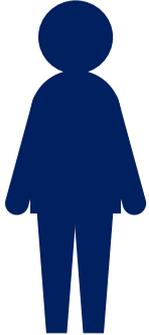


非圧縮性流体シミュレーションコードMiFlowの

X^{calable}MPによる実装と評価

神戸大学大学院 システム情報学研究科
横川研究室 大河原 修

並列化の需要はどこにあるか？

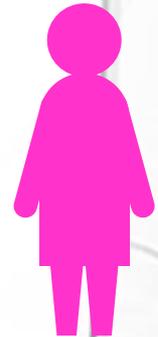


高速化を専門とする計算機科学の研究者：

MPIを使って並列プログラミングするのは大変なので、
もっと生産性を上げたい

シミュレーションを行いたい研究者：

研究に必要なシミュレーションコードを書いてはみたけど、
並列化しないとまともな速度で動かない
でも、専門家ではないから並列化するのは大変



研究背景

- XcalableMP(XMP)は開発中のPGAS言語である
 - 先ほどのような人々に対して、より**簡便**な並列プログラミング環境を提供し、**高性能**に実行できるようにすることを目指す
 - 正式版のリリースに向け、XMPを「**枯れた技術**」にしないといけない
 - 技術を枯らすには、様々な人間がテストを行う必要がある
- 「並列化の初心者」のレビューが必要
 - XMPは「シミュレーションを行いたい研究者」の需要に応えられるか？

研究概要

- 発表者は並列計算というフィールド自体の初心者である
 - 本研究開始時には、ソフトウェアのビルドすらままならない状態
 - シミュレーションコードを見るのは本研究が初めて
 - 当然、MPIの利用も本研究が初めて
- シミュレーションコードの並列化をMPIとXMPの両方で行い、**初心者の立場からのレビュー**を行う

発表の流れ

- 研究背景・研究概要
- 並列化を行うシミュレーションコードについて
- 作業の経過
- XMPを用いたプログラミングに対する評価
- まとめ
- 今後の課題

並列化を行うシミュレーションコードについて

- MiFlow
 - 3次元非圧縮性流体シミュレーションコード
 - FORTRAN77で書かれた古いコード
 - 圧力のポアソン方程式は、差分法によって連立一次方程式を立て、4-color SOR法で解く。時間発展は時間分割法による。
- 4-color SOR法
 - MiFlowで用いられている連立一次方程式の反復解法
 - 第3次元が奇数である格子点について解いた後、偶数である格子点について解く

作業の経過

並列化に慣れる

- 簡易なサンプルを作って並列化し、動作を確かめる

MiFlowの並列化

- Fortran 90への書き換え
- MPI→XMPの順に並列化

評価

- プログラミング環境への定性的な評価
- 実行速度への定量的な評価

作業の経過

並列化に慣れる

- 簡易なサンプルを作って並列化し、動作を確かめる

MiFlowの並列化

- Fortran 90への書き換え
- MPI→XMPの順に並列化

評価

- プログラミング環境への定性的な評価
- 実行速度への定量的な評価

MiFlow並列化の経過

- アルゴリズムをあえて詳しく学ばず、**コードのみを見て**並列化
- しかし、不適切なデータ分散を行ってしまい、通信が複雑化
 - 最初にコードの内容を見ずにデータ分散を決め、それに従うようMPIで通信を生成したため
 - 通信が複雑になりすぎて**XMPで再現できず**、差し戻し
- 現在、適切なデータ分散での実装がMPIのみ完了
 - XMPでも正しい計算結果を出力してはいるものの……

プログラミング環境への定性的な評価

並列化を修得済ならば生産性は上がる

- XMPを使えば、並列化に携わった経験のある人々の生産性は上がると思われる
 - 指示文はシンプルで使い回しが効きやすい
 - 書換え箇所がほぼ同じなのでMPIを使える人ならば容易に修得可能
- 初心者がいきなり並列化を行うのは無理
 - 通信箇所・データ分散などを自分で考えられるレベルには達していません
 - もちろん、これはMPIを使った並列化を学ぶ場合でも同じ

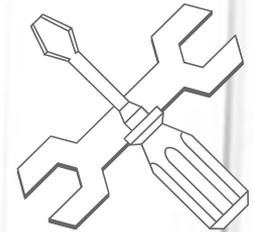
デバッグが非常に大変

- MPIの実行エラーが出た場合、バグの特定が困難である
 - XMPでは中間コードを生成し、それをコンパイルして実行しているため、実行環境の知らせているエラー箇所がコードと対応しない
- **参照できないデータがある**ため、コンソール出力によるデバッグが出来ない場合がある
 - 例) リダクション演算を行う直前に各ノードが持っている変数の値
 - 例) 袖領域に指定されているデータ
- **トラブルシューティングの蓄積**が必要である



チュートリアルの内容が物足りない

- ファイル分割を行う際、データ分散はどう指定するか？
 - 少なくとも公式サイトには言及がなく、仕様書でも記述が少ない
- MPIとの相互運用の具体的な方法が示されない
 - 公式サイトには「可能である」と記述があるのみ
 - Fortranについては仕様書にもサンプルが未掲載
- 組み込み関数についての具体的な言及が仕様書のみ
- 現状、本当に**最低限**しか用意されていない



プログラミング環境への評価のまとめ

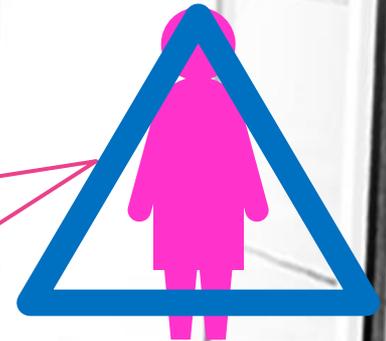


高速化を専門とする計算機科学の研究者 :

MPIを使って並列プログラミングするのは大変なので、
もっと生産性を上げたい

シミュレーションを行いたい研究者 :

研究に必要なシミュレーションコードを書いてはみたけど、
並列化しないとまともな速度で動かない
でも、専門家ではないから並列化するのは大変



プログラミング環境への評価のまとめ

- 並列化を修得済みの人には良いプログラミング環境である
 - MPIと同じ箇所に指示文を挿入すれば良い
 - 指示文はMPIより簡便な上に使い回しが効きやすい
- プログラミング自体は簡単でも、初心者にはあまり優しいとは言えない状況
 - デバッグがしにくいのにトラブルシューティングの蓄積がない
 - Web上のドキュメントが最低限で止まっており、そこから先に進むことが出来ない

どのようにして改善するか？（案）

- 現状、困ったときにググっても何も出てこないのは問題
 - 初学者が自己解決できる環境がない
 - しかし、科学技術計算向けの言語なので、今後普及したとしても
Web上に情報が充実することは考えにくい



- Webサイトでのサポートを充実させるべき
 - トラブルシューティングを皆で蓄積できるようにする
 - チュートリアルを誰にでも書ける状態にする

今後の課題

- XMPの実行性能について定量的な評価を行う
- 現在使っているものと別の計算機上にて実行性能を調べ、移植性の評価を行う

ご清聴ありがとうございました。