

# FX10 教育利用報告「並列数値計算論」

須田 礼仁

東京大学情報理工学系研究科

## 1. はじめに

「並列数値計算論」は東京大学情報理工学系研究科コンピュータ科学専攻の大学院向け講義であり、隔年で夏学期に開講している。主に修士の学生を想定しているが、博士課程学生でも受講できる。2014年度の講義では、情報理工学系研究科と工学系研究科の学生が、あわせて20名ほど履修した。

## 2. 講義の概要

講義は英語で行っており、英語のタイトルは“Parallel Numerical Algorithms”である。英語開講のため、留学生の履修も多く、2014年度の講義では約1/3が留学生であった。

マルチコアプロセッサが広く普及し、またGPGPUも利用しやすくなっていることから、並列計算は一段と多くの人々が触れる技術となっている。今後は数値計算以外の様々なアプリケーション分野において並列計算が広く用いられることになるものと思われる。新しいアプリケーションの台頭は、新しい並列処理のプログラミングパラダイムの誕生を誘発する可能性も考えられる。これらのことを考慮し、2014年度の講義では、並列計算の一般論と特定のパラダイムに依存する部分をできるだけ明確に分離する方針とした。一方で、事例として挙げる具体的なアルゴリズムやデータ構造は数値計算から採用することとした。前提知識としてはCまたはfortranによるプログラミングを要求し、数値計算に関する知識は必ずしも必要としない立場とした。

講義は全13回行うことができた。各回の主な内容を以下に示す。

第1回：並列アーキテクチャと並列性能。まず、並列計算とは何かを定義し、対照として並行計算、分散計算との違いを示した。次に、並列計算機アーキテクチャの基本的な概念として、複数演算器とパイプライン方式、SIMDとMIMD、共有メモリと分散メモリ、均一なアーキテクチャと不均一なアーキテクチャ等を紹介。次に、並列実行時間を定義し、そこから導かれる高速化率、並列化効率を紹介。強スケーリング・弱スケーリングの考え方と、Amdahl則・Gustafson則を紹介。また、絶対的な性能指標としてflops、ピーク性能の考え方を示した。

第2回：依存性とその解決。並列性とは依存性（逐次性）のないことであり、依存性を分析することが並列性を見つけることであることを指摘。データ依存と制御依存、逆依存・出力依存などの基本的分類を示す。リスト、木、ステンシル計算、リダクション、スキャンなどを事例として、依存性の分析、および並列性の抽出法として超平面法、ループ傾斜、カラーリング、カスケード、アトミック演算、投機実行などを紹介した。

第3回：局所性。局所性とは何か、なぜ局所性が並列計算において重要かを導入。計算強度の

概念を定義し、行列積を事例としてアンローリングなどのループ変換によって計算強度が変わることを示した。行列・ベクトル積、リダクション、ステンシル計算、FFT などにおいて局所性の高い計算分割法を例示。また、メモリやネットワークのバンド幅と計算強度が相互に関係して性能を決定する要因のひとつとなることを説明した。続いて粒度の概念を定義し、これがメモリやネットワークの遅延と相互に関連することを説明した。パイプラインを事例に、並列性を犠牲にして局所性を高めた方が高速な場合があることを説明。

第4回：スケジューリング理論。タスクと実行時間、プロセッサのモデル、先行制約、スケジューリングの概念を導入。ガント図、グラハム記法、リストスケジューリング、クリティカルパスとレベルスケジューリング、タスク挿入・タスク重複などを紹介し、計算複雑性に関する基本的な結果を説明した。また、Divisible Load Theory とループスケジューリングも紹介した。

第5回：MPI 入門。ここから分散メモリ並列計算に入る。SPMD を導入した後、MPI の基本的な使い方を説明した。

第6回：通信性能と集団通信。Affine な通信性能モデルを紹介して、通信ブロッキングを説明。次に、集団通信の主なものについて紹介し、いくつかの実装方法とその所要時間の解析を行った。

第7回：並列探索。ナント大学の Florian Richoux 准教授により、並列探索とその性能分析について講義していただいた。

第8回：分散データ構造。まず分散メモリにおけるデータの配置の様々な方式について概論。次に1次元配列のブロック分割、サイクリック分割、ブロックサイクリック分散、可変サイズブロック分散などを紹介し、特性を比較。LU 分解を事例として、2次元配列の分散の特性を論じた。次にステンシル計算を事例として、袖領域、piggy-backing を紹介。疎行列・ベクトル積を簡単に見たのち、動的負荷分散（データ再分散）について説明した。

第9回：OpenMP 入門。OpenMP の基本的な使い方を説明。間違いやすい点として、共有・私有変数の区別、レース条件、弱い一貫性モデルについて注意した。

第10回：FX10 の使い方。情報基盤センターの大島聡史助教により、FX10 の使い方および高速化のポイントを説明していただいた。

第11回：高性能な共有メモリ並列計算。同期オーバーヘッド、ループスケジューリングによる動的負荷分散、ファーストタッチ、アフィニティ、false sharing を説明。共有メモリへのアクセス遅延を改善するための方策として、局所性の重要性を指摘し、各種ループ変換、cache oblivious algorithm、AoS/SoA などを紹介した。

第12回：GPUとCUDA入門。演算加速器という概念、GPUのアーキテクチャを最低限説明しながら、CUDAプログラミング入門を説明した。

第13回：高性能なGPGPU計算。GPUのアーキテクチャをより詳細に検討し、並列性、レジスタ使用量、遅延隠蔽、ワープ分散とその低減、正則メモリアクセス、計算粒度などについて説明した。

### 3. スーパーコンピュータの利用

レポート課題として、(1)性能測定の基本、(2)FX10を用いたMPIの初歩、(3)FX10を用いた自由課題、の3つに取り組んでもらった。最初のレポート課題は手元のPCを使う課題とし、これを提出した学生にFX10のアカウントを発行し、2つ目および3つ目の課題に取り組んでもらった。

2つ目のMPIの初歩では、ピンポン通信、AllReduceの自前実装、ステンシル計算から1つを選び、性能を分析する課題を与えた。

3つ目の課題は自由課題としたが、偏微分方程式の数値計算、LU分解、ソート、スキャン、Particle Swarm Optimizationなど、多様な問題に取り組んだレポートが提出された。

3つ目の課題では、なかなか並列性能が得られない学生も見受けられた。主な高速化のコツは紹介してあるものの、自分の問題においてどれを適用したらよいか、すぐにはわからないことも多いようである。全体的に課題提出のスケジュールを前倒しし、事例ごとに高速化のヒントを具体的に挙げるのが望ましいことが明らかになった。

### 4. おわりに

大学院講義「並列数値計算」では、情報基盤センターのご協力を得て、FX10を用いた並列計算の入門を講義することができた。特に、大島先生によるFX10の使い方の講習は非常に有効であった。関係諸氏に、この場を借りて感謝の意を表す。

内容的には基本的なところに留めるように常に留意し、HPC屋らしい性能を深く最適化する話は封印した。講義終了後にアンケートがあったが、講義のレベルについての設問は「易すぎる」「難すぎる」などは少なく、「適切」が多かった。修士1年生向けの並列計算入門としてはこの程度が適切なのであろうと思う。

一方でレポート課題においては学生の理解不足も見受けられた。今回程度を受講者であれば個別指導的な方法で対処することも考えられる。より受講者が多い場合には、助教やTAを付けて演習形式で行うか、取り上げる課題を指定するなどの対応が必要であらう。これらの反省を今後の授業に活かしていきたい。