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（３）
― 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」(2008 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[1]

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。

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。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、2008 年度夏学期で 2 回目の開講となった。

双方の講義の受講生に対し、夏学期は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 モデル、1 ノード（8 コア））のアカウントを無料で発行して演習を行った。これは、スーパーコンピューティング部門の「教育利用」サービスの一環である[1]。2008 年 6 月には、新スパコンである 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）が稼働した。以降 T2K マシンと呼ぶ。T2K マシンで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）が増加した。著者の知る限り 64 並列の並列計算機が教育利用で行える状況はまれであり、初等の並列処理教育環境としては日本有数となる環境である。2008 年度冬学期から T2K マシンを利用して演習が行われている。

本報告は 2007 年度冬学期（2007 年 10 月～2008 年 3 月）の報告[2]に引き続き、2008 年度を通じての教育利用報告である。

２．学際科学・工学人材育成プログラム

（１）概要

高度な計算科学分野の人材を育成するため、計算科学・工学分野の人材育成プログラムが必要とされている。C 言語などの計算機言語に始まり、通信ライブラリ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リテラシー・技術・アルゴリズムをセンターの計算機を利用して教育するプログラムが試行されている。これは、東京大学「学際計算科学・工学人材育成プログラム」と呼ばれ[3]、平成 21 年度から既存科目を読み変えることで順次開講されている。図 1 に概要を載せる（再掲)。

[1] 本原稿は、前号（Vol.11,No.3, 2009 年 5 月）に掲載された原稿を加筆修正したものです。

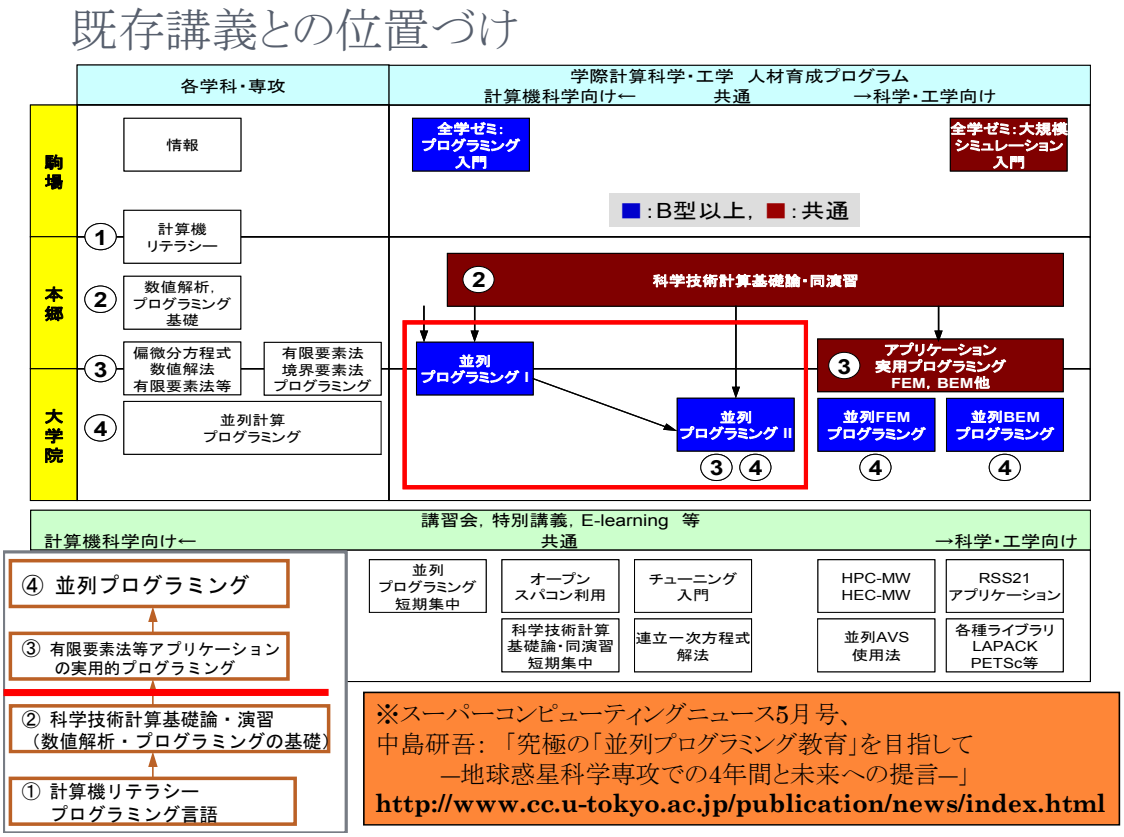


図 1　東京大学「学際計算科学・工学人材育成プログラム」（試行）

図 1 では、計算機シミュレーション側の計算科学分野と、システムソフトウエア側の計算機科学分野におけるハイ・パフォーマンス・コンピューティング（HPC）に関する教育を融合して、総合的に HPC 教育を行うことを目的とする。

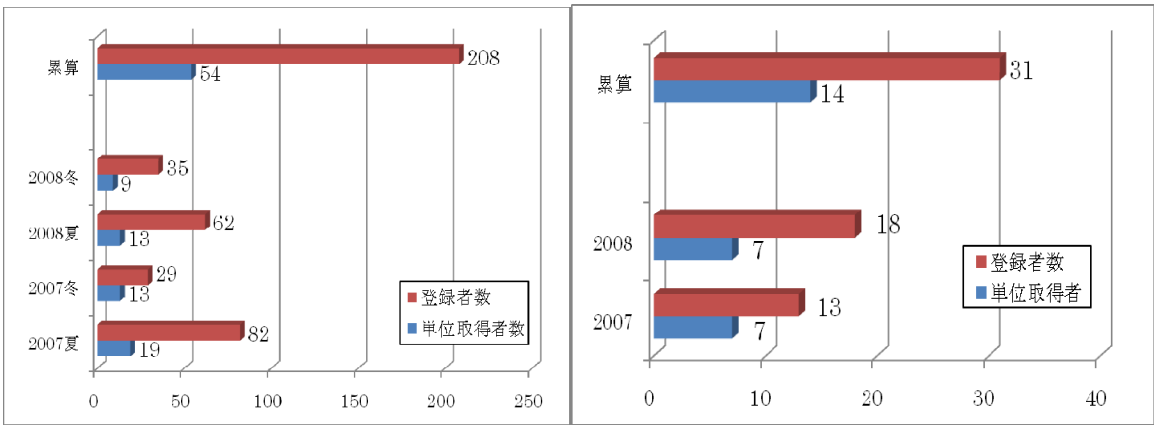
HPC プログラミングに必要なソフトウエア（以降、「HPC ミドルウエア」と呼ぶ）を使える人材（C 型）、HPC ミドルウエアを改編して利用できる人材（B 型）、HPC ソフトウエアを開発できる人材（A 型）、そして新概念・新アルゴリズム・新 HPC ミドルウエアを発明できる人材（S 型）の 4 種類の段階的な能力を持つ人材を育成する。さらに、センターにおける講習会も教育プログラムと連結させる。

教育内容に関して、①計算機リテラシー、②科学技術計算基礎、③実用的プログラミング、④並列プログラミングという段階がある。一般には、②から③への段階で大きなギャップがある。本プログラムでは、学部までに③までの学習を終え、大学院では③以上の内容を教える。学習内容の連続性を意識して、教育プログラムが組まれている。

「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」は、この教育プログラムの一環で開講されている。図 1 の「並列プログラミング I」と「全学ゼミ：プログラミング入門」に相当し、A 型以上の人材育成を目的に、④の内容を教授する。内容は計算機科学向けという立場を取りながらも、数値計算処理を中心とした計算科学アプリケーションを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していく。

（２） 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 2 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 2　講義登録者数および単位取得者数

（３） 受講者の所属

スパコン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
  - 機械情報工学科、機械工学科、電子情報工学科、航空宇宙工学科、計数工学科、マテリアル工学科、化学システム工学科、システム創成学科
- 理学部
  - 天文学科、地球惑星物理学科、化学科、数学科、物理学科
- 工学系研究科
  - システム創成学専攻、社会基盤学専攻、原子力国際専攻、精密機械工学専攻、物理工学専攻、航空宇宙工学専攻、機械工学専攻、電気系工学専攻、建築学専攻、海洋工学専攻、地球システム工学専攻、システム量子工学専攻、マテリアル工学専攻、化学システム工学専攻、人間環境学専攻
- 情報理工学系研究科
  - コンピュータ科学専攻、数理情報学専攻、電子情報学専攻、知能機械情報学専攻
- 新領域創成科学研究科
  - 基盤情報学専攻
- 理学系研究科
  - 地球惑星科学専攻、天文学専攻
- 数理科学研究科
  - 数理科学専攻
- 農学生命学研究科
  - 生産・環境生物学専攻

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エア、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、並列計算機の分類の一つである。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。

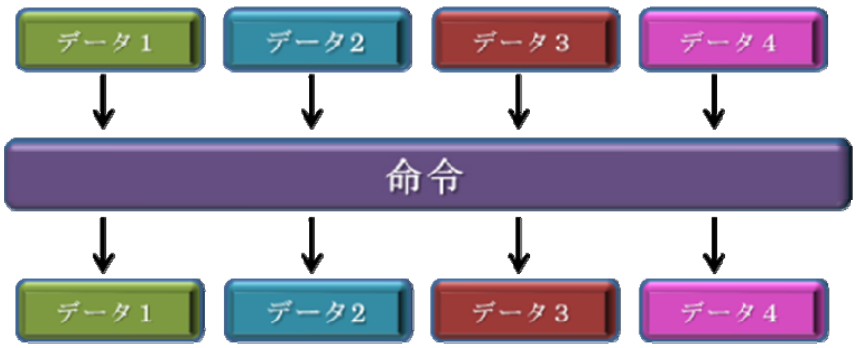


図 3　SIMD の概念

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。

この、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、ソフトウエア工学的によい並列化の作法を身に付けさせる。具体例として図 4 に、行列 - ベクトル積（y＝Ax）処理における並列化の方針（C 言語）をのせる。

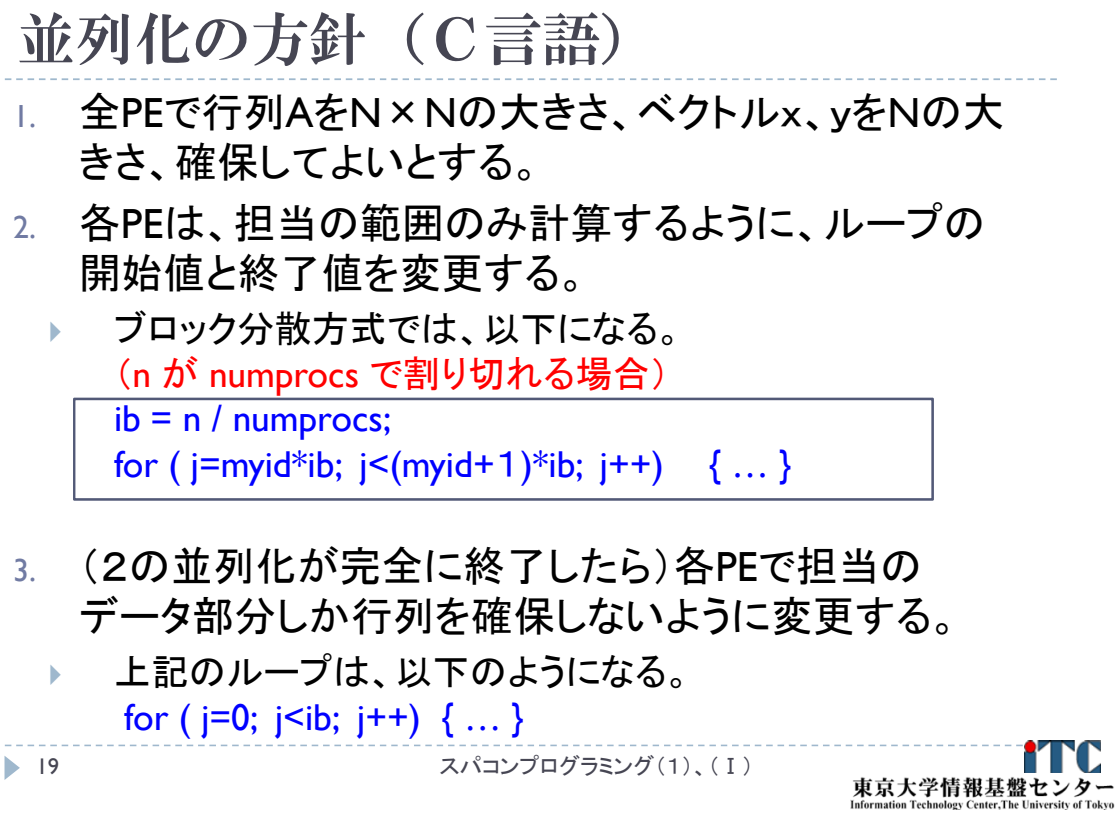


図 4　行列 - ベクトル積（y=Ax）の並列化の方針（C 言語）

（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）：簡単版）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**：ソフトウエア自動チューニング：背景、ソフトウエ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、ソフトウエアデモ、レポート課題

本講義の特筆すべき教授項目は、キャッシュ最適化技術を並列化の前に教えることである。計算機アーキテクチャのトレンドとして、今後もキャッシュを考慮したプログラミングができないと高性能化が実現できない。したがってその概念を、座学として教えるだけにとどまらず、サンプルプログラムを用いて速度を実感させる。さらに実感させるだけではなく、最適化の課題を出すことで、実際にコードチューニングを行わせる。キャッシュ最適化の効果を実体験させることで、実学として身につけさせる工夫がなされている。

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、2008 年度冬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。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。

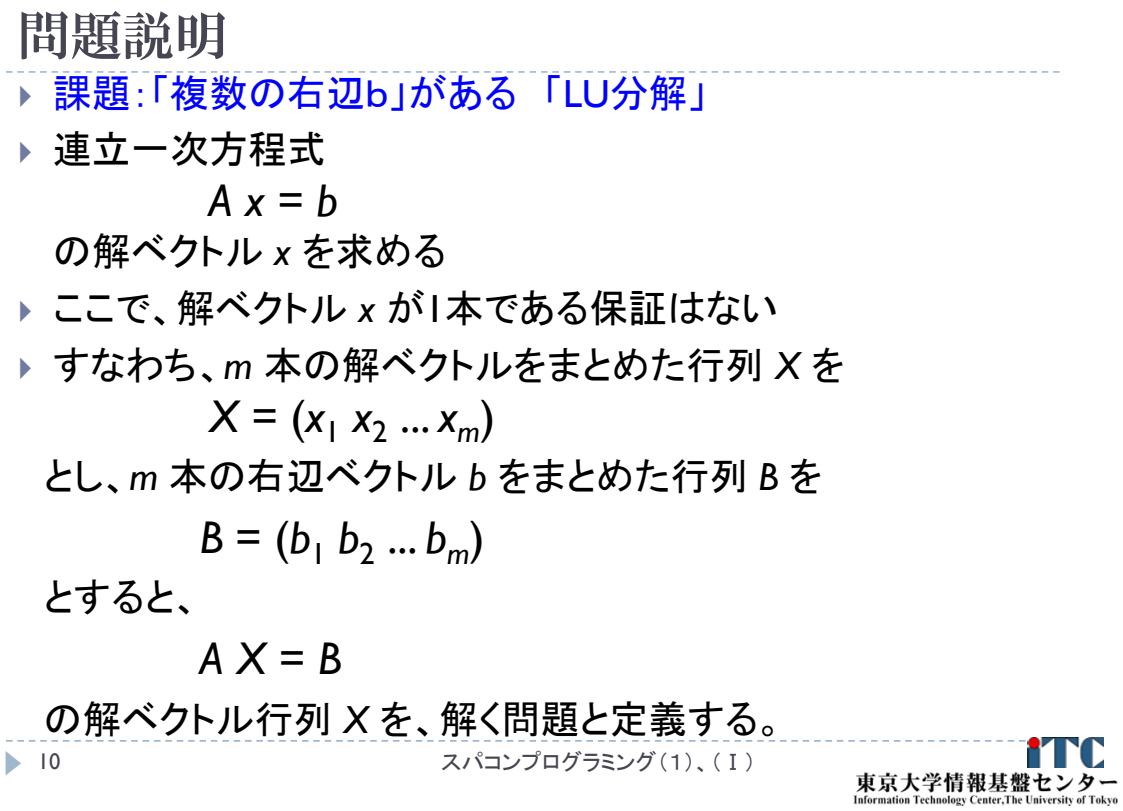


図 5　プログラミングコンテスト課題

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、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。

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. sp200Xsamples.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. sp200XMat-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. sp200XMat-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. sp200XMat-vec.tar （段階２、３）	行列-ベクトル積の逐次プログラム
#5. sp200XPowM.tar （段階３、４）	べき乗法の逐次プログラム
#6. sp200XMat-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. sp200XMat-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. sp200XLU.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. sp200XSP_Isend.tar （段階１、２）	非同期通信の並列プログラム

（４）並列環境で特質すべき教育・演習事項

T2K マシンの稼働により、並列実行数が 8 コアから 64 コアに大幅増加された。このような環境下では、並列化ができない逐次部分の増大が全体の性能劣化を引き起こすことが、少なからず体験できる。特に通信処理の実装は、アルゴリズムに依存して性能差が出やすい。

そこで本講義では、MPI の各プロセッサに分散されているデータを加算して、その後結果をプロセッサに収集する処理を例題にとり、効率の良い実装アルゴリズムの重要性を紹介している。具体的には、バケツリレーのように逐一隣接のプロセッサに転送する方式（逐次転送方式）では通信が *O(process)*かかるが、二分木上に分割して並列に送る方式（二分木通信方式）では *O(log(process))*となる。*process=1024* の実行では、1023 回と 10 回の大きな差になることを教えている。さらに例示するだけではなく、サンプルプログラムを配布して実性能を測定する課題を出し、実機での性能差を体験させている。

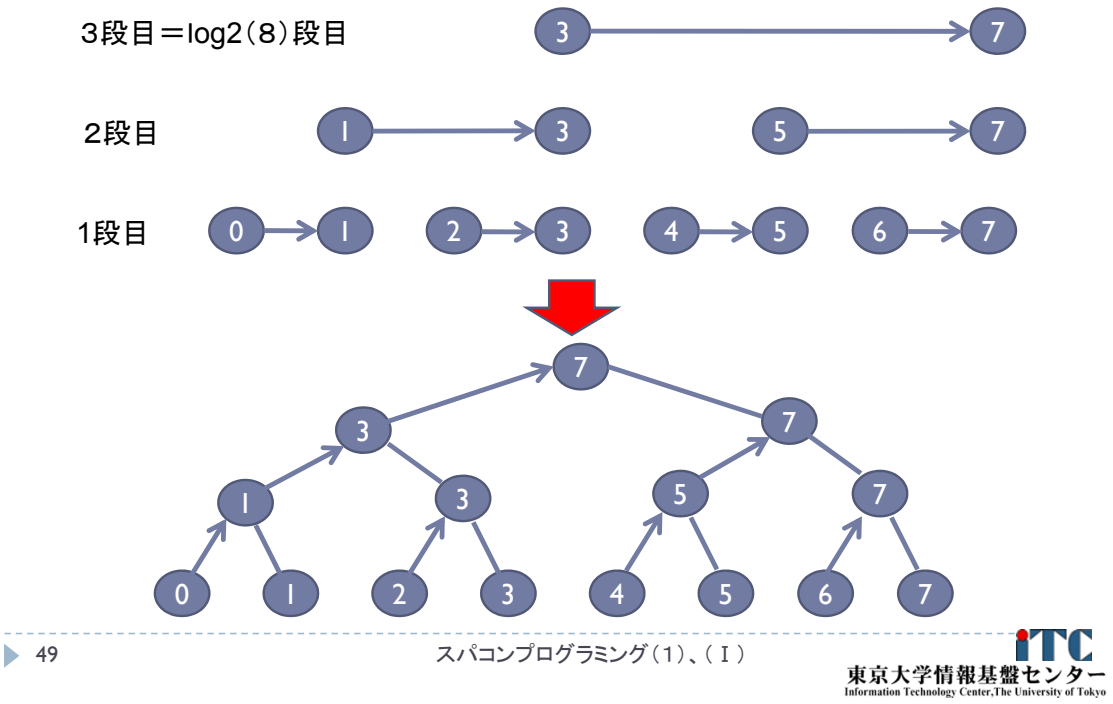


図 6　二分木通信方式（８プロセスの例）

（５）講義の経過

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。

今回は、コンテスト課題を一新したので、その結果を報告する。

- 夏学期：コンテスト課題提出者：3 名（全学ゼミ 2 名）(HITACHI SR11000/J2 を利用)
  - N=7200、M=1

順位	学籍番号	タイム	加点	参考加点
1.	共通 1	25.14	10	10
2.	共通 2	39.80	5	5
3.	共通 3	43.61	2	1
		参考記録	参考順位	参考加点
	駒場 1	48.08	5	0
	駒場 2	43.42	3	2

  - N=800、M=32000

順位	学籍番号	タイム	加点	参考加点
1.	共通 3	3.70	10	10
2.	共通 1	4.83	5	5
3.	共通 2	38.73	2	0

	参考記録	参考順位	参考加点
駒場 1	5.18	4	1
駒場 2	5.08	3	2

✧ 総合順位

順位	学籍番号	点数
1.	共通 1	15
2.	共通 3	12
3.	共通 2	7

参考順位

順位	学籍番号	点数
1.	共通 1	15
2.	共通 3	11
3.	共通 2	5
4.	駒場 1	4
5.	駒場 2	1

➢ 冬学期：コンテスト課題提出者：3 名
（T2K マシン（HITACHI HA8000 クラスタシステム）を利用）

✧ N=4224、M=1

順位	学籍番号	タイム	加点
1.	共通 1	1.95	10
2.	共通 2	9.62	5
3.	共通 3	10.36	2

✧ N=640、M=30720

順位	学籍番号	タイム	加点
1.	共通 1	0.12	10
2.	共通 2	1.38	5
3.	共通 3	2.59	2

✧ 総合順位

順位	学籍番号	点数
1.	共通 1	20
2.	共通 2	10
3.	共通 3	4

- 夏学期、冬学期ともに、大きな時間差がある。これは、問題レベル（右辺方程式の数、LU 分解レベル）の並列性を利用して並列化したかどうかである。
- 冬学期の一位は、それ以外より 1 桁速い。この理由は、最適化された数値計算ライブラリ BLAS をインストールして利用したことによる。このようなライブラリ利用が容易

にできるようになったのは、T2K マシンにおいてシステムの OS とコンパイラがオープンなものになったことが要因の 1 つである。

例年、コンテスト課題を提出できるのは 3 名程度である。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、少なくとも受講者の 1 割程度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、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重要となる。

５．おわりに

本講義の事前登録者は、2008 年度まで累算し 239 名にものぼる。最終的な単位取得者数は、累算で 68 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きたい。

一方、登録者数より実際の単位取得者数が少ないという問題がある。これは、追跡調査をしていないが、ひとつの要因は、受講者のプログラミング能力が低く、丁寧に指導しても並列化までたどり着けない点があげられる。この場合、抜本的には、逐次プログラムから勉強をしなおしてもらい、ある程度のスキルを付けたうえで本講義に来てもらうように制度を再考する必要があると思われる。

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] 東京大学情報基盤センタースーパコンピューティング部門　教育利用
http://www.cc.u-tokyo.ac.jp/use_info/education/

[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（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―、スーパーコンピューティングニュース、Vol.10, No.5, pp.66-76,2008 年 9 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.