI は分散メモリ型の並列計算機環境に向く通信ライブラリである。このことから図1に示される各データが、並列計算機を構成する PE に分散されており、図1で示される命令を定める仕様が MPI で規定されているので、SIMD の概念さえ身につけば MPI を用いて高並列かつ高性能な並列プログラムを開発できるようになる。

しかし、SIMD の概念を並列化をしたことがない受講生に教えることは容易ではない。理由は、いままで扱ってきた逐次プログラム(と受講生の頭にある「逐次」プログラミングモデル)との乖離があるからである。

そこで、SIMD の概念が自然に身につくように、以下のような段階を経て並列化ができる方針を立てた：

- **段階1** 並列環境になじむため、簡単な並列プログラムを実行する演習を行う。また、並列プログラムを眺めてみる。
- **段階2** 逐次プログラムを実行してみる。その後、逐次プログラムを解析してみる。
- **段階3** 逐次プログラムに存在するループについて、ある変更を施すだけで並列化ができる演習をさせる。
- **段階4** 逐次プログラムに存在するループに、ある変更を施し、かつ単純な MPI 関数を記述することで並列化できる演習をさせる。
- **段階5** データ構造の変更、多数の MPI 関数を記述することで並列化できる演習をさせる。

なお、ここで指摘する並列化のための変更とは、並列化をするために汎用的に必要となる記述

であり、特殊な記述ではない。たとえば、以下のような変更を指す。

逐次のループ：

```
for (i=0, i<N; i++) {  
    A[i] = B[i] * C[i];  
}
```

並列のループ：

```
ib = N / nprocs  
istart = myid * ib  
iend = (myid+1) * ib  
for (i=istart, i<iend; i++) {  
    A[i] = B[i] * C[i];  
}
```

ここで、nprocs は並列計算機全体の PE 数が入っている変数（各 PE で同じ値をもつ）、myid は自分の PE につけられた個別の認識番号(0 ~ nprocs-1)である。

すなわち並列ループの外側に注目すると、各 PE で担当する範囲は共通の ib であるが、ループ変数 (i) の範囲については、各 PE で別の範囲をもつ。しかしながら、 $A[i] = B[i] * C[i]$  という命令は、逐次ループと全く同じである。つまり、図 1 で示される命令は同一、図 1 のデータ 1 ~ 4 に相当する配列  $A[i]$ 、 $B[i]$ 、 $C[i]$  は別となる SIMD のモデルに従うことになる。

段階 1 から 5 を通じて、逐次プログラムから並列プログラムを作成できる「方法論」を習得するというのが、本講義の最終的な目的である。

### (3) 内容

本講義で行った授業内容を表 1 に示す。表 1 のように、本講義で用いたアプリケーションは、行列ベクトル積、べき乗法（行列ベクトル積が使われている固有値・固有ベクトルの計算法）、行列行列積、LU 分解法の 4 種である。

表 1 2007 年度夏学期の授業内容

日付	内容・コメント
2007年4月17日	ガイダンス資料
2007年4月24日	<b>スーパーコンピュータを利用しよう</b> ：スパコンを利用しよう、並列プログラミングの基礎、二分木総和演算
2007年5月1日	<b>並列数値処理の基本演算</b> ：性能評価指標、基礎的な MPI 関数、データ分散方式、ベクトルどうしの演算、ベクトル - 行列積、リダクション演算、数値計算ライブラリについて

2007年5月8日	<b>非同期通信</b> : 前回の演習の続き、1対1通信に関するMPI用語、サンプルプログラム(非同期通信)の実行
2007年5月15日	<b>行列-ベクトル積</b> : サンプルプログラム(行列-ベクトル積)の実行、並列化の注意点
2007年5月29日	<b>べき乗法</b> : べき乗法とは、サンプルプログラム(べき乗法)の実行、並列化の注意点
2007年6月5日	<b>行列-行列積(1)</b> : 行列-行列積とは、ループ交換法、ブロック化(タイリング)法、Cannonのアルゴリズム、Foxのアルゴリズム、SUMMA、PUMMA、Strassenのアルゴリズム、サンプルプログラム(行列-行列積(1): 簡単版)の実行、並列化の注意点
2007年6月12日	<b>行列-行列積(2)</b> : サンプルプログラム(行列-行列積(2): ちょっと難しい完全並列版)の実行、並列化の注意点、並列化のヒント
2007年6月19日	<b>高性能プログラミングの基礎(1)</b> : 階層キャッシュメモリ、演算パイプライン、ループアンローリング、配列連続アクセス、キャッシュとキャッシュライン、キャッシュライン衝突、サンプルプログラムの実行、演習課題、レポート課題
2007年6月26日	<b>高性能プログラミングの基礎(2)</b> : ブロック化、その他の高速化技術、OpenMP超入門、サンプルプログラム(OpenMP)の実行、演習課題、レポート課題
2007年7月3日	<b>LU分解法(1)</b> : LU分解法(ガウス・ジョルダン法、ガウス消去法、枢軸選択、LU分解法(外積形式、内積形式、クラウト法、ブロック形式ガウス法、縦ブロックガウス法、前進・後退代入))、サンプルプログラム(LU分解法)の実行、並列化のヒント、演習課題、レポート課題
2007年7月10日	<b>LU分解法(2)</b> : レポート提出の注意、レポート課題採点基準、コンテスト課題発表、コンテストプログラムの実行、LU分解の並列化のヒント(2)
2007年7月17日	<b>発展的課題</b> : ソフトウェア自動チューニング: 背景、ソフトウェア自動チューニングとは、FIBER方式、自動チューニング記述言語ABClibScript、ソフトウェアデモ、レポート課題

表1では、基本的な並列処理の用語や法則(アムダールの法則)などを教える座学も行った。特に、アルゴリズムの重要性を教えるため、通信アルゴリズムを二分木形態でおこなうと、素朴な実装の $O(P)$ のアルゴリズムに対して $O(\log(P))$ となる方式(ここで $P$ はPE台数で、 $p=1024$ 程度の例が現実として存在する)ことを説明し、通信アルゴリズムの重要性と効率のよいアルゴリズムの素晴らしさを理論と実装の観点から教えた。

また、数値計算ライブラリを利用する分野も多いため、数値計算ライブラリ概観、および代表的な数値計算ライブラリに使われているアルゴリズムも簡単に説明した。

演習としては、PCでは達成不可能な性能を受講生に体験させるため、スーパーコンピュータ上で高効率で実行できる行列-行列積をアプリケーションとして選んだ。これに加え、受講生自ら手でチューニングし、段階的に高速化する楽しみを体験させる演習（ループアンローリング演習）もとり入れている。

演習課題を与える一方で、受講生が参加できるプログラムコンテストも講義の一環として開催した。コンテストの参加者、すなわちコンテストにおける出題をすべて解答する並列プログラムを提出した場合、レポートに加点を与えた。また、コンテストにおいて入賞（1位～3位）した場合、無条件で「優」を与えるという条件を付した。なお、2007年度夏学期のコンテスト課題は、特殊なデータ分散をおこなった「行列-行列積」である。

本講義の演習のために、表2に示すサンプルプログラム9本を新規に開発した。このサンプルプログラムは、受講生が講義中にダウンロードして実行確認をした上で、演習で用いるものである。このようなサンプルプログラムを提供した理由は、スクラッチから並列プログラムを開発するには講義時間が少なすぎるので、サンプルプログラムの逐次プログラムを基にして並列化演習をさせるためである。

表2 サンプルプログラム一覧

サンプルプログラム (並列化の段階)	サンプルプログラムの内容
#1. Sp2007ex1.tar (段階1、2)	並列版 Hello プログラム、並列円周率計算プログラム、逐次転送方式による並列総和演算プログラム、二分木通信方式による並列総和演算プログラム、時間計測方法に関する並列プログラム
#2. Sp2007ex2.tar (段階1、2)	非同期通信の並列プログラム
#3. Sp2007ex3.tar (段階2、3)	行列 - ベクトル積の逐次プログラム
#4. Sp2007ex4.tar (段階3、4)	べき乗法の逐次プログラム
#5. Sp2007ex5.tar (段階3)	行列-行列積の逐次プログラム (お手軽並列化用)
#6. Sp2007ex5-2.tar (段階3、4)	行列-行列積の逐次プログラム (完全分散並列化用)
#7. Sp2007ex6.tar (段階2)	行列-行列積の逐次プログラム (逐次チューニング用)
#8. Sp2007ex7.tar (段階2、3)	行列-行列積の逐次プログラム (OpenMP 並列化用)
#9. Sp2007ex8.tar (段階4、5)	LU 分解法による連立一次方程式の求解の逐次プログラム

#### (4) 講義の経過

- 4月17日のガイダンスで、スーパーコンピュータのアカウント発行の都合から、事前登録が必要なことを通知。
  - 事前登録者数は、82名（所属内訳：工学部（機械情報、航空宇宙、電子情報、計数、マテリアル、化学システム、システム創成）、理学部（情報科学）、工学系研究科（社会基盤、建築、機械、環境海洋、航空宇宙、システム量子、地球システム、マテリアル、化学システム、原子力国際）、情報理工学系研究科（コンピュータ科学、数理情報、電子情報、知能機械情報、創造情報）、新領域創成科学研究科（基盤情報、人間環境）、理学系研究科（化学）、農学生命科学研究科（生産・環境生物））。
- 4月24日、スーパーコンピュータのアカウント配布。スーパーコンピュータを用いた演習の開始。
- 5月8日まで、約60名程度の学生が受講し演習を受ける。
- 5月15日以降、定常的に受講する学生が約30名程度に減少。
- 講義終了。レポート提出者：19名
  - 成績内訳：「優」：16名、「良」3名、「不可」：0名
- コンテスト課題提出者：4名
  - うち、入賞者（1位～3位）3名、完走者1名

### 3. 講義を通じて感じた問題点と今後の改善策

#### (1) 問題点

- **UNIX コマンドの未習熟者**
  - 本講義の受講条件として、UNIXの基本コマンドを習得していることを条件とした。しかしながら、UNIXのファイル構造、cp, rmなどの基本コマンドを理解していない受講生が多数いた。また、エキスパート向けのviではなく、初心者向けのemacsを本実習での推奨エディタとしたが、emacsできえも最低限扱えるスキルがない受講生も相当数いた。
  - センターのスーパーコンピュータ環境ではコマンドラインでコンパイルすることが前提であるが、コマンドラインでのコンパイル環境に慣れていない受講生が多かった。コンパイラが出すエラーの箇所（英語）の理解や、emacsを用いてプログラム上のエラー行への移動操作ができない受講生が多かった。このことが、スーパーコンピュータの敷居の高さを感じさせる要因となった。
- **C言語の未習熟者**
  - C言語を本講義での指定計算機言語としたが、C言語の基本事項を理解していない受講生が多数いた。本講義のガイダンス時に、習得計算機言語に関するアンケートを取ったところ、C、C++、Fortran習得者が85%を占め、Fortranのみの習得者が11%であったため、受講生のほとんどがC言語習得者であると判断した。しかしながら、並列化をする以前の逐次プログラムの読解能力が低いため、並列化に至らない受講生が多数いた。
  - 具体的に、C言語のスキルとして問題となった点を以下に列挙する。

- ◇ ループ構造、IF 文などの制御構造がわかっていない。
- ◇ グローバル変数、ローカル変数の概念がない。
- ◇ ポインタ引き渡しなど、基本的な関数引き渡しの概念がない。
- ◇ ローカル変数はスタック領域からデータを確保するなどのメモリ確保に関する知識がない。その結果、大きな配列が確保できないため、実行時エラーとなる。

#### ● スーパーコンピュータのパスワード設定ポリシーが難解

- センター側の計算機管理ポリシーの問題であるが、パスワード設定時に英語の大文字を始めと終わりを除いた場所に混在させる必要や、数字を混在させる必要などの条件があり、普通の計算機パスワードより厳しい。したがって、パスワード変更の際の警告によりパスワードの変更がすぐにできない、パスワード変更後の失念、5 回パスワードを間違えることによるアカウントロックが生じる受講生が多数出た。この点も、スーパーコンピュータの敷居の高さを感じさせる要因となった。

#### ● SIMD の概念の教授の難しさ

- 受講生が大幅に減少したのは、SIMD の概念を必要とする行列ベクトル積の並列化の講義の後である。並列化に至る前の段階である UNIX や C 言語の習熟度の低さから履修をあきらめた可能性もあるが、それに加えて、並列化特有の概念である SIMD の考え方が身に付かないことによる履修の断念が大きな要因として考えられる。

### (2) 今後の改善策

#### ● UNIX と C 言語のスキルの低さ

5 月中旬より TA 2 名を導入して個別に対応できるようにした。しかしながら、UNIX コマンドなどの基本スキルを教授するためには、最初の演習からのサポートが必須であった。また、工学部において最低限のスキルを身につける講義を受講しないと本講義を受講できなくなるなどの、本講義の受講条件の制度化・明確化が望まれる。

#### ● SIMD の概念の教授法について

本講義の演習の構成方針で、並列化に至るまでの段階を 5 段階に設定し、段階的にレベルを高める工夫をしたが、まだ不十分であったと思われる。特に、逐次プログラムの解読時間が少ないことも問題であった。逐次プログラム解析の時間を多めに与える、手計算でアルゴリズムを確認させるなどの配慮をする必要がある。

また夏学期では、並列処理としては高度な内容である「非同期通信」を早めに行い、かつ並列化には無関係だが演習課題としては面白い手動の「チューニング演習」を、「行列ベクトル積」の並列化演習の後で行っている。「行列ベクトル積」より先に手動の「チューニングの演習」、本講義の最後に「非同期通信」の演習を行うように配慮をすべきであった。

### 4. おわりに

82 名の事前登録に対し 19 名の単位を与えたので、23%という低い単位取得率となった。2007 年度冬学期では、この低い単位取得率の改善を試みるが、以下の 3 点がポイントとなる：

- (1) TA の早期からの導入；(2) 講義内容の再構成；(3) 丁寧に SIMD の概念を教授する；

なお、本講義において「優」を与えた 16 名の学生においては、提出されたレポートの内容、分量、完成度ともに卓越である者が相当数いた。したがって当初掲げた、「**実際に高性能プログラムを基盤センターのスーパーコンピュータ上で開発できるという、実用的でかつ、研究者として生き残るために必要な技能の習得を目指す**」という目的は十分に達成できたものと自負している。

なお 2007 年度冬学期には、本講義と同様の内容を駒場教養学部の学部 1、2 年生に行う全学ゼミ「スパコンプログラミング研究ゼミ」も開講されたことを付け加えておく。この全学ゼミは、センターのスーパーコンピュータの啓蒙のみならず、将来の HPC 分野における天才的なプログラマーの発掘と養成を指向している。冬学期の「スパコンプログラミング 1 および I」、および全学ゼミ「スパコンプログラミング研究ゼミ」の報告は、来年度中に行う予定である。

本講義の資料については、東大校内においてのみ以下のページにて PDF 形式で公開している。

<http://www.kata-lab.itc.u-tokyo.ac.jp/class-matr2007s.htm>

## 参 考 文 献

- [1] 東京大学情報基盤センタースーパーコンピューティング部門 教育利用（試行）サービス、  
[http://www.cc.u-tokyo.ac.jp/use\\_info/education.html](http://www.cc.u-tokyo.ac.jp/use_info/education.html)
- [2] P. パチェコ 著 / 秋葉 博 訳、MPI 並列プログラミング、培風館、2001
- [3] 中島研吾、Hitachi SR11000 を利用した並列プログラミング教育：東京大学 21 世紀 COE プログラム多圏地球システムの進化と変動の予測可能性、スーパーコンピューティングニュース、Vol. 8、No. 3、pp. 21-30、東京大学情報基盤センター、2006. 5