

FDPS 議題

谷川

目 次

1	15/02/10	2
2	15/02/18	4
3	15/02/24	5
4	15/03/03	6
5	15/03/10	10
6	15/03/17	10
7	15/03/23	10
8	15/04/01	10
9	15/06/17	12
10	15/06/24	12

1 15/02/10

- ユーザーチュートリアル構成
 - － 目次作った
- 外部仕様書構成
 - － 目次作った
- テストスイート構成
 - － 項目作った
- エラー処理素案
 - － エラー処理の仕方 (わかりやすいメッセージを吐いて `MPLAbort`)
 - * 例外処理 (メッセージを吐いて、`throw` しっぱなし / こっちが `MPLabort`)
 - * 例外処理をやらないというオプション
 - － エラーコードいる? (やりなおせないのでは?)
 - － エラーメッセージの様式 (`PSERROR`: わかりやすいメッセージ、関数名、こまかいメッセージ)
 - － エラー処理必要なところ
 - * 入力ファイルない
 - * メモリ確保失敗
 - * 規定 (2G) より大きな配列
 - * ツリーのルートセル外に粒子
 - * ユーザーの不適切な初期設定 (コンパイルエラーにできないものとかの場合)
 - * 浮動小数点例外?
 - * セグメンテーションフォールト?
 - － `signal handler` 要検討
- API
 - － 相互作用リスト (`v1.0` ではスレッドセーフではないということを書く)
 - － 通信データクラスにブロードキャストがほしい
- 実装
 - － Paritlce Mesh どういうテストすべきか? (検討)
 - － 領域分割をユーザーが手で変更できるようにするか? (裏 API)
 - － ツリー内の位置情報を 64bit にするか (する)

- 相互作用ツリークラスの高レベル関数について
- メルセンヌツイスターの名前空間
- テストスイート API ごと
- 最優先 3/1 までにアルファ版をつくる
 - インストールガイド
 - * ソースの場所
 - * 取得の仕方
 - * 展開の仕方
 - * サンプルコードの場所
 - * サンプルコードのコンパイルの仕方
 - * サンプルコードの動かし方

2 15/02/18

- S64 の typedef が long だけど大丈夫？
- clang を除くではなく、openmp 対応のコンパイラ
- C++03 以降対応のコンパイラ
- MPI 対応のコンパイラ
- configure についてはできてないならチュートリアルにはかかない(今のところ configure は必要でない)
- サンプルコードの結果の見方
- MPI データ型の修正 (これはなし)
- 来週までに github に上げる

3 15/02/24

- FDPS 内の setPos の手前で shift を足すの危くない？
 - そもそも内部で桁落ちする
 - ユーザーが F32vec にしたときにも桁落ちする
- 詳細記述書 (fdps/document/document_description/doc.pdf) の動作概略的なのがチュートリアルか仕様書のどこかに必要な気がする。縮小版をチュートリアルに、フルを仕様書に

4 15/03/03

- ファイル

- このファイル: \$(FDPS)/sandbox/tanikawa_box/memo/memo_tanikawa.pdf
- 仕様書: \$(FDPS)/doc/doc_specs.pdf
- チュートリアル: \$(FDPS)/doc/doc_tutorial.pdf
- 論文下書き: \$(FDPS)/sc15/fdps_sc15.pdf

- ユーザーの問い合わせ先のメールアドレス (今は仮に ataru.tanikawa@riken.jp になっているが、専用のアドレスを作った方がよいのではないか) fdps-support@mail.jmlab.jp

- アルファ版について

- SC15 registration

- Rules
 - * Up to a maximum of 10 pages, not including references
- Title
- Author
- Requested Areas (第 2 希望まで)
 - * Algorithms
 - * Applications 第 2 希望
 - * Architecture and Networks
 - * Clouds & Distributed Computing
 - * Data Analytics, Visualization & Storage
 - * Performance
 - * Programming Systems 第 1 希望
 - * State of the Practice
 - * System Software
- Keywords (2 – 5 in total, 1 – 3 per area from the above)
 - * Applications: Computational earth and atmospheric sciences これ
 - * Applications: Computational astrophysics/astronomy, chemistry, fluid dynamics, mechanics and physics これ
 - * Performance: Analysis, modeling or simulation for performance, power and/or resilience
 - * Programming Systems: Parallel application frameworks これ
 - * Programming Systems: Parallel programming languages, libraries, models and notations

- * Programming Systems: Tools for parallel program development (e.g., debuggers and integrated development environments)
- Abstract
 - * No more than 150 words
 - * Paragraph breaks by a blank line in the text field
- Conference Presentations
- SC15 paper 色々
 - Categories and Subject Descriptors (<http://www.acm.org/about/class/ccs98-html>)
 - General Terms (Algorithms, Design, Documentation, Economics, Experimentation, Human Factors, Languages, Legal Aspects, Management, Measurement, Performance, Reliability, Security, Standardization, Theory, Verification) **いくつでも選べる (0 でもよい)**
 - Keywords (選択肢なし?)
- SC15 paper 構成
 - Introduction (なんで FDPS みたいなものが必要か?)
 - * 粒子シミュレーションは有用
 - ・ 構造が複雑な場合は格子法に比べて有利
 - ・ 様々な分野で利用
 - * 粒子シミュレーションの大規模化に対する要請
 - ・ より計算コストの大きい、複雑な現象を解きたい
 - ・ 大規模並列計算機が存在
 - * 粒子シミュレーションを大規模化する時の問題点
 - ・ プログラムが困難：ロードバランスのための動的領域分割、領域分割に合わせた粒子交換 ノード間通信の削減と最適化、キャッシュ利用効率の向上、SIMD ユニット利用効率の向上、アクセラレータ
 - ・ やることは同じなのに各グループが個別に開発：
 - * 共通のフレームワークを作ることによって問題を解決
 - ・ 利点：研究者がよりクリエイティブなことに専念できる
 - ・ 過去のアプローチとその問題点：**アプリケーションごとにはある (Gadget, PAM-CRASH etc.), 汎用?(portable parallel particle program)**
 - ・ 我々のアプローチとその利点：わかりやすいインターフェース、高い性能
 - * 論文の構成：実装、サンプルコード、性能、デモ、結論
 - Implementation
 - * 粒子シミュレーションのループの overview (我々による抽象化?)

- ・ 概念式

$$\frac{d\mathbf{u}_i}{dt} = \sum_j f(\mathbf{u}_i, \mathbf{u}_j) + \sum_j g(\mathbf{u}_i, \mathbf{v}_j) \quad (1)$$

f : 粒子-粒子相互作用, g : 粒子-超粒子 (相互?) 作用, \mathbf{u}_i : 粒子の物理量ベクトル, \mathbf{v}_j : 超粒子の物理量ベクトル

- ・ ロードバランスのための領域分割 (微分方程式の分担決め 1)
- ・ 領域分割に合わせた粒子交換 (微分方程式の分担決め 2)
- ・ 相互作用リストの作成: ローカルツリー、LET 交換、グローバルツリー、 i グループ作成、 i グループに対する相互作用リスト作成 (シグマ内の j 決め)
- ・ 相互作用計算 (シグマ内)
- ・ 時間積分, 組成進化, etc.
- * 考え方、FDPS とユーザーの役割分担
 - ・ FDPS: 並列化がからんだ複雑な処理 (領域分割、粒子交換、相互作用リストの作成)、これらをモジュール化する
 - ・ ユーザー: 特殊化 (粒子の定義、相互作用の定義など)、API の呼出、並列化のからまない単純な処理 (相互作用計算、時間積分、組成進化、外場など)
 - ・ スキーマティックな図 (領域クラス、粒子群クラス、相互作用ツリークラス、ユーザー)
- * ユーザーの仕事
 - ・ 特殊化: 粒子の定義 (粒子群クラスのテンプレート引数)、相互作用の定義 (相互作用クラスのテンプレート引数 (既存のものから選択も可); 関数オブジェクト; 境界条件)
 - ・ API の呼出 (C++)
 - ・ 時間積分とか
- * 使用言語: C++ **なくてもいい**
- * モジュール構成 **なくてもいい**
 - ・ 領域クラス (データ: 領域情報; API: decomposeDomainAll)
 - ・ 粒子群クラス (データ: 粒子情報; API: exchangeParticle; テンプレート引数: FP)
 - ・ 相互作用ツリークラス (データ: ツリー構造; API: calcForceAllAndWriteBack; テンプレート引数: SEARCH.MODE, Force, EPI, EPJ, MomentLocal, MomentGlobal, SPJ; 関数オブジェクト: calcForceEpEp, calcForceSpEp; マクロ: 座標系指定, MPI/OpenMP のオンオフ; 動的指定: 境界条件)
 - ・ 通信データクラス

- ・ Particle Mesh クラス
- Sample codes
 - * N-body こちらだけ; 記述は、粒子定義、相互作用定義、(main 関数)
 - * SPH
- Performance
 - * N-body (weak scaling, strong scaling)
 - * SPH+N-body, giant impact (weak scaling, strong scaling)
 - * MD 無しで
 - * 長時間やっても動くと言
 - * なんかきれいな絵 (スパイラル?)
- Discussion and Conclusion
 - * 補足
 - ・ 独立時間刻みへの対応
 - ・ SIMD, アクセラレータへの対応
 - ・ etc.
- SC15 paper 予定
 - 4/3 (UTC-12) (アブストラクト締切)
 - * Introduction, Implementation, Sample code 完成
 - * Performance の数字
 - 4/3 (UTC-12) – 4/17 (UTC-12)
 - * Performance, Demonstration, Conclusion 完成
- 役割責任者
 - Performance(N-body) : 岩澤
 - Performance(SPH)、きれいな図作成 : 細野
 - 本文、他細々したこと : 谷川
 - SIMD: 似鳥
 - 英語チェック: 村主

5 15/03/10

- 今日のファイル
 - このファイル: \$(FDPS)/sandbox/tanikawa_box/mem/memo_tanikawa.pdf
 - このファイル: \$(FDPS)/doc/doc.specs.pdf
- Phantom-GRAPe とかをライブラリとして加える？

6 15/03/17

- 今日のファイル
 - このファイル: \$(FDPS)/sandbox/tanikawa_box/memo/memo_tanikawa.pdf
 - SC15 メモ: \$(FDPS)/sc15/memo/memo_sc15.pdf
- 性能がでない場合の相談をユーザーサポートに加えるか？
- 相互作用計算用の関数オブジェクトの高速化についての相談をユーザーサポートに加えるか？

7 15/03/23

- 今日のファイル
 - このファイル: \$(FDPS)/sandbox/tanikawa_box/memo/tanikawa_memo.pdf
 - SC15 メモ: \$(FDPS)/sandbox/tanikawa_box/sc15_memo/sc15_memo.pdf
- SC15 メモ見る
 - 概念式確認 (sec.2)
 - サンプルコード相談 (sec.3)
 - 性能評価の役割確認とチケット発行 (sec.4)

8 15/04/01

- 今日のファイル
 - このファイル: \$(FDPS)/sandbox/tanikawa_box/memo/tanikawa_memo.pdf
 - SC15 メモ: \$(FDPS)/sandbox/tanikawa_box/sc15_memo/sc15_memo.pdf
- fdps-support にメールが来たときの対応 (済)

- 担当決め: 谷川
- 自動返信はあり。
- 解決法: 担当者はチケットのところに対応を書く。対応がOKかどうかは牧野さん。
- 返信先: メール作成は担当者。宛先は質問者への返信。Cc に fdps-support。差出人は fdps-support。
- 変更履歴について (済)
 - ソースコードの変更は `src` のしたの CHANGELOG へ
- リリースのバージョンについて
 - 微修正として ver. 1.0.1
 - 微修正しましたと README.md に書く。文書の変更履歴はここ、ソースの変更履歴はここ。
 - 来週くらいにリリース
- SC15 Technical paper
 - Title: FDPS: A Novel Framework for Developing High-Performance Particle Simulation Codes for Distributed-Memory Systems
 - Author Information: SC15 メモへ
 - Requested Areas: 第1希望 Programming Systems, 第2希望 Applications
 - Keywords (2 – 5):
 - * Algorithm: Numerical methods, linear and nonlinear systems
 - * Applications: Computational earth and atmospheric sciences
 - * Applicatoins: Computational astrophysics/astronomy, chemistry, fluid dynamics, mechanics and physics
 - * Programming Systems: Parallel application frameworks
 - * Programming System: Tools for parallel program development (e.g., debuggers and integrated development environments)
 - Abstract: SC15 メモへ
 - Conference Presentations: Yes
- 論文のサンプルコードまでを来週の水曜までに

9 15/06/17

- 仕様書の節 7 について
 - － getPos は位置を返すことが仕様
 - － 前提、備考は実装例の説明 (実装例は付録もしくは別の節)

10 15/06/24

- Appendix に実装例のサンプルを加える
- 実習の手引 (レジユメ的なもの) (– 7/7)
- FDPS 概要のスライド (– 7/7)
- 実際のテスト (– 7/7)
- PS::SEARCH_MODE に
PS::SEARCH_MODE_LONG_SCATTER,
PS::SEARCH_MODE_LONG_CUTOFF_SCATTER,
PS::SEARCH_MODE_LONG_SYMMETRY,
PS::SEARCH_MODE_LONG_CUTOFF_SYMMETRY を追加