

大阪大学大学院情報科学研究科

コンピュータサイエンス専攻

伊野研究室

並列処理工学講座

教員

教授：伊野 文彦

准教授：置田 真生

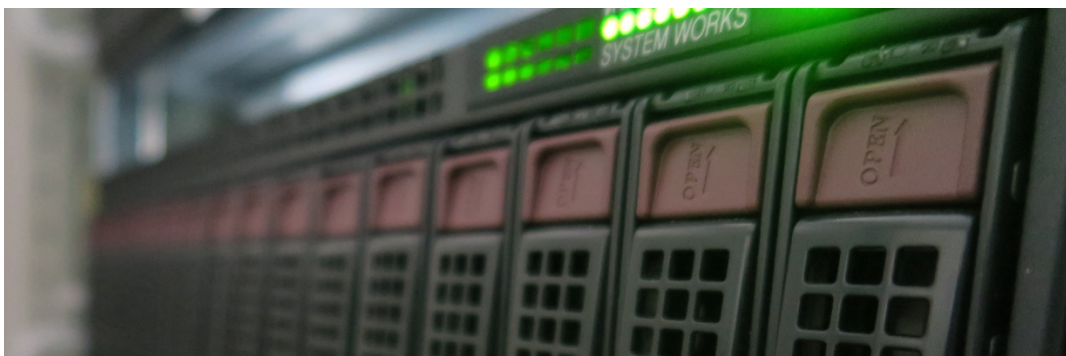
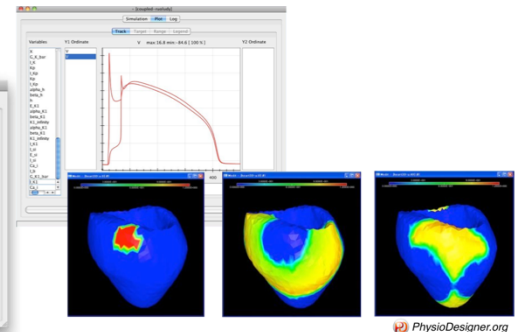
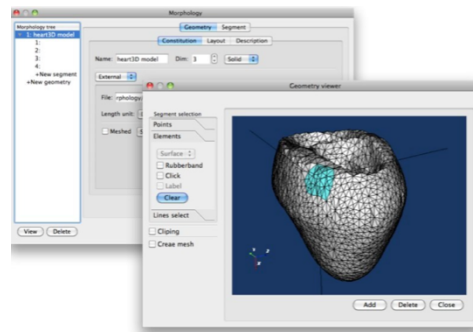
助教：榊井 晃基

連絡先

〒565-0871 大阪府吹田市山田丘 1-5 情報 B 棟 4 階

TEL : 06-6879-4353 FAX : 06-6879-4354

<http://www-ppl.ist.osaka-u.ac.jp/>



研究室の様子 ～ どんな感じなの? ～

特徴

よく学び よく遊び よく食べ よく遊び よく寝る

年間行事

2024年度(予定)

- 4月 新歓コンパ
- 8月 院試
- 9月 院試就活お疲れ会
研究室旅行
- 12月 忘年会&クリスマスパーティ
- 1月 新年会
- 2月 卒論・修論発表&打ち上げ
- 3月 追い出しコンパ

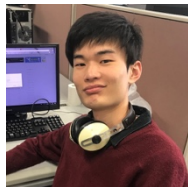


福利厚生

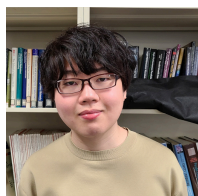
- ボードゲーム (麻雀, ドミノオン, カタン, ニムトなど)
- テレビゲーム (PS3, Wii U, レトロゲームなど)
- 書籍 (ジャンプ, マガジン, Software Design など)
- To3 商店 (後払い制の売店)



メンバー紹介 ～ どんな人がいるの? ～



D3 李彦辰
・インドカレーを自制したい



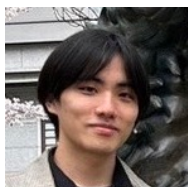
D1 青山 昂生
・趣味で競技プログラミングをしています



D1 王 珏
・アルパカを飼いたい



M2 荒川 雄登
・ゲームと読書が大好きです



M2 末永 楓馬
・彼女と Raspberry Pi が大好き人間



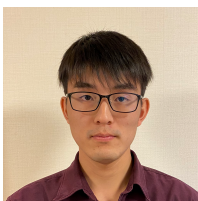
メンバー紹介 ～ どんな人がいるの? ～



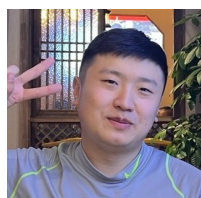
M1 張 梓航
・ノー豆腐, ノーライフ



M1 上木 脩生
・趣味はMLB観戦



M1 中野 秀紀
・趣味はゲームとプロレス観戦



M1 陳 子越
・筋トレしないと眠れない



メンバー紹介 ～ どんな人がいるの? ～



教授 伊野 文彦
・誰もが認める日本酒の神

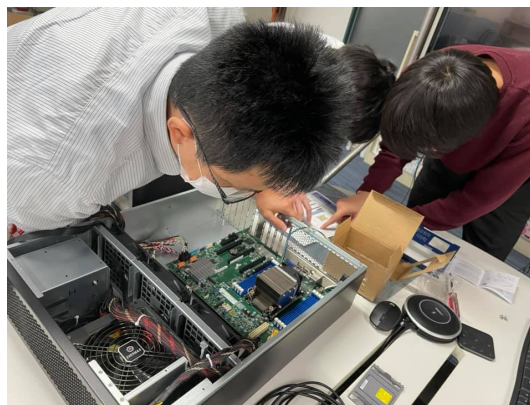
准教授 置田 真生
・近頃、サッカー熱が再燃中



助教 榊井 晃基
・スポーツと数学が大好き



秘書 笹川 知子
・穏やかでやさしい秘書さん



研究内容 ～ どんな研究をしているの? ～

研究目標

ハイパフォーマンスコンピューティング技術による高度情報化社会への貢献

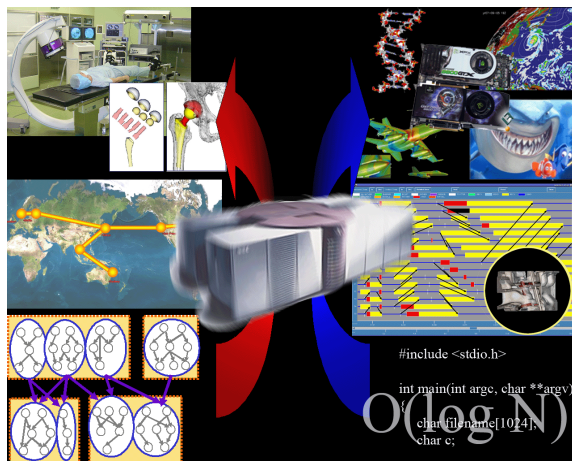
自動運転技術や自動翻訳など、近い将来には解決できないと思われていた問題が、深層学習によって解決されつつあります。これらの画期的な成果は、数千個もの演算器を内包するGPU（Graphics Processing Unit）をはじめとする、大量の計算を高速化できる超並列計算技術が実現したのですが、その技術に精通している人材が不足しています。また、ポストムーア時代を見据えた量子計算への期待とともに、社会は数百万個の演算器や量子コンピュータを自在に扱える人材を強く要望しています。本研究室では、高性能ソフトウェア開発の第一線で活躍できる専門家を育成するために、企業との実用化や異分野研究者との共同研究を通して下記のテーマに取り組んでいます。

研究チームの一覧

- ・ GPU Computing：GPUを開拓するための応用開発
- ・ Deep Learning：深層学習の基盤となるソフトウェア開発
- ・ Automated Parallelization：並列プログラミングの自動化
- ・ Quantum Computing：量子力学の原理に基づく超並列計算
- ・ Large-scale Computation：大規模数値計算を実現するシステム開発

研究キーワード

GPU コンピューティング、高性能計算、医用画像処理、3次元可視化、量子情報科学、科学シミュレーション、クラウド・クラスタコンピューティング、自動並列化。

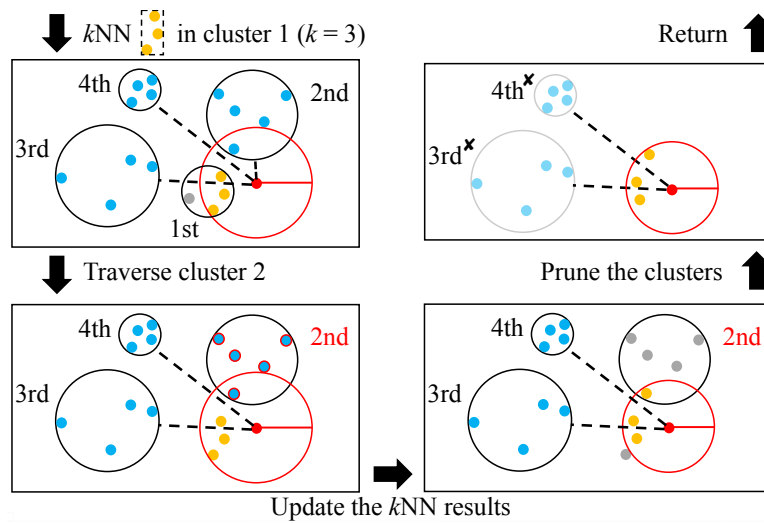


研究紹介 ~ GPU : GPU を開拓するための応用開発 ~

GPU 向けの並列アルゴリズムの開発

目的 : GPU を用いた最近傍探索手法の高速化

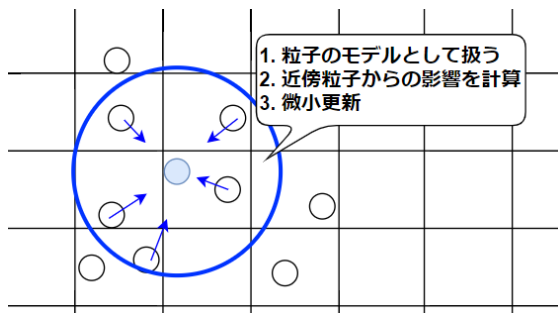
- ・ 様々な次元数や距離関数に対応でき、汎用性が高い最近傍探索手法を開発する
- ・ クラスタリングしたデータをソートし、クラスタの枝刈りによって計算量を削減する
- ・ GPU 性能を最大限に引き出す並列アルゴリズムを開発するところに醍醐味がある



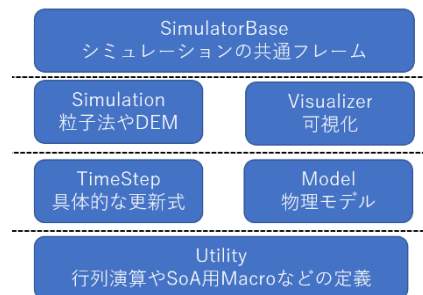
GPU を用いた物理シミュレーションの高速化

目的 : 弾塑性変形シミュレーションの高速化

- ・ 最新の GPU アーキテクチャを用いてシミュレーションの高速化を図る
- ・ 雪は流体と固体の両方の性質をもち、弾塑性変形によって運動を記述できる
- ・ 高速化によって FPS が向上し、CG 分野に貢献できる



粒子法のアルゴリズム



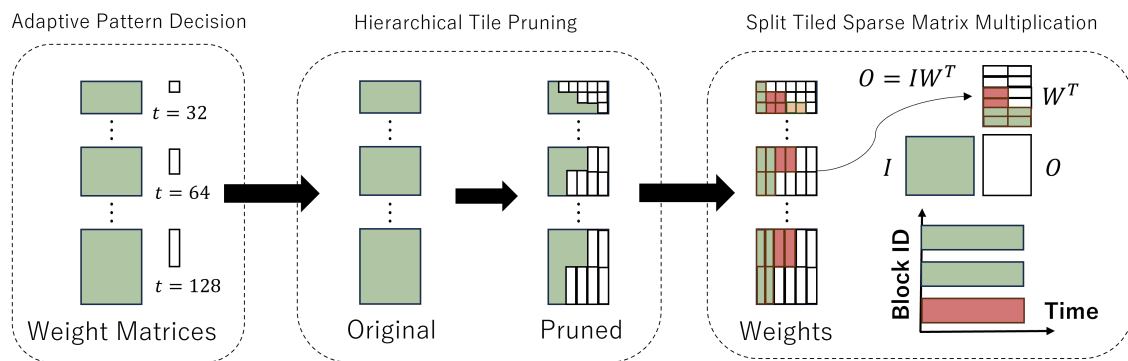
クラス設計

研究紹介 ～ DL：深層学習の基盤となるソフトウェア開発 ～

ニューラルネットワークの圧縮

目的：深層学習モデルの空間・時間計算量を削減する

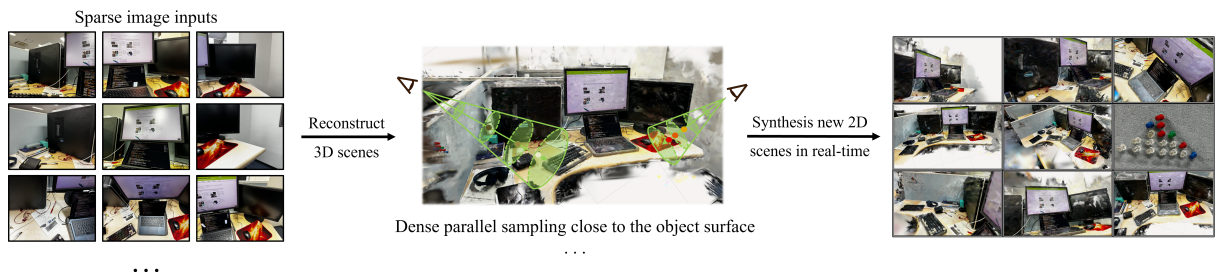
- ・ AIの進化を促すために、深層学習モデルの巨大化が進む
- ・ 不要なパラメータを削除する枝刈り法を改良して、モデルの高速化・軽量化を目指す
- ・ 深層学習モデルの動作原理を理解できる



三次元シーン再構築の高速化

目的：画像AI “NeRF” の最適化

- ・ 高精度に3次元シーンを再構成できる深層学習モデル “NeRF” が注目を集めている
- ・ NeRFモデルの訓練と推論は膨大な時間を必要とするので、高速化が望まれている
- ・ 高性能計算の技術を駆使して、NeRFの構造や計算を効率化
- ・ Take photos, and share 3D moments with your friends in a short time!

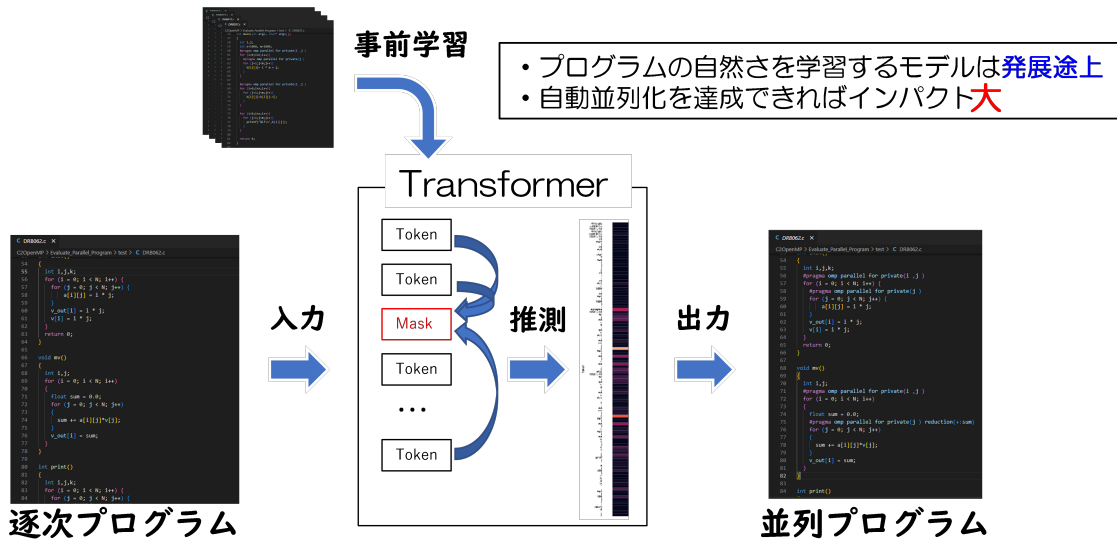


研究紹介 ～ AP：並列プログラミングの自動化 ～

プログラムの自然さに基づく自動並列化

目的：自動並列化の可能性を拡張する

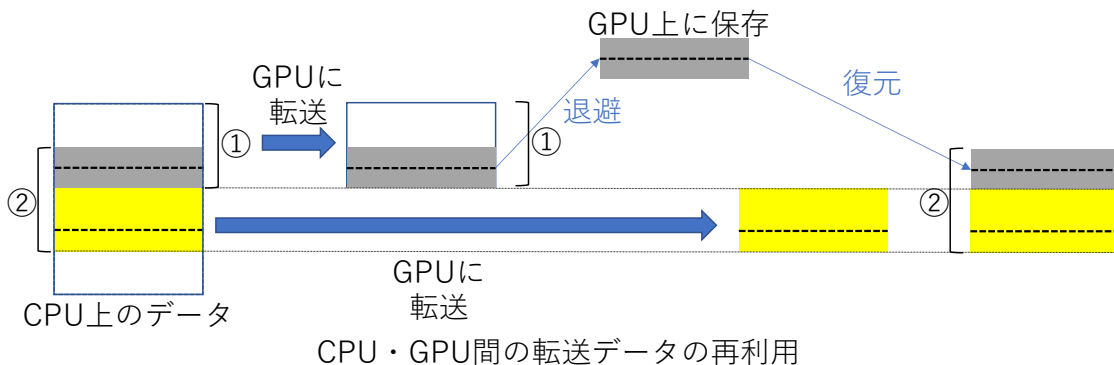
- ・ プログラムの自動並列化は、ごく一部の範囲しか成功していない難しいテーマ
- ・ “並列プログラムの自然さ” を利用する新奇的発想が可能性を秘めている
- ・ 深層言語処理を土台に、これまで不可能だった自動並列化の新領域を開拓したい



高速化手法が効果を発揮する条件のモデル化

目的：高速化手法を実装すべきか判定する

- ・ プログラムの高速化手法が有効かどうかは計算機環境に依存
- ・ CPU・GPU間の転送データの再利用がプログラムの高速化に寄与するかどうか判定
- ・ 高速化手法の実装前に効果の程度を見積もろうとする点が難しいかつ面白い

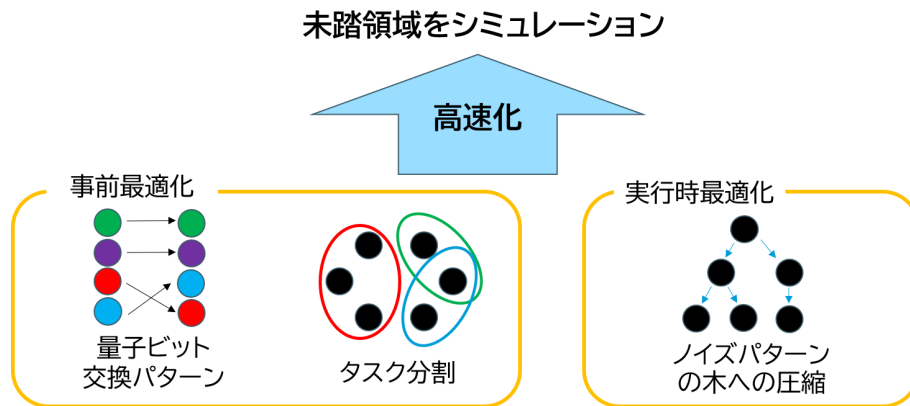


研究紹介 ～ Q：量子計算機のソフトウェア ～

量子回路シミュレーションの高速化

目的：量子計算機の計算能力を測る

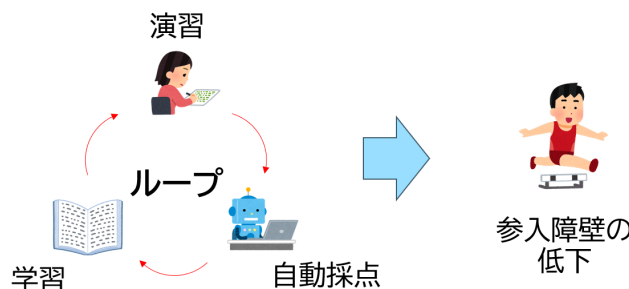
- 量子計算機の実力を測るためには、まず古典計算機の限界を知る必要がある
- 古典計算機上で動作する量子回路シミュレータを高速化
 - 量子ビットの交換パターンやタスク割り当てパターンなどの事前最適化
 - 量子回路の特性と途中計算結果を活かした実行時最適化
- 高速化によって、実行時間の問題で難しかった未踏領域のシミュレーションを現実的な時間内に実行することが醍醐味



自動採点を用いた量子教育

目的：情報分野と量子分野の分断を繋ぐ量子人材育成

- 情報系の人間にとって、量子情報分野は新たに学ぶ知識が多くハードルが高い
- テキストと演習問題の高速ループによる教育手法を開発
 - 演習問題への解答を自動採点することにより、自律的な学習を促進
- 目指すべき未来を考え、その未来に必要なパーツを教材として実装する点が醍醐味

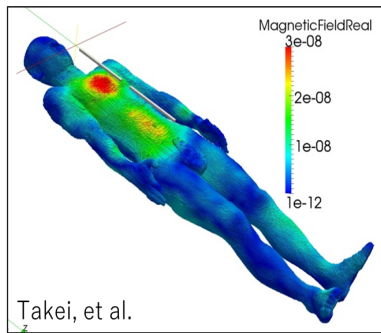


研究紹介 ～ L：数値計算 ～

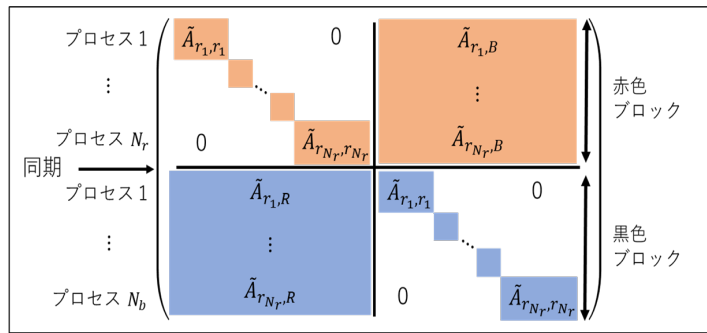
大規模数値計算を実現するシステム開発

目的：高性能な電磁場解析のための反復法の開発

- ・ 電磁場解析の過程で現れる複素数の線形方程式を前処理付き反復法で解く
- ・ 変数の数が 1,000 万個以上の大規模な連立一次方程式を計算するため計算に膨大な時間を要する（場合によっては数か月かかることも）
- ・ スーパーコンピュータ向けに計算領域を分割する
- ・ 電磁場問題の行列の特性に応じた効率の良い並列化手法を考案する点が面白い



大規模電磁場解析における可視化



反復法における前処理計算の並列化手法

複素数問題向け反復法の高速度化

目的：収束性の悪化を防ぎつつ反復ごとの演算回数を削減

- ・ 大規模電磁場解析において、収束性への影響が小さい前処理行列の虚部を切り捨て
- ・ 虚部の格納に使用していた領域分のメモリを削減
- ・ これまでに研究例がない虚部の切り捨てによる効果を発見する点が面白い

$$(L_{re} + iL_{im}) \times (y_{re} + iy_{im}) = (L_{re}y_{re} - L_{im}y_{im}) + i(L_{re}y_{im} + L_{im}y_{re})$$



虚部の切り捨てにより計算コストを削減

$$L_{re} \times (y_{re} + iy_{im}) = L_{re}y_{re} + i(L_{re}y_{im})$$

前処理行列を用いた乗算

研究環境

個人マシン

- ハイエンドGPU装備
- ディスプレイ (1台 or 2台)



クラスタマシン

- 10万個以上の演算器



最後にちょっとだけ Q&A

Q. 研究は忙しい?バイトとか...

A. 好きな時間に研究できます。バイトもOK

Q. コアタイムはありますか?

A. 登校時間はわりと自由です

Q. プログラミングしますか?

A. C++, Java, Python など色々