

HPCI FS第4回全体ミーティング フルアプリ調査報告・ミニアプリ化方針

鈴木惣一郎、丸山直也(理研AICS)

2013年1月21日

目次

1. 第2回フルアプリ募集

1. ミニアプリ化方針

古典分子動力学アプリを例に、ミニアプリ化作業について説明

第2回フルアプリ募集スケジュール

- 近日中:
「フルアプリ申請テンプレート」をMLに流します
- ~2/28: フルアプリ提供受付
上記テンプレートに記入の上、遅くとも2/22ぐらいまでに返信してください。折り返し、ドキュメント定形フォーム、アップロード手順をご案内します
- 2/28: フルアプリ提出締め切り
締め切りまでに、指定サーバにアップロードをお願いします

(第3回フルアプリ募集は5月末を予定)

ミニアプリの取り扱い

- ソースコードをプロジェクト終了後速やかに公開します
 - ライセンスはGPLもしくはApacheの2択
 - 著作権者はオリジナルアプリ著作権者＋ミニアプリ作成実施作業
- プロジェクト実施期間内はアプリFSおよびアーキFS内にて共有
- ミニアプリ開発で行った作業のフルアプリへのフィードバックはライセンス・著作権規定に従う限り自由
 - ミニアプリがGPL → ミニアプリのコードをフルアプリで使い場合はフルアプリもGPLにする必要あり(ミニアプリがApacheであれば特に制限なし)

フルアプリの取り扱い

- フルアプリのミニアプリ作業班への提供は必須ではありません
 - ミニアプリ化をフルアプリ開発者側で行う場合
 - ミニアプリの提供だけでかまいません
 - ミニアプリ化をFSミニアプリ作業班で行う場合
 - フルアプリの提供が必要です
- フルアプリを提供された場合
 - アプリケーションの情報はアプリFS内で共有します
 - アプリの名前、開発者、計算内容、など
 - ソースコートは**ミニアプリ作業班内**でミニアプリ化のために共有します
 - ミニアプリ作業班メンバー
 - AICS:丸山、鈴木、松田
 - 東工大: 松岡、野村、遠藤、佐藤、滝澤、Pericas, 青木、下川辺、小野寺
 - その他のケースについては、適宜開発者側に確認します
- アーキテクチャFSの3チームには共有可能かどうかはアプリ次第とします

フルアプリの条件

- 言語（以下のいずれか）
 - C99, C++ (C++0xはのぞく)、Fortran 2003
 - その他スクリプト系 (Perl、Python、Ruby)
 - できれば京で動くものが望ましいが必須ではない
- Intelマシン＋Linuxで動作確認がとれていること
 - 性能チューニングがされているかどうかは問わない
- 並列動作すること
 - 並列化の手段は問わない (MPI、OpenMP、自動並列化、アレイジョブ、等)

フルアプリ提供前の準備

- 確認事項
- プログラムの整備
- サンプル入力データの用意
- 実行結果検証手順の用意
- ドキュメントの整備

フルアプリ提供前の準備(続き)

- 確認事項

- 将来的にミニアプリとして公開されることの確認
- 「フルアプリの条件」に合致することの確認
- ミニアプリのライセンス(Apache or GPL)
(プロジェクト終了時まで決定していただければ、かまいません)

- プログラムの整備

- ビルド手順
 - ビルドコマンド
 - 依存外部ライブラリ

フルアプリ提供前の準備(続き)

- サンプル入力データの用意
 - 実行に必要な入力・設定ファイル
 - 実行時間、必要メモリサイズ、入出力ファイルサイズの情報
 - 2サイズのサンプルデータをお願いします
 - Small系
現在のマシンで1ノードで簡単に実行できる程度のサイズ(目安として、Intel Xeonマシンで1回の実行が数分)
 - Large系
できれば、京の1万ノード規模で実行可能なデータ
- 実行結果検証手順の用意
 - ひとまず簡便なものでかまわないので、検証プログラムを用意してください(のちのち、より厳密な検証プログラムへ置き換えも可能とします)

フルアプリ提供前の準備(続き)

- ドキュメントの整備(定形フォームを用意します)
 - プログラム説明
 - 計算内容
 - 使用アルゴリズム、スキーム
 - ビルド方法
 - 実行方法
 - 計算結果検証方法
 - 現在の実行規模(問題サイズ、実行時間、必要資源)
 - 2018～2020年頃の想定実行規模

ミニアプリ化方針

古典MD(分子動力学)アプリMARBLE, MODYLASを例に、
ミニアプリ化作業について説明

- 方針

- 生体高分子系のMDシミュレーション時のパフォーマンス特性を(なるべく)再現できること
- MDアプリとしての最低限の機能は残す
- 上記2条件の下で、なるべく小さなプログラムにする

古典MDアプリのミニアプリ化

- 機能を制限してプログラムを小型化
 - 水分子系(高分子なし)専用
 - NVE(ミクロカノニカル)アンサンブル専用
 - その他に削れる機能(エネルギー最小化、位置束縛、...)
- 特に、水分子系専用にすると
 - 削れるコード部分が多い
 - 非専門家でも入力データ作成が容易
 - 任意サイズのデータが作成可能 → 弱スケーリング測定が容易

古典MDアプリのミニアプリ化(続き)

コード削減後に残る部分(赤色がホットスポット)

- 毎タイムステップ
 - 短距離2体力(クーロン力の短距離成分+vdW力)計算
+隣接通信
 - グローバル通信(エネルギー集約など)
- 1~数ステップごと
 - クーロン力の長距離成分計算(PME, FMMなど)
- 数10ステップごと
 - ノード間の原子移動(隣接通信)
 - ペアリストなどのアップデート
- 指定ステップごと
 - ファイル出力(サイズは原子数オーダー、
100万原子なら40MB, 10億原子なら40GB)

古典MDアプリのミニアプリ化(続き)

棲み分け

- MODYLAS
 - 長距離力はFMMで計算
 - エクサスケールでも、全ノードを使って大規模計算
- MARBLE
 - 長距離力はPME(3次元FFT使用)で計算
 - エクサスケールでは、現在の計算規模のまま、アンサンブル計算
 - 必要メモリ量が少ないので、共有メモリマシン専用ミニアプリ化も検討中