

Wisteria 利用講義「並列数値計算」

須田 礼仁

東京大学情報理工学系研究科

1. はじめに

「並列数値計算」は情報理工学系研究科の大学院講義である。隔年で実施しており、2020年に続く開講となる。英語名は“Parallel Numerical Computations”としている。講義はスライド・説明とも英語で実施しているため、留学生が多く参加している。今年度は情報理工学系研究科、工学系研究科のほか、理学系研究科、数理科学研究科、総合文化研究科、新領域創成科学研究科、学際情報学府の学生を含め約50名が履修登録した。

2020年からは新型コロナウイルス感染症 COVID-19の影響を受けてきたが、2022年も多少影響が残存した。増減を繰り返しながら感染は継続したので、この講義もほぼオンラインで実施することとした。

2. 講義の概要

計算機の性能向上は並列性によって実現されている。今後まだコア数は伸びてゆくと思われるので、並列計算は誰にとっても重要な技術となってきた。膨大な計算量を必要とする近年の機械学習の隆盛もあって、GPUをアクセラレータとする計算機が広く使われるようになっていく。さらに、FPGAの性能も向上し、解く問題によっては最新のCPUやGPUを凌駕する高い性能を達成しているところである。

このような背景のもと、本講義では並列計算の一般論と各論とを分離し、一般論として述べられる点はできるだけ一般的な論述をするように構成した。これにより一般論が統一的な視点で理解できるようにしたつもりである。ただし、一般論部分は話が抽象的になりがちで実感がわきにくくなっているかもしれない。今回は後述の確認テストを新たに導入したことで、一定の効果があつたと感じたところであるが、今後とも継続的な改善を要するところである。一方、各論においては、特に大規模並列を念頭においてMPI, OpenMP, CUDAの3種類のプログラミングモデルを取り上げている。AVXなどCPUのSIMD機構については十分な説明をする時間がなかった。分散メモリのMPIを最初に詳しく説明して、その後OpenMPとCUDAをやや軽めに説明している。レポート課題はMPIのみである。近年注目を浴びているタスク型の並列性、PGASなどのメモリモデル、FPGAといった新しいプロセッサについては説明できていない。同じ情報理工学系研究科において、共有メモリ並列処理でタスク型のモデルもきちんと取り上げている講義があり、多くの学生はそちらも受講してくれているようである。しかし、情報理工学系研究科の中でFPGAによる高性能計算を教えている講義はないかもしれない。FPGAは数値計算で必要とする倍精度浮動小数点数は得意としていない点がネックであるが、注目されているアーキテクチャであるので、実際に触って演習することもできる講義があればよいのだが。

3. 講義の内容

令和4年度には12回の講義を実施した。ほとんどはzoomによるオンライン講義で、配信の

録画を復習用に学内限定で提供した。この講義での録画の活用状況は確認していないが、一般的には復習ができることが便利らしい¹。

以下に各回の内容の概要を示す。

第1回：イントロダクション

この講義で取り上げる並列計算とは何かを、類似性のある並行計算、分散計算と対比する形で定義した。特に、この講義では再現性があり予測可能な計算を取り上げ、動的な並列性は最小限とする。続いて、ハードウェア並列性の基礎概念として、複数演算器とパイプライン、SIMD と MIMD、分散メモリと共有メモリ、UMA と NUMA、およびそれらのハイブリッド、均一性・不均一性などを導入した。また、この講義では所要時間を主なメトリックとすること（スループットではない）、そこから高速化率、並列化効率、理想高速化率、スーパーリニアなどの概念を導入した。また、強スケールリングと弱スケールリング、アムダール則とグスタフソン則といった相対性能の基本法則を導入した。加えて、絶対性能として flops およびピーク性能を説明した。また、所要時間測定時の注意点についても説明した。

第2回：並列性と依存性

並列性とは依存性のないこと、よって並列性の解析は依存性の解析に他ならないことからスタートした。続いて、データ依存と制御依存、RAW/WAR/WAW 依存などの依存性の類型を示した。配列、リスト、木などのデータ構造へのアクセスの並列性、ステンシル計算と超平面法・ループスキュー、木によるリダクション、カスケードによるスキャン、3項漸化式の並列性、パイプラインなどの依存性と並列性を事例として挙げた。一方で計算順序が変えられる場合にはリオーダーリングにより並列性が変わること、その表現としてカラーリングが便利であることも示した。また配列への間接アクセスがある場合を取り上げ、その並列性の抽出方法には多数ありうることを示した。

第3回：局所性

並列計算は高性能を目指すものであるが、性能という観点では局所性は避けて通れない重要事項である。まずは分散メモリでも共有メモリでも局所性が低いと高い性能が期待できないことを説明した。局所性には計算強度と粒度の2つの側面がある。まず計算強度や B/F 値について説明し、行列積を例として、プログラムの書き方次第で計算強度が変わることを示した。また行列ベクトル積のような計算強度の弱い計算もあることも注意した。リダクション、FFT、ステンシル計算の計算強度について概観し、領域分割と Owner-computes-rule が局所性の観点から利点があることを説明した。次に粒度について説明し、パイプライン計算を例に並列性と粒度のトレードオフがあることを示した。最後に単純な通信性能モデルを導入し、計算強度と粒度が性能にどのように影響を与えるかを解析できることを示した。

第4回：スケジューリング理論

この講義では分散最適化問題に分類される古典的なスケジューリング理論についても紹介を

¹ タイパ (time performance?) なる言葉が 2022 年の流行語らしい。HPC はコンピュータ上でのタイパを追究することが、学問として確立したものと言えよう。

している。HPC 分野とはやや用語が異なるため、まずその点を注意した。そのうえで、タスクとスケジューリングの基本概念、不均一プロセッサの類別、メイクスパン、ガント図、グラハム記法などを導入した。 $P|C_{max}$ に対する LPT の最悪性能比、 $P|prec|C_{max}$ に対するリストスケジューリングの最悪性能比、 $P|intree, p_j=1|C_{max}$ に最適解を与えるレベルスケジューリング、 $P \infty|prec|C_{max}$ に最適解を与えるクラスタリングスケジューリングを説明した。また、クリティカルパス、タスク挿入、タスク重複などの概念を紹介した。不均一プロセッサやオンラインスケジューリングについては既知の性能オーダーを示すにとどめた。そのかわり、Divisible Load Theory およびループスケジューリングについて紹介した。

ここまでが一般論であり、このあと各論に入る。

第5回：MPI 入門

メッセージパッシング、local view、SPMD を説明したのち、MPI の基礎の対一通信を説明した。Eager と Rendezvous プロトコル、ブロッキング・非ブロッキングの違い、メッセージの到着順序などについて注意をしたうえで、リダクションとステンシル計算を題材に実際の簡単な MPI 並列プログラムを説明した。

第6回：集団通信

MPI の集団通信について、基本的な集団通信と、それを実現するためのいくつかのアルゴリズムを説明した。プロセッサ数とデータサイズにより、異なるアルゴリズムが最適になりうることを確認した。また Broadcast と Reduction などの双対性、Scatter と AllGather で Broadcast が実現できるなどの集団通信の組み合わせについて解説した。

第7回：分散データ構造

まず、分散メモリ並列計算におけるデータ構造のあり方の一般論を導入した。続いて、最も単純な1次元ベクトルについて、ブロック、サイクリック、ブロックサイクリック、可変長ブロック、要素ごと指定の5種類の分散方法を示した。続いて2次元の場合には列1次元分散、行1次元分散、2次元分散の選択があることを説明し、LU分解を例題として利害得失を考察した。またステンシル計算における袖領域を説明し、遅延隠蔽、piggy-backing を説明した。さらに、疎行列ベクトル積の事例、および動的負荷分散（この文脈では動的データ分散）の概念を紹介した。

第8回：OpenMP 入門

OpenMP の入門として、global view であること、また自動並列化ではなく正しさはプログラマの責任であることを注意した。基礎的な構文を説明したのち、OpenMP で陥りがちな誤りとして、共有・私有変数の区別、レース条件、弱い一貫性とメモリ同期の必要性を説明した。あわせて、reduction 節や atomic 節も紹介した。

第9回：OpenMP 高性能プログラミング

OpenMP の性能低下要因とその回避・軽減策を紹介した。排他制御のための構文である atomic、critical および lock 関数の違い、ループスケジューリングの選択肢、アフィニティに関する注意などを行った。また、キャッシュ周りの性能向上手法として、パディング、タイリング、ル

ープ交換，ループ結合，Arrays of Structures/Structures of Arrays などの手法を紹介した。また MPI + OpenMP ハイブリッド並列化の必要性について説明した。

第 10 回：CUDA 入門

アクセラレータという概念，GPU のハードウェア特性を導入した。続いて，CUDA の基本的な書き方の説明をした。ホストとデバイス，ブロックやスレッド，メモリ階層や同期など，しっかり理解しなければならないことが多いため，なかなか大変である。

第 11 回：GPU における高性能化

GPU において高い性能を達成するために重要な概念を説明した。GPU のハードウェアと CUDA の並列性階層の対応を説明し，SM ごとに走るブロックの数を決定する式を示した。またスレッドを実行する機構を説明し，並列性により遅延隠蔽が可能となることを説明した。分岐命令の削減，coalesced メモリアクセスの性能，CPU-GPU 間データ転送時間について注意を促した。

CUDA の更新は早く，講義のたびに新しい GPU と CUDA の仕様を確認している。今回はやや古くなってしまったが，ほぼ Pascal アーキテクチャに準拠して説明をした。

第 12 回：Wisteria ハンズオン

この回のみ対面とした。事前に配布しておいた Wisteria の使い方の資料についての不明点への回答や，アカウント登録・ログインや操作のトラブルへの対応について，必要な学生に指導を行った。

4. 課題

講義の最初の 3 回で，UNIX コマンドと C/Fortran プログラミングのごく初歩的な問題を 3 分で解いてもらった。「カレントディレクトリを表示するコマンドは何か」「.o で終わるファイルを全部消すコマンドを示せ」とか、「vi か emacs で，現在カーソルがある 1 行を消すコマンドは何か」とか、「整数変数 n と m を宣言せよ」，「Hello, World! を表示するプログラムを書け」というレベルである。4 年前は平均点が半分程度であったが，前回と今回は 7 割ぐらいまでできていた。オンライン環境であったので，その場で調べた学生もいるかもしれない。今回は，ITC-LMS 上で 3 分の制限時間で回答をしてもらったが，自動採点には向かないので，得点は ITC-LMS のアンケート機能による自己申告としている（成績には反映されない）。

また，ほぼ毎回の授業の後，その授業での内容の肝のところを確認する主旨で，「Check your understanding」と称する簡単な復習問題を解いてもらった。こちらも ITC-LMS のテスト機能を用いており，全問選択式にして自動採点で確認できるように設定した。内容はごく簡単なものであるが，話を聞くだけになりがちなオンライン授業に，多少のアクティブな要素を入れられるので，一定の効果があつたと感じている。特に，2 年前，前半の一般論の内容を後半に入ると忘れてしまっている場合が散見されたのだが，今回はそのような例はほとんどなかった。

さらに，全部で 3 回のレポート課題を課した。最初の 2 つの課題は各自が使える計算機により行い，これらの課題を提出した学生には Wisteria のアカウントを配布して，第 3 の課題を行わせた。

第1回：CPU モデルと計時方法の確認

各自のプロセッサのモデルを確認し、ピーク性能を計算させた。また、計時方法を適当に選ばせ、その計測精度を測定により求めさせた。

第2回：Flops への挑戦

「どんな計算でもよいので、できるだけ高い flops 値を出すプログラムを書け」という課題を出した。また、これに加えて、十分長いベクトルのコピー、内積、和の性能を測定させ、それらからメモリスループットを測らせた。

例年は、問題の主旨（高い性能はどのような場合に出るか）がわからない学生もちらほら見受けたが、今年はそのような学生はほとんどいなかった。これも「Check your understanding」の効果かもしれないと思っている。

第3回：MPI プログラミング

密行列計算、偏微分方程式の数値解法、もしくは個人的に興味のある計算のいずれかを選んで、MPI でプログラミングをして Wisteria で実行し、弱スケーリングと強スケーリングでの性能を報告させた。前回に引き続き、Cholesky 分解と熱伝導方程式については逐次プログラムをダウンロードできるようにして、それを修正して課題として提出してよいこととした。

以前は、並列性ばかりに注意が行って、局所性や通信コストを忘れる学生もいたが、今回はそのような学生はほとんどなかった。

なお、今回は GPU の話ができた（2年前はコロナの影響でできなかった）ので GPU を使ってもらってもよかったのだが、Wisteria が CPU 並列部分と GPU 並列部分にすばっと分かれているところで、少し残念な思いがした。

5. おわりに

全体として講義はスムーズに進み、Wisteria のアカウント発行も問題なくできた。例年はアカウント発行の申請が遅れてしまう学生が数名出るのだが、今年度は1名だけで、事前に準備していただいた予備アカウントで足りた。オンライン授業に学生が慣れてきたということなのかもしれない。スーパーコンピュータを用いた講義という貴重な機会を提供していただき、いつものようにもろもろご迷惑をおかけしましたが適時にご対応いただきました、情報基盤センターの皆様方に感謝を申し上げます。