

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（２）
— 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」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（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いる。センターのスパコン啓蒙と、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。

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」を通年科目として開講している。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。一方、天才的なスパコンユーザを早期から養成し、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２００７年度冬学期に「全学ゼミ」として開講した。双方の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「教育利用（試行）」サービスの一環である[1]。

本報告は、２００７年度夏学期（２００７年４月～２００７年７月）の報告[2]に引き続き、２００７年度冬学期の報告である。

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（再掲）。

「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。」

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。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。

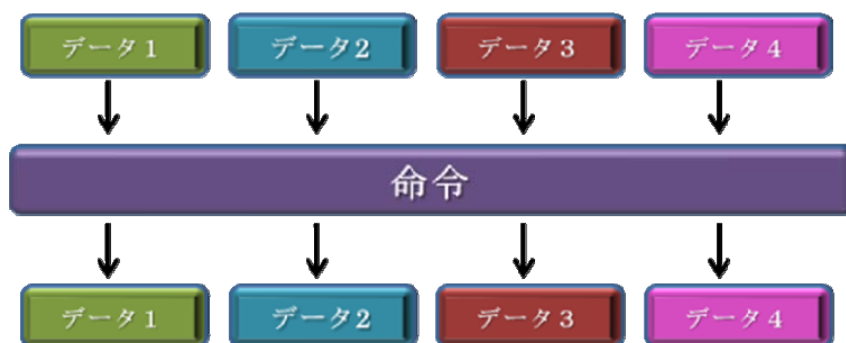


図 1 SIMD の概念

この 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。

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。この講義内容は、平成 19 年度夏学期と変わっていない。

表 1 2007 年度冬学期の授業内容

日付	内容・コメント
2007 年 10 月 1 日	初回ガイダンス
第 1 回講義 2007 年 10 月 15 日	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 2007 年 10 月 22 日	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 2007 年 10 月 29 日	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 2007 年 11 月 5 日	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 2007 年 11 月 19 日	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 2007 年 11 月 26 日	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 2007 年 12 月 10 日	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):簡単版)の実行、並列化の注意点
第 8 回講義 2007 年 12 月 17 日	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 2008 年 1 月 7 日	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 2008 年 1 月 21 日	LU 分解法(2) :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 11 回講義 2008 年 1 月 28 日	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 12 回講義 2008 年 2 月 4 日	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

表 1 の内容は、2007 年度夏学期に行った講義の内容と同じであるため、説明を割愛する。ただし変更点として、高度な内容にもかかわらず夏学期に第 3 回の講義で行った「非同期通信」を、冬学期には後半の第 11 回講義に移動した。また、並列処理に関連しないがチューニング技法として面白い実習である、「ループアンローリング」実習などの単体性能を改善する演習を、初めての演習として第 3 回講義と第 4 回講義に行った。

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。なお、2007年度冬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（図2参照）。このプログラムコンテストは、教養学部の全学ゼミの学生もハンデなしに参加を募った。

問題説明

- ▶ 「行列 - 行列積」
- ▶ 各PEは、初期状態として、 $NN \times NN$ のサイズの行列を持っているとする。
- ▶ 全PE数は、8PEとする。
- ▶ PEの構成は、 2×4 のグリッド構成とする
- ▶ したがって、 $C=A * B$ の各行列サイズは、以下のとおり。
 - ▶ A: $2NN \times 4NN$ / B: $4NN \times 2NN$ / C: $2NN \times 2NN$

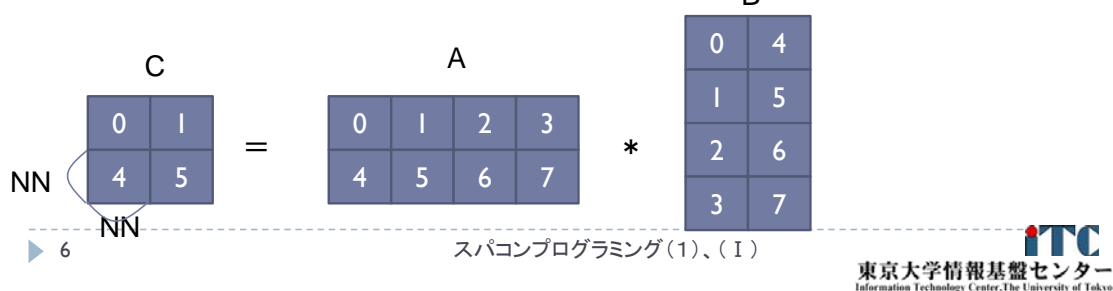


図2 プログラミングコンテスト課題

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で演習で用いる。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. sp2007samples.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. sp2007Mat-Mat-noopt.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. sp2007Mat-Mat-openmp.tar (段階2、3)	行列-行列積の逐次プログラム (OpenMP 並列化用)

#4. sp2007Mat-vec. tar (段階 2、3)	行列-ベクトル積の逐次プログラム
#5. sp2007PowM. tar (段階 3、4)	べき乗法の逐次プログラム
#6. sp2007Mat-Mat. 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. sp2007Mat-Mat-d. 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. sp2007LU. 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. sp2007SP_Isend. tar (段階 1、2)	非同期通信の並列プログラム

(4) 講義の経過

- 10月1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29名。
 - ◇ 所属内訳：工学部（機械情報、航空宇宙、計数）、理学部（天文科学、物理）、工学系研究科（社会基盤、機械、環境システム、航空宇宙、システム量子、システム創生、地球システム、マテリアル、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）、理学系研究科（化学、地球惑星科学）
 - 全学ゼミ登録者数：13名（演習室の都合から、上限17名）
- 10月22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。
- 11月19日以降、定常的に受講する学生が約15名程度に減少。
- 講義終了。
 - 工学部共通科目 レポート提出者：13名
 - ◇ 成績内訳：「優」：13名
 - 全学ゼミ レポート提出者：7名
 - ◇ 合格：7名
 - なお全学ゼミのほうは、課題提出のほかに、自由課題でもレポートを受け付けた。その結果、自由課題として2課題を受理した。内容は、講義レポートとして十分質の高いものであった。
- コンテスト課題提出者：5名（うち、全学ゼミから1名）
 - 入賞者（1位～3位）3名、完走者1名。1位は同着2名で、全学ゼミからの参加者（学部1年生）が1位を受賞したことは特筆に値する。

◇ 問題サイズ NN = 500 (小規模問題)

順位	名前	実行時間[秒]	加点
1位	工共通 A	0.066	10
2位	全学 A	0.077	5
3位	工共通 B	0.082	2
4位	工共通 C	0.203	1
5位	工共通 D	0.282	0

◇ 問題サイズ NN = 6000 (大規模問題)

順位	名前	実行時間[秒]	加点
1位	全学 A	90.49	10
2位	工共通 A	111.4	5
3位	工共通 B	118.6	2
4位	工共通 C	356.1	1
5位	工共通 D	701.5	0

- 以上の2問題の合計点数で順位を決定。結果として、1位と2位が総合で1位となる以外は、順位の変動は無い。
- 全学ゼミからの参加者（全学 A）が、大きい問題サイズで2位と優位な差のプログラムを作成した点は興味深い。

3. 講義を通じて感じた問題点と今後の改善策

(1) 講義の感想

以下にレポート提出者における、本講義を受けた感想を原文のまま掲載する。なお、講義内容に関係ない記載については削除した。

- 並列プログラムの作成に習熟することができました。また、実際に並列プログラムを走らせる環境があったため、作成したプログラムの実際の速度などを知ることができて、並列計算により詳しくなることができました。特に、LU 分解の並列化が容易でないことなどが印象に残りました。卒業論文では、並列プログラムの作成支援についての研究を行いました。また、この際に、実際に並列プログラムの作成を行いました。講義で習ったMPI によるプログラミングが非常に役に立ちました。これから先に、並列プログラムを作成する機会は増えていくと思いますが、この機会ですぐに得た技術を生かしていきたいと思っています。
- 並列化を意識したプログラムを書いたのは初めてで、興味深かったです。また、同時に普通プログラミングよりも考慮すべきことが多く、難しかったです。ループアンローリングやパイプラインなどのチューニングは、専攻で理論よりの勉強をしているためか、煩雑に感じてしまいました。実験を行うときに困ったのが、自分のスキル不足かもしれませんが、講義で与えられたプログラムで、問題のサイズやPE 数を色々変えて実験するためには、手動で何回もサブミットする必要があったことです。同様に、これを書き換えた自分のプログラムでも同じような手間がかかり、困りました。
- 一年半ほど前、前の大学にいた時、MPIやOpenMPの本を図書館にMPIの本を買っても

らい、研究室のクラスタにインストールし習得しようと試みたが、許可がおりず、もろもろの事情から、あきらめざるを得ない状況になってしまったが、今回、その技術を習得する機会に恵まれ、また、アーキテクチャ上こうなっているからこうすれば高速化可能であるといった根拠とともに、高速化技法について学習することができて非常にありがたかった。ただ、おしむらくは、授業端末がMacであるため、ものすごくキーボードが扱いにくく、SR11000のターミナルでもle colorやcannaすらないという状況に非常に戸惑いを受けた。MPIそのものに関して言えば、特にMPI_Comm_split関数を使えた時、その便利さに感動した。今期の授業では、MPIのMPIらしい使い方を学習したが、今後、もっと習熟度を上げていきたいと思う。

- 私は現在修士2年で、自分の研究でも並列計算を含めて計算機を利用したシミュレーションを多用しました。しかしながらそれはツールとしての利用であり、研究の範囲では中身にまでは深く踏み込んで扱うつもりはありませんでした。それでも、本講義で効率的なプログラミング技法、及び並列化アルゴリズムを実践的に学ぶことができ、それまでブラックボックスとして利用していたコードが、実際に何を目的としているのかを理解し、改良やより効率的な運用を行なうことができました。M1の時に取っておけばもっと良い研究ができたと後悔しております。
- 始めは、教養学部の計算機プログラミング1修了者程度でも分かるレベルから講義が始まったので、大変分かりやすかった。徐々に講義が進むにつれ、概念が難しくなっていくが、段階的だったので、興味を減じることなく聞くことができた。アルゴリズムの改善はパズルゲームの用で実におもしろい。今回、講義の内容を一部しか習得することができなかつたので、スパコンのアカウントがなくなるのはとても惜しい。ただ、並列化に限らず、アルゴリズムと計算量についての基礎知識が着いたことはとても楽しかった。TAの方々は説明が丁寧で分かりやすく、親身で丁寧だった。こちらが分からないことに対しても否定的な意見を述べられることがなかつたので、気持ちよく学習することができた。
- キャッシュミスを防ぐという概念は今まで全く持っていなかつたものなので、大いに感じ入るところがあつた。特に行列積ではブロック化をするだけで大幅に改善されるとあつて、キャッシュという概念の重要性と同時に、面白さを学んだ。繰り返してforループを用いるのは当然のように考えていたが、まさかその書き方でここまで結果が違ふとは思わず、非常に驚いた。並列化など何も特別なことをせず、ただループをすこし手で書き下すことでこうも結果が変わるのであれば、普通のプログラミングにも採り入れていくことに何の躊躇も持たない。
- スパコンというものを実際に触れてみてスパコンに出来ることとそうでないことがより良く分かつた。スパコンに対する誤ったイメージが払拭された。個人的にはスパコン上で実行するための並列化処理よりも、個人の計算機でもできるlapackの説明やアンローリングなどの技法の方が興味がありおもしろかつた。
- 元々、プログラミング自体は好きでしたが、スーパーコンピューターを利用した並列計算という未知の世界に魅かれて受講させて頂きました。振り返ってみると、非常に有意義な機会だつたと思います。規模は小さいながらも並列プログラミングに取り組む機会が得られたのは大変刺激的でした。問題の難度、テキスト中のアドバ

イスの程度も、私には適度で、前提知識はかなり欠如していたかとは思われますが、それでも与えて頂いた情報をもとに頭を使って考えるうちに、小さいながらも成果が見えてきて、大変やりがいのある経験でした。特に最後のLU 分解には苦しみましたが、出来上がってみると楽しい経験でした。

- やはり、C言語はやっておかないとこの授業はついていけないと思います。よって、履修者はFortranではなくてC言語経験必須にすべきと思います。並列計算はスパコン利用者でないとニーズがないので、スパコンを利用していそうなところからの受講生を集めれば、それなりの教育ができると思います。結局、理学系は知りませんが、工学系でのC言語の授業を開講すべきと思います。
- プログラムの並列化というものが実際にはどうやって実装されているのかという点に興味があってこの授業を受講したのですが、想像以上にハードな内容で大変勉強になりました。並列化に関してはシンプルな発想と手段によって実装されていることが多く、それはまだ基礎的な部分しか触れられていなかっただけかもしれませんが、意外に感じると同時に面白かったです。
- 個人的にはもう少しIsend, Irecv 周りを色々やったり、MPI の内部実装に起因する高速化などをやって欲しいが、それはちょっと無理がある。中盤ではバグで悩み、取れなかった人が次の週からぼろぼろ来なくなっていった。授業では新しい概念をコードに落とすという事をするのだが、「新しい概念」以外のところに不安な所があり、身動きが取れなくなってしまうという人が多い。しかし開発でCを使った事があるレベルの人のみに受講者を絞っても、そんな人は勝手に調べてやっていけるのであまり嬉しくない。どうするのが良いのだろう。

(2) 今後の改善策

前節の感想から見出せる問題点を、以下に列挙する。

1. 並列処理の概念の難しさ：

夏学期から指摘されている問題である。教授法の改善点としては、明快な図解、反復学習、実例を挙げた解説しか解決策がないと思われる。本講義では、前回に比べSIMDの考え方について例題ごとに図解し、総合で3回程度繰り返すなどの改善は行った。

2. プログラム実行の煩雑さ：

センターのジョブ実行方式が、バッチジョブ形式になっていることの問題である。現在、インタラクティブなジョブ実行を許可していない。計算機運営方針との都合であり、センターの計算機を利用する上では、インタラクティブな実行を許可するという改善は困難であろう。しかしながら、講義時間中は同時実行ジョブ数を多くするなど、受講生の待ち時間を少なくする配慮を行っている。

3. テキストエディタ環境の悪さ：

スパコン上で提供されているテキストエディタの環境設定の問題である。本年度は、センター側にある emacs を利用して、プログラムの改編をおこなうように教授した。平成 20 年度からは、ECCS 端末側の emacs を利用して、センター側のファイルを操作する機能(tramp 機能による ssh) が利用できることを確認しており、本件については改善されるものと期待している。

4. 継続的なスパコンアカウントの維持：

講義の枠組みでは、継続的なアカウントの利用は困難である。受講生が所属する研究室において、廉価なスパコン利用のコース（たとえば、SR11000 の研究室コース）を利用するなどのセンターサービスの利用をもって、継続的なスパコンアカウントの利用がなされることを期待している。

5. C言語未習熟者：

夏学期から問題となっている。工学部の講義と連携して講義内容を修正する、もしくは、センターの講習会においてC言語などの初歩的なプログラミングの演習を企画する、などの改善策が考えられる。なお、本改善に関連する「教育プログラム」については、次節で記載する。

4. 人材育成プログラムとの連結

前節での問題5にあったように、高度な計算科学分野の人材を育成するために計算科学・工学分野の人材育成プログラムが検討されている。C言語などの計算機言語に始まり、通信や数値計算ライブラリ、さらには、数値計算アルゴリズムといったような鍵となる技術、スキル、アルゴリズムをセンターの計算機を利用して教育する枠組みが検討されている。（これは、「学際計算科学・工学 人材育成プログラム」と呼ばれている[3]）。図3に現在検討中の教育プログラムを載せる。

既存講義との位置づけ

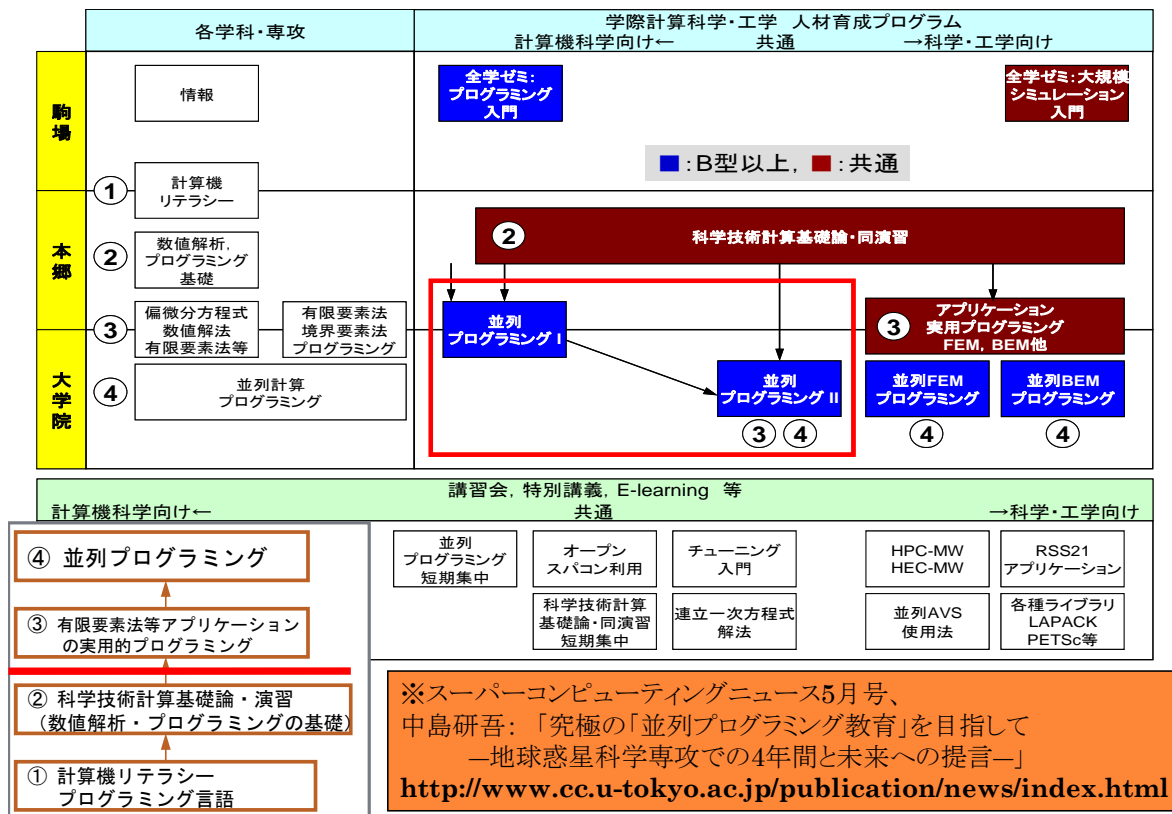


図3 検討中の教育プログラム

図 3の教育プログラムでは、計算機シミュレーション側の計算科学分野と、システムソフトウェア側の計算機科学分野のHPCに関する教育内容を総合的に教育する。高性能プログラミングに必要なソフトウェア（以降、「HPCミドルウェア」と呼ぶ）を使える人材（C型）、HPCミドルウェアを改編して利用できる人材（B型）、HPCソフトウェアを開発できる人材（A型）、そして新概念を含む新HPCミドルウェアを発明できる人材（S型）という、4種類の段階的な能力を持つ人材を育成する。さらに、センターにおける講習会も、本教育プログラムと連結させる。

この教育プログラムと本講義を連結することで、本講義における教育効果を一層増進させる。具体的に本講義は、計算機科学側から並列プログラミングを教授する「並列プログラミングⅠ」に相当するが、計算科学におけるアプリケーションも意識した内容となっている。また、計算科学側の並列プログラミングⅡへのシームレスな移行も考慮する。HPC分野の人材を多数育成し、計算科学分野におけるいっそうの研究活性化を行っていきたい。

5. おわりに

本講義の事前登録者は、新規開講された2007年度だけでも124名にもものぼる。最終的な単位取得者数は、1年を通じて39名になった。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ものと自負している。

本講義の授業資料は、東大校内においてのみ公開している。ただし、講義の概略などの情報は東大以外でも閲覧できる。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-matr2007w.htm>

<http://www.kata-lab.itc.u-tokyo.ac.jp/class-matr2007semi.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
- [2] 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—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—、スーパーコンピューティングニュース、Vol. 10, No. 2, pp. 12-19, 2008年3月
- [3] 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. 10, No. 3, pp. 30-44, 2008年5月