



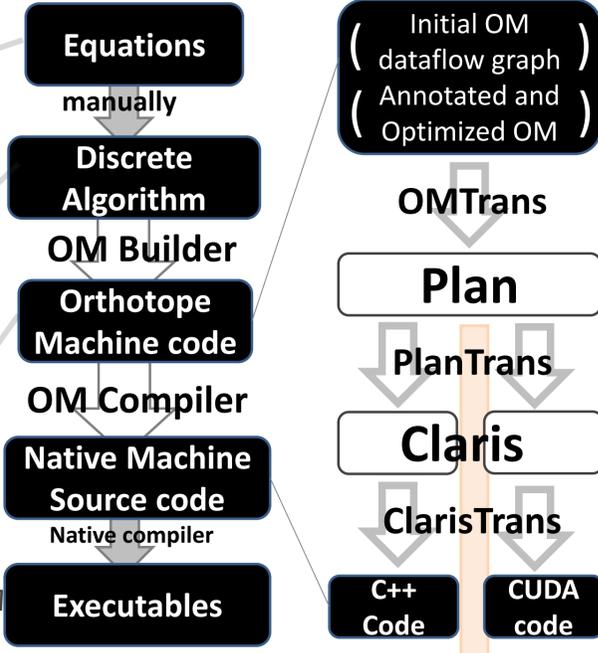
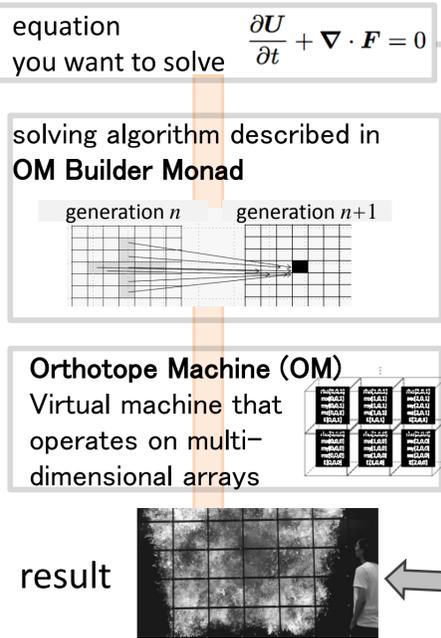
これまでに実現したこと

テンソル記法などが使える
数学的・簡潔な記述言語

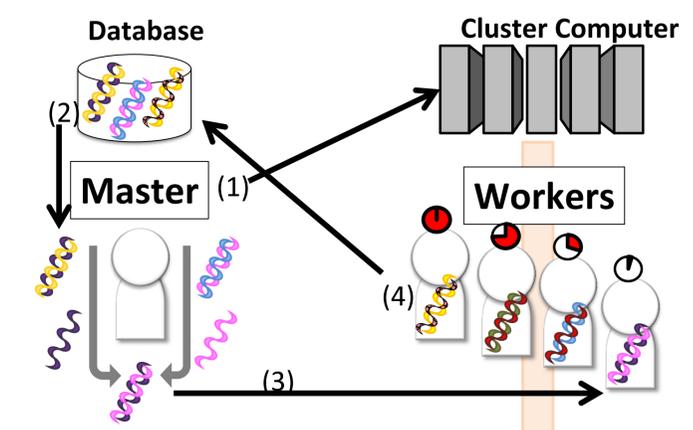
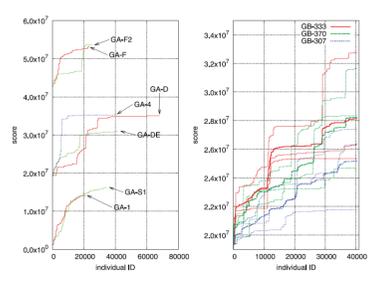
OpenMPおよびCUDA向けの
コード生成

遺伝的アルゴリズムに基づく
自動チューニング

実際速い



Analysis & Optimization



- (1) 空ノードの問い合わせ
- (2) 新種の合成、ジョブ投入
- (3) 並列・非同期ベンチマーク
- (4) スコアの書き込み

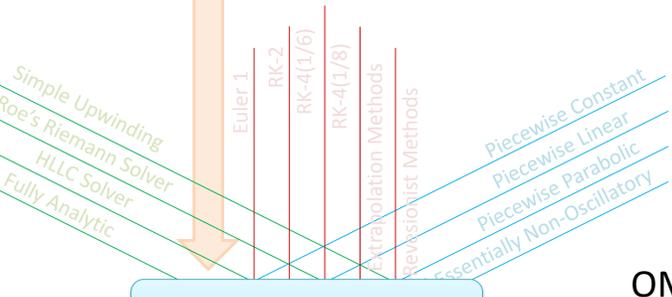
見えてきた課題

アルゴリズムの要素を
modularな形で持つ
Paraisoだからできること

勃興する並列言語たち
を有効利用したい

チューニングが局所最適に
陥っていないか？

自動チューニングに
膨大な所要時間



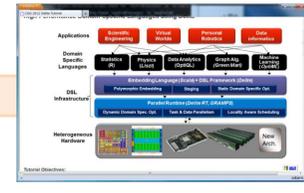
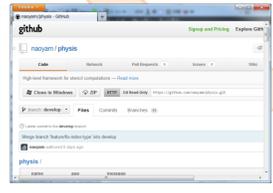
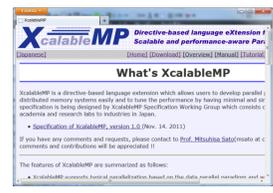
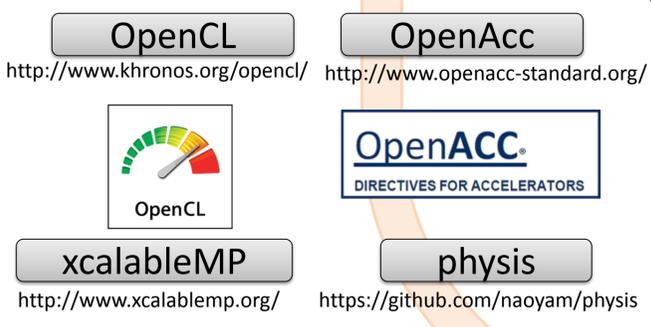
Combo Paper

さまざまな(まずは教科書的な) Riemannソルバ、空間補間法、時間積分法。それぞれを他の要素とは独立な形で持てることを示し、その組み合わせを網羅的に生成し、安定性や速度を評価できることを示す。もってアルゴリズムの自動選択や新規コードの開発に貢献する。

OM Specification

多次元配列仮想並列マシン(OM)の正確な仕様を策定。

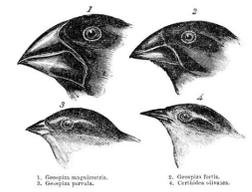
OMプログラムから多様な並列言語のプログラムを生成できるようにするとともに、それらの中での性能比較やテスト、相互乗換えを可能にする。



Perf
演算器や各所の帯域幅の利用率、キャッシュミスヒット率、コード長など、パフォーマンスに直接・間接的に影響するさまざまな性能指数を収集

Niche

ベンチマークスコアだけではなく、多次元の評価基準を設けることにより多様な種を生成。それらの交配により局所最適解の壁を突破



Reinforcement Learning

強化学習によりベンチマーク候補を提案、チューニングの効率を上げられるか？

Predict

性能指数や、ベンチマークスコアを予測できるか？単一のアプリケーションを超えた知識は学習できるか？それを実現できたら、プログラムやそれに施す変換をどう表現したらよいか？

...への挑戦!!