

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（５）
一 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」（2012年度
夏学期）、および、全学ゼミ「スパコンプログラミング研究ゼミ」（2011年度冬学期）
を通じて

片桐 孝洋

東京大学情報基盤センター 准教授

1. はじめに

計算科学・工学、ハードウェアの急速な発達を背景に、「第3の科学」としての大規模並列シミュレーションへの期待は、産学において一層高まっている。最新ハードウェアを駆使し、大規模並列シミュレーションプログラムを開発するための体系的な教育プログラムの事例は、世界的に見ても少ない。東京大学では2008年初頭から、「T2K オープンスパコン、次世代スーパーコンピュータ等を駆使した大規模シミュレーションを実施し、新しい科学を開拓する人材」を育成するための全学的な教育プログラムとして「学際計算科学・工学 人材育成プログラム」を実施している[1][2]。

一方、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（以降、スパコン）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高性能計算（HPC）教育の支援を行ってきた。センターのスパコン啓発と、ユーザの研究分野（主として大規模数値シミュレーション分野）における高性能並列プログラム開発の長期的支援を行なうことが目的である。

以上の背景から、本講義は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）」を通年科目（夏学期、冬学期）として開講している。2012年夏学期で11回連続の開講となる。本講義は、工学部や工学系研究科以外の学生も受講している。

一方、天才的な学際領域の研究者を早期から養成し、高性能計算分野における優秀な人材を発掘する目的で、東京大学教養学部の学生に対し工学部と同様の講義を行う「スパコンプログラミング研究ゼミ」を開講している。2007年度冬学期に「全学ゼミ」として開講して以来、2011年度夏学期で5回目の開講となった。

双方の受講生に対し、2008年6月稼働のT2K オープンスーパーコンピュータ（東大版）（HITACHI HA8000 クラスタシステム、以降T2Kとよぶ）のアカウントを無料で発行し演習を行ってきた。また、2012年夏学期からは、2012年4月に試験運転を開始したFujitsu PRIMEHPC FX10（FX10 スーパーコンピュータシステム、以降、FX10）の教育利用を、日本において先駆けて開始している。これらは、スーパーコンピューティング系の「教育利用」サービスの一環である[3]。

T2Kは教育利用方針が強化され、4ノード（64コア）まで利用できるコア数（並列数）となった。これは、本講義の開始同時の2007年度に利用可能であったHITACHI SR11000/J2の1ノード（16コア）に対し、大幅に増加した。また、FX10では、12ノード（192コア）の利用が可能となり、教育利用で使える並列数が大幅に増加した。著者の知る限り、最新アーキテクチャのCPUを192並列で定常的に利用できる教育環境はほとんどない。初等の並列処理教育環境としては日本有数の環境だといえる。

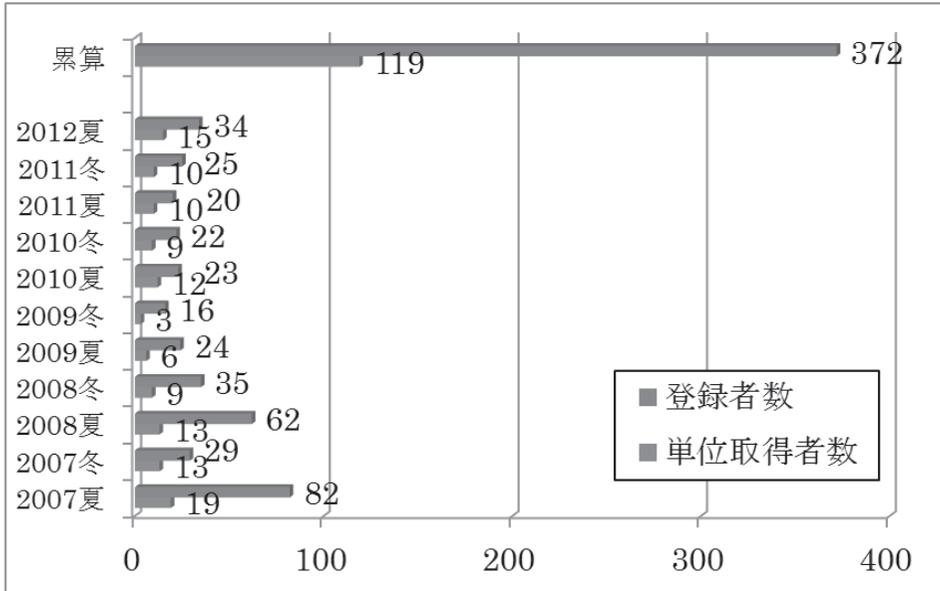
本報告は2009年度（2008年4月～2009年3月）の報告[4]に引き続き、2012年度夏学期の

教育利用報告である。

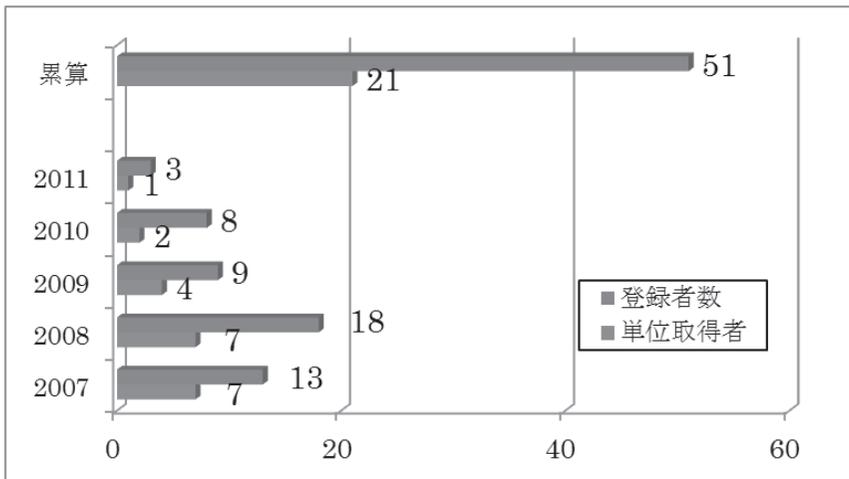
2. 受講者に対する統計データ

(1) 受講者数

本講義は2007年度から夏学期、冬学期と年2回、連続して開講している。講義に事前登録した上でスパコンアカウントを発行した数（登録者数）と、単位を取得した人数（単位取得者数）を図1にのせる。



(a) スパコンプログラミング (1) および (I)



(b) 全学ゼミ：スパコンプログラミング研究ゼミ

図1 講義登録者数および単位取得者数の累計

図1から、工学部の講義においては登録者数の累計は372名、単位取得者数は119名に達し

ている。

(2) 受講者の所属の累計

スパコンプログラミング(1)および(I)において、いままでアカウントを発行した学生の所属は、以下のとおりである(順不同)。

- 工学部
 - 産業機械、機械情報、機械、精密、電気電子、電子情報、計数、システム創成、社会基盤、航空宇宙、マテリアル
- 理学部
 - 天文学、地球惑星物理、化学、物理
- 農学部
 - 生命化学・工学、法学部、第3類
- 工学系研究科
 - システム創成、社会基盤、原子力国際、精密機械、物理工学、航空宇宙、機械情報、電気系、先端学際、応用化学、技術経営、金融、化学システム
- 情報理工学系研究科
 - コンピュータ科学、電子情報学、知能機械情報、数理、創造情報学
- 新領域創成科学研究科
 - 情報生命、先端エネルギー
- 理学系研究科
 - 地球惑星科学、天文学、農学生命科学、応用生命、数理科学研究科
- 学際情報学府
 - 学際情報学
- 経済学研究科
 - 現代経済学、統計理論、金融システム
- 医学研究科
 - 国保

以上から、本学の多種の学科から受講がなされている。本講義は、学際領域研究の支援になっているものと期待される。

3. 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、以下の指針を示している。

「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要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。」

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のような方針ではなく、センターユーザの多数を占める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、センターのスパコンを利用するための最低限の技術の習得を目的にする。

(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ている。SIMD とは、並列計算機の種類の一つである。図 2 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。

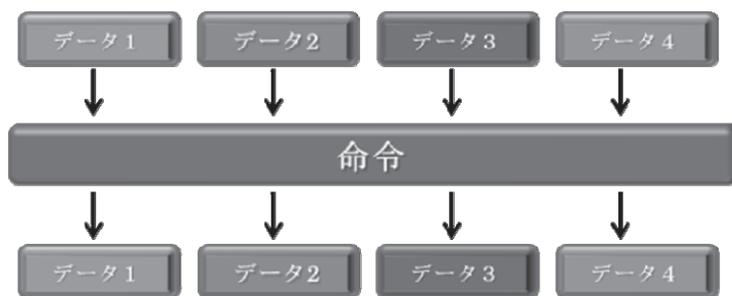


図 2 SIMD の概念

図2のSIMDの概念を、並列プログラミングの経験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** 並列環境に慣れるため、簡単な並列プログラムを実行する。並列プログラムを眺める。
- **段階 2** 逐次プログラムを実行する。逐次プログラムを解析する。
- **段階 3** 逐次プログラムに存在するループに変更を施すだけで並列化ができる演習。
- **段階 4** 逐次プログラムに存在するループに変更を施し、かつ単純な MPI 関数を記述することで並列化できる演習。
- **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。

並列化のための段階 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。この方法論は、一般的であり、

特に行列計算処理の並列化においては効果的である。この方法論を習得することで、ソフトウェア工学的により並列化の作法を身に付けさせる。

(3) 内容

本講義で行った授業内容を表1に示す。この内容は初めて開講された2007年度夏学期以来、大幅な変更は無い。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・固有ベクトルの初等的な数値計算法)、行列-行列積、LU分解法の4種である。

表1 授業内容

講義回数	内容概略
ガイダンス	初回ガイダンス、高性能計算の基礎
第1回講義	並列数値処理の基本演算 :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
第2回講義	スーパーコンピュータを利用しよう :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
第3回講義	高性能プログラミングの基礎(1) :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
第4回講義	高性能プログラミングの基礎(2) :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
第5回講義	行列-ベクトル積 :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
第6回講義	べき乗法 :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
第7回講義	行列-行列積(1) :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):単年版)の実行、並列化の注意点
第8回講義	行列-行列積(2) :コンテスト課題発表、コンテストプログラムの実行、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
第9回講義	LU分解法(1) :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
第10回講義	LU分解法(2) :LU分解の逐次アルゴリズムの解説
第11回講義	LU分解法(3) :レポート提出の注意、レポート課題採点基準、LU分解の並列化のヒント(2)
第12回講義	非同期通信 :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

第13回講義	発展的話題: 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
--------	---

演習課題を与える一方、受講生が参加できる「プログラミングコンテスト」を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。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。

2012年度夏学期のコンテスト課題は、複数の右辺ベクトルをもつ「連立一次方程式の解法」である。2012年の夏学期の期間では、FX10のFortran90言語とC言語による同一処理の実行において、コンパイラ最適化能力の違いが確認された。そのため、Fortran部門とC部門の2種を設定し、それぞれに対して入賞者を決定した。

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本を教材として開発している。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、演習で用いる。C言語版とFortran言語版の2種を用意している。このサンプルプログラムも、T2Kで提供したものと大差は無い。FX10の実行環境でも提供されて、実習に用いることができた。

表2 サンプルプログラム一覧

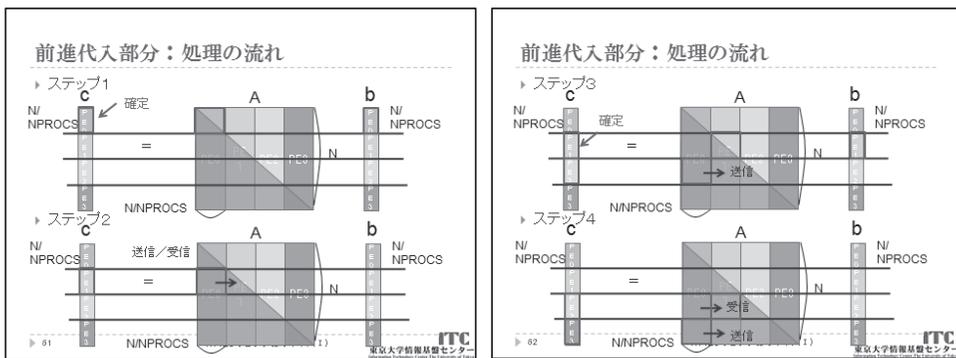
サンプルプログラム (並列化の段階)	サンプルプログラムの内容
#1. Samples-fx.tar (段階1、2)	並列版Helloプログラム、並列円周率計算プログラム、逐次転送方式による並列総和演算プログラム、二分木通信方式による並列総和演算プログラム、時間計測方法の並列プログラム
#2. Mat-Mat-noopt-fx.tar (段階2)	行列-行列積の逐次プログラム (逐次チューニング用)
#3. Mat-Mat-openmp-fx.tar (段階2、3)	行列-行列積の逐次プログラム (OpenMP 並列化用)
#4. Mat-vec-fx.tar (段階2、3)	行列-ベクトル積の逐次プログラム
#5. PowM-fx.tar (段階3、4)	べき乗法の逐次プログラム
#6. Mat-Mat-fx.tar (段階3)	行列-行列積の逐次プログラム (お手軽並列化用)
#7. Mat-Mat-d-fx.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)
#8. LU-fx.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
#9. SP_Isend-fx.tar (段階1、2)	非同期通信の並列プログラム

(4) スーパーコンピュータを用いた環境で特質すべき教育・演習事項

一連の本報告で説明してきたので、重複する事項を割愛する。

そのうえで指摘する一つの特徴は、最終課題は密行列の LU 分解法の並列化を対象にしていることである。密行列の LU 分解法は数値計算の応用に移るために必須の処理であり、かつ、MPI を用いた並列化のレベルは必ずしも初心者向きのレベルではない。著者の感想では、並列化に関しては中級者以上の能力が必要である。そのため、多くの受講生が LU 分解の並列化で断念している。しかし受講生の中には、高度な並列化を実装してくる者も少なくない。

LU 分解法の並列化に関して、比較的単純な LU 分解部分の並列化に加えて、前進代入・後退消去の部分についてウェーブ・フロント法を用いて並列化する方法を紹介している。図 3 に、このウェーブ・フロント法の処理の流れを示す。



(a) ステップ 1 および 2

(b) ステップ 3 および 4

図 3 LU 分解法における前進代入時のウェーブ・フロント法による処理の流れの例

図 3 のウェーブ・フロント法の並列化は非同期的に行われるため、並列処理の実行形式と、そのための実装方法がわかりにくい。そのため敷居が高いが、MPI 初心者でも筋の良い受講生は並列化できる。より高度な並列化方式を学ぶための入口的な問題ととらえると、よい問題であると思っている。

(5) コンテスト結果

以下の表 2 にコンテストの結果を示す。

表 2 コンテスト結果

■C 言語部門

順位	参加者	問題1 求解時間[秒]	配点	問題2 求解時間[秒]	配点	総合配点
1	参加者 A	0.71	5	0.618	10	15
1	参加者 B	0.69	10	1.859	5	15
3	参加者 C	24.722	2	58.122	1	3
3	参加者 D	25.341	1	18.15	2	3

■Fortran 言語部門

順位	参加者	問題1 求解時間[秒]	配点	問題2 求解時間[秒]	配点	総合配点
1	参加者 A	0.697	10	0.515	10	20
2	参加者 E	4.747	5	24.46	5	10

表2のように、全体で5名の参加者があったが、全員が入賞となった。

表2では、参加者A、Bの実行速度がきわめて高速である。これは、数値計算ライブラリを利用したためである。本講義では数値計算ライブラリの利用方法は教えないため、利用のためには相当苦勞して独自に勉強したようである。いずれにせよ、MPI で動作する数値計算ライブラリが利用できるレベルであれば、本講義の単位取得に必要な技量が備わっていると判断できる。

(6) 講義の経過と感想

今までの記事において講義の経緯を紹介しているので割愛する。

例年、コンテスト課題を提出できるのは3名程度であり、今回の5名参加は多い。コンテスト課題は、少なくとも中級以上のスキルを要求する。したがって、数名は本講義終了時において中級以上のレベルに達する。また上位入賞するプログラムは、実装的・アルゴリズム的に相当工夫がなされている。逆にいうならば、定常的に数名程度は天才的な技量をもつ学生が受講していることになる。このような<センスのよい>学生を応援し、HPCが必要とされる研究分野で育成していくことが、特に重要である。

以下、レポート提出者による本講義を受講した感想を列挙する。なお、改行以外の文章の修正は行っていない。

スパコンプログラミング (1) および (I) 受講者

- 大学にいられる内にスパコンに触れてみたかったということが、この授業を選択した理由だったが、考えていた以上に得られるものがあつた。MPI はほぼ通信関連の実装だけなので、並列化の肝心要の部分は自分で考えて実装しなければならず、それにはメモリの構造などコンピュータの仕組み自体の知識が必要となる。そういった知識は、スパコン上でなくとも今後プログラムを書く上で非常に役に立つだろう。また、講義内容はほとんどUNIXについて知らなくてもついていけるようになっており、とても分かりやすかつた。終盤でもきちんと理解できたのは、ひとえに図を多用したスライドや、易しい内容から順を追つた授業のためである。
- スパコンプログラミングという名前から非常に難しそうだという先入観を持っていたが、講義の内容はとても丁寧で、並列化の初心者である自分にも理解しやすくプログラミングを楽しむことができた。また、行列計算の高速化のアルゴリズムはスパコンでの計算以外にも役立つことができそうで、非常に勉強になった。専攻では流数值体力学を扱ったり、また、研究では位置推定問題を扱ったりすることがあり、非常に大きなサイズの行列計算を行うことがある。このようなプログラミングの際に、講義で学んだループアンローリングを施すだけでも計算速度が早くなり、効率良くシミュレーションを行うことができた。シミュレーション実験などを行う場合には知っておくと非常に有益な知識であり、プログ

ラミング経験の浅い学部生にも勧めたいと思う講義であった。

- 研究でシミュレーションを行っているが、情報系の授業を受けたことがなかったので、受講する前はついていけるか不安だった。しかし、かなり基本的なところから ppt で丁寧に指導して下さったので、講義の概要は理解できたかと思う。講義の ppt が web に上がっているというのも、復習をしやすくてよかった。実習も、新しいスパコンを使えてとてもいい経験が出来たと思うが、少々使いづらいのが気になった。授業時間外では満足には使えないというのと、あとデバックがすごいしづらいつと感じた。自分で mpi 環境が用意できれば、自らの環境で主要なデバックを行い、そのコードの性能をスパコンで試す、ということが出来たと思うが、そうでない自分にとっては少々取り掛かりづらかった。行列(行列積の話はとても面白く、分かりやすかった。数値計算の高速化を考えている研究者がどのようなことを考えているのか、少しだけ知ることが出来たと思う。自らの研究にも生かしていきたいと思う。
- この講義によって MPI の基本知識や CPU の内部構成などの知識を得ることができました。また、世界で 18 位のスーパーコンピュータのリソースを一部でも使うことができたというのも非常に面白く感じました。実際に、為になる演算などは行うことはできませんでしたが、今後この知識を高めていき、高い並列性を持つプログラムを書けるようになり実際の研究等に活かせたらなと思います。非常に面白い講義をありがとうございました。
- 去年は登録だけで途中で時間がなくなり、今年はリベンジでした。去年一度見たこともあり、スムーズに進みました。講義に対しての意見は特にありません。これ以上ないと思います。
- 卒論時、修士課程以降と、スーパーコンピュータや PC クラスタを使用する研究室 2 つに所属したが、実際に MPI のコードを見る・書く・動かすことが皆無であったため、「プログラミング入門」の講義として非常に有意義であった。研究室会などの議論の場で多に役立てられることが予想される。また、C 言語と FORTRAN の両方に対応している点に好感を持った。多くの工学系の講義は FORTRAN の使用を前提としていて、C 言語 er として苦労した経験がある。
- 並列数値計算の実装手法について、効率的な並列化のポイントや MPI 関数の利用法といった実践的な内容を学ぶことができた。また行列演算の数値計算手法をアルゴリズムと実装テクニックの双方から学ぶことができた。専攻している知能機械情報学でもロボットの動作軌道生成における IK の計算等高次元の行列計算が必要とされる場合があるので、今後の自分の研究に学んだ並列化計算手法を生かしていきたい。
- 私は、所属する研究室の後期博士過程の先輩に受講しないかと誘いを受け、この講義を受講しました。私の所属する研究室は、基本的には計算力学の研究を行っていますが、研究員の方々の中には領域分割法を用いて有限要素法の並列化を行っている人も多くいます。私は流体構造連成解析手法の開発が研究テーマであり、まだ基礎的なモデルの計算しか行わないので並列化に真面目に取り組んだことはなかったのですが、研究室会での研究報告などで MPI での並列の話題をよく耳にしていました。ハードは若干の知識があったので所々は理解できていたのですが、基本的な知識のない私にとってはある意味魔法のようで、非常におくぶかい世界だと感じると同時に、とても難解に感じ、ある種の恐怖心のようなものを抱いていました。この講義を受講してみて、スーパーコンピュータの世界はやはり

奥深く難しいものであるという実感を得たと同時に、苦手意識がなくなり興味深い分野だ
と思うようになりました。自分自身でこの分野を研究することが今後あるかはわかりませ
んが、私の研究の社会的な意義が津波や水害の防災である以上、社会に役立てるためには
解析の大規模化が不可欠であり、それらの分野と関わっていくことが重要であると感じて
います。その意味で、並列コンピューティングの知識と実装力を得られたのはとても有意
義であり、やりがいある授業だったと思います。幸いこれらの分野に近いところにいま
すので、今後も自分自身で勉強を深めていきたいと思っています。片桐先生、1 学期にわたる御
教授どうもありがとうございました。

- 私は研究で主に Fortran を用いており、C 言語には不慣れであったので、サンプルプロ
グラムやアルゴリズムの説明の際、常に C 言語と Fortran の両方が用意されていたことは勉
強になりました。自然と C 言語と Fortran の配列アクセスの違いを意識するようになり
ました。また、キャッシュやレジスタなど、ハードウェアを意識したプログラムの書き方
について学ぶのは初めてだったので、コンパイラの最適化が行っていることを自分の手で
実際に行って計算速度の向上を実感できたことは、非常に良い経験となりました。レポ
ート問題は、各回それぞれ徐々に難易度が上がるように設定されており、自分のレベルにあ
わせて得点を得ることができるようになっていて親切だと感じました。ただ、MPI 関数に
ついての第二回の講義は、私は事前に MPI についてある程度の知識を持っていたので何と
かついていけましたが、初見の学生が一度に全て理解するのは大変なのではないかと感じ
ました。自分が予備知識を持たないで受講していたとしたら、おそらくこの回で挫折した
のではないかと感じました。過去の履修状況からとてもハードな授業だとい
う印象を持っていたので、戦々恐々とした思いで履修登録いたしました。蓋を開けて
みれば親切で分かりやすい講義でしたので、最後まで諦めず受講することができました。
こうして締め切り前にレポートを提出するに至ることができ、安心しています。自身の研
究において、この講義で得た知識は役に立つものであると確信しています。4 か月間有難
うございました。
- C 言語初修という無謀な状態でスパコンプログラミングに突撃したがために、大事な 1 対
1 通信の話のときにそもそも C の文法がよく分かっていないというとんでもない事態に
陥ってしまいました。行列-行列積のあたりでは、序盤で理解し損ねた 1 対 1 通信をなんと
か理解しようと努め、LU 分解のあたりでは、そもそも LU 分解とはなんぞや状態で、まず
LU 分解の理解を頑張るという有様で、散々な有様でした。C 言語の基礎や数値解析の勉強
をしっかりとってからスパコンプログラミングに臨むべきだったかなと反省しております。
こんな初心者が紛れ込んでいるがために、「本当に分からない人」向けの「ほぼ解答が載
っています」は大変ありがたかったです。案の定、自分はトンチンカンで無茶苦茶なこと
をしており、「ほぼ解答」がなかったら、一生迷宮を彷徨っていたような気がします。ソ
ースを見ながら、ここがこうなって、ああなって、というようになんとか理解をするこ
とことができました。まだまだ理解の及ばないところも多々あるので、これで並列プログラミ
ングの勉強をやめてしまうのではなく、これからも引き続き勉強をしていきたいと思っ
ております。授業のスタイルは、現状のスタイルで問題ないかと思います。パワポ資料もネッ
ト上で閲覧できるので、家での勉強に大変役立ちました。色々な参考書やネット上の資料
を参照しながら勉強をしましたが、結局のところ、授業資料が一番詳しくて分かりやすか

ったです。授業で一番困った点は、やはりスパコンが使えなくなる場面が多々あったことでしょうか。タイミングが悪いのか、[ENABLE, STOP] 状態の「lecture」に出会うことがほとんどで、挙句の果てに、レポート締切日も月曜のはずなのに[ENABLE, STOP] 状態でジョブが投入できないという事態に陥り、やや物足りないレポートになってしまいました。以上のように、主に自分自身の阿呆さ加減のせいで、なかなか理解が進まなかった並列プログラミングですが、少なくとも 1 対 1 通信関数については理解できたと思っております。LU 分解はまだ怪しいですが……。ともかく、並列プログラミング導入の足掛かりとして、この授業は大変役に立ちました。ありがとうございました。

- スパコンを使うことが出来る貴重な機会が得られるということで履修しましたが、初めはプログラム以前にターミナルの扱いやエディタの使い方に苦戦しました。今ではそれなりに扱えるようになったので、そういった意味でもよい経験が出来ました。LU 分解やべき乗法については数値解析という講義で以前学習したことがありましたが、並列プログラミングで実装するとなると難易度が非常に上がり、特に LU 分解についてはとても苦労しました。また、アンローリングや OpenMP のように比較的簡単に実装できる技術も身につけることが出来ました。今後、スパコン以外では数十もの PE を用いて並列プログラミングすることはあまりないかもしれませんが、いい体験が出来てよかったです。
- 内容が多くて、授業の速度が速かったんですが、FX 10 のスーパーコンピュータに触れられたり、スパコンでのプログラミングの技術を習ったりして嬉しいと思います。将来に役に立つと思います。今回の授業は内容を全部納得できたとは言えないですが、授業のスライドを貴重な資料として保存し、後から必要な時参考になると思います。
- 高性能計算の基礎から網羅的に説明してもらえたので、幅広い知識が得られた。演習内容も充実しており、実際にプログラムを書きながら、学習する事ができた。
- スパコンといえば、Fortran や特殊な言語を使わなければならないものと思っていたが、今回初めて実際にスパコンに触ってみて、C 言語と MPI の組み合わせで十分高速化をできることを知り、敷居が下がった。知能機械の分野でもスパコンを使った学習や探索をしているところはあるので、今度自分の研究分野でも使えそうなところでは利用していこうと思う。実際の演習では、行列演算のためにループで計算をする際の開始終了条件を決めるのに苦労した。また、プログラムが途中でハングして終了してしまうと、printf の出力も消えてしまい、どこで計算が止まっているか分からず苦労した。gdb のようなデバッガが使えると便利だと思った。

本講義の最終目標は、並列処理技術の習得により、個人の研究を各段に進めることにある。本講義終了後に実際に研究に適用できそうだという感想や、高速化の楽しさがわかったという感想があった。このことは、並列処理学の教育者冥利に尽きるものであり、大変うれしい反響である。

4. おわりに

本講義の事前登録者は、2012 年度まで累算し 372 名にもものぼる。最終的な単位取得者数は、累算で 119 名である。本講義を継続していくことで、本センターのスーパーコンピュータが利用できる人材にとどまらず、高性能計算分野に貢献できる多数の人材を輩出できる。学界にと

どまらず産業界にも人材が提供される。高性能計算分野の経済発展に貢献し、センターの社会貢献に資する礎となる教育を行っていく所存である。

本講義の授業資料は、平成 21 年度分から一般に公開している。以下のページを参照いただければ幸いである。

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

参 考 文 献

- [1] 中島研吾、究極の「並列プログラミング教育」を目指して—地球惑星科学専攻での 4 年間と未来への提言—、スーパーコンピューティングニュース、Vol. 11, No. 4, pp. 30-44, 2009 年 7 月.
- [2] 東京大学 学際計算科学・工学人材育成プログラム.
<http://nkl.cc.u-tokyo.ac.jp/CSEedu/>
- [3] 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系 教育利用.
<http://www.cc.u-tokyo.ac.jp/service/education/>
- [4] 片桐孝洋：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（4）— 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」（2009 年度 夏・冬学期）、および、全学ゼミ「スパコンプログラミング研究ゼミ」（夏学期）を通じて、スーパーコンピューティングニュース、Vol. 12, No. 3, 2010 年 5 月.