

東京大学工学部・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)」

埜 敏博

東京大学情報基盤センター

1. はじめに

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(Sセメスター、Aセメスター共通)として開講している。前任者から引き継いで2016年度から実施しているものであり、2021年度Aセメスターで連続30回の開講を数える。後期教養教育科目にも登録されており[1]、後で述べるように、工学部や工学系研究科以外の学生も数多く受講している。また、2017年度より計算科学アライアンスの認定講義にもなっている[2]。

2021年度は、本講義の受講生に対して、2021年5月に運用を開始したWisteria/BDEC-01を用いた演習を行った。Sセメスターでは、運用開始直後の5月18日に、いち早くアカウントを配布し、富士通A64FX CPUを搭載したWisteria-Odysseyを用いて演習を開始した。また、講義の終盤には、Intel Xeon IceLake, NVIDIA A100 GPUを搭載したWisteria-Aquariusを用いた演習も行った。受講生にとって、スーパーコンピュータ富岳と同じアーキテクチャを持つA64FX CPUなど、最新、最先端のスパコン環境を使用できることで、より高い動機付けになると考えている。

Wisteria-Odysseyでは12ノード(48コア×12ノード=576コア)、Wisteria-Aquariusでは4GPUまで1ジョブで利用可能であり、教育利用であっても十分に高い並列数を扱うことができる。

2021年度については引き続き、コロナ禍により、Sセメスターの開始時点から、全てZoomを用いたオンライン講義とした。同様の形式になって2年目であり、概ね順調に進めることができた。これまでも講義資料はWebに掲載していたが、加えて、Zoomの録画をLMSに掲載することで、学生からは復習ができて良いと好評であった。演習の際に、Zoom共有画面での操作のデモと、自分の端末とを見比べて作業がしやすいとのコメントがあった。以上を踏まえて、今後も引き続きオンラインでの講義を想定している。

2. 講義内容

本講義で行った講義内容を表1に示す。内容については、C, Fortranに加えて、Pythonを用いた並列処理や機械学習フレームワークの使用方法についても紹介した。Sセメスターにおいては、Wisteria/BDEC-01が稼働を開始する5月15日以降にスパコンを利用開始するため、第2回より先行して第3、4回を実施し、その間は手元のPCを用いて演習を行ってもらった。アプリケーションは、行列-ベクトル積、ベキ乗法(行列-ベクトル積を応用した固有値・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種類である。

加えて、講義の一環として、受講生が参加できる「プログラミングコンテスト」を開催した。すべての出題を解答できるプログラムを提出したコンテストの参加者には、成績に加点した。さらに、コンテストにおける入賞者(1位~3位)においては、無条件で「優」以上を与えた。

演習には、表 2 に示す 11 本のサンプルプログラムを教材として用いている。受講者はこれらをダウンロードした上で、動作確認した上で演習を実施する。表 2 最後の Python によるものを除いて、それぞれ、C 言語版と Fortran 版を用意している。

表 1 講義内容

講義回数	講義内容
ガイダンス	初回ガイダンス、高性能計算の基礎
第 1 回	並列数値処理の基本演算
第 2 回 (S: 第 4 回)	スパコンを利用しよう：スパコン(Wisteria-Odyssey)を用いた実習
第 3 回 (S: 第 2 回)	高性能プログラミング技法の基礎(1)：階層メモリ、パイプライン処理、ループアンローリング、キャッシュヒット率
第 4 回 (S: 第 3 回)	高性能プログラミング技法の基礎(2)：キャッシュブロック化、OpenMP 超入門
第 5 回	行列-ベクトル積
第 6 回	べき乗法
第 7 回	行列-行列積(1)：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、簡単な並列化
第 8 回	行列-行列積(2)：完全な並列化
第 9 回	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))、コンテスト課題発表
第 10 回	LU 分解法(2)：LU 分解のアルゴリズム詳細
第 11 回	GPU プログラミング(1)：(OpenACC による Wisteria-Aquarius の利用、性能比較)
第 12 回	Python & 機械学習

表 2 サンプルプログラム一覧

	サンプルプログラム内容
Samples-wo (第 4 回(S), 第 2 回(A))	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
Mat-Mat-noopt-pc (第 2 回(S)) Mat-Mat-noopt-wo (第 3 回(A))	行列-行列積の逐次プログラム(逐次チューニング用)
Mat-Mat-openmp-pc (第 3 回(S))	行列-行列積の逐次プログラム(OpenMP 並列化用)

Mat-Mat-openmp-wo (第4回(A))	
Mat-vec-wo (第5回)	行列-ベクトル積の逐次プログラム
PowM-wo (第6回)	ベキ乗法の逐次プログラム
Mat-Mat-wo (第7回)	行列-行列積の逐次プログラム (お手軽並列用)
Mat-Mat-d-wo (第8回)	行列-行列積の逐次プログラム (完全分散並列用)
LU-wo (9, 10回)	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
Isend-ofp (第10回)	非同期通信の並列プログラム
Mat-mat-acc (第11回)	行列-行列積の逐次プログラム (OpenACC 並列化用)
Python (第12回)	Pythonによる並列プログラム、機械学習サンプル

3. 受講者についての統計データ

(1) 受講者数

2021年度のSセメスターの履修登録者は43名(学部:17名、大学院:26名)、Aセメスターの履修登録者は39名(学部:21名、大学院:18名)であった。そのうち、単位取得者は、37名(学部:10名、大学院:27名)であった。

(2) 受講者の所属

今年度の履修登録者について、以下の通りである。

- 医学部: 1名
 - 医学部医学科 1
- 工学部: 31名
 - 工学部社会基盤学科 1
 - 工学部建築学科: 1
 - 工学部機械工学科: 3
 - 工学部航空宇宙工学科: 1
 - 工学部精密工学科: 2
 - 工学部電子情報工学科: 8
 - 工学部電気電子工学科: 1
 - 工学部計数工学科: 1
 - 工学部マテリアル工学科: 4

- 工学部応用化学科：1
- 工学部化学システム工学科：3
- 工学部化学生命工学科：1
- 工学部システム創成学科：4
- 理学部：3名
 - 理学部情報科学科：1
 - 理学部物理学科：1
 - 理学部生物情報科学科：1
- 経済学部：1名
 - 経済学部金融学科：1
- 教養学部：1名
 - 教養学部学際科学科：1
- 教育学部：1名
 - 教育学部総合教育科学科：1
- 総合文化研究科：1名
 - 総合文化研究科広域科学専攻：1
- 理学系研究科：5名
 - 理学系研究科物理学専攻：2
 - 理学系研究科地球惑星科学専攻：3
- 工学系研究科：27名
 - 工学系研究科社会基盤学専攻：1
 - 工学系研究科建築学専攻：1
 - 工学系研究科機械工学専攻：3
 - 工学系研究科航空宇宙工学専攻：4
 - 工学系研究科システム創成学専攻：2
 - 工学系研究科電気系工学専攻：6
 - 工学系研究科物理工学専攻：4
 - 工学系研究科マテリアル工学専攻：3
 - 工学系研究科化学システム工学専攻：1
 - 工学系研究科原子力国際専攻：2
- 医学系研究科：2名
 - 医学系研究科社会医学専攻：1
 - 医学系研究科内科学専攻：1
- 新領域創成科学研究科：5名
 - 新領域創成科学研究科先端エネルギー工学専攻：2
 - 新領域創成科学研究科複雑理工学専攻：1
 - 新領域創成科学研究科人間環境学専攻：2
- 情報理工学系研究科：4名
 - 情報理工学系研究科コンピュータ科学専攻：1
 - 情報理工学系研究科電子情報学専攻：3

以上から、工学系・情報系を中心に、学内の多様な学部・学科から受講生が集まっていることがわかる。

本講義を通じて、本センターのスーパーコンピュータシステムの利用者を育成するだけでなく、スーパーコンピュータの活用を通じて学際領域分野に貢献できる人材を輩出していると考えられる。また、スパコン利用の経験が多くの学生に広がることにより、学界のみならず、産業界においてスパコンを利活用できる人材育成につなげていけるものと考えている。

Wisteria/BDEC-01 については、導入直後のタイミングから使用を開始したが、Fortran と C 言語でコンパイラの違いによる性能差が時折見られるものの、安定して利用できた。関係諸氏に深く感謝する。

参 考 文 献

[1] 東京大学 後期教養教育科目について

<http://www.u-tokyo.ac.jp/stu04/koukikyoyou.html>

[2] 東京大学 計算科学アライアンス <http://www.compsci-alliance.jp/>