

FDPS 講習会 イントロダクション

牧野淳一郎

神戸大学理学研究科惑星学専攻
および 理化学研究所 計算科学研究センター
兼 粒子系シミュレータ開発チーム

2021/9/9 FDPS 講習会

今日の予定

- 共通部分

- 13:00-13:10 1. イントロダクション (牧野)
- 13:10-13:30 2. FDPS の概要説明 (岩澤)
- 13:30-13:50 3. FDPS の API と内部構造 (岩澤)

- C++パート Zoom ブレイクアウト 1

- 14:00-14:20 4. C++について (細野)
- 14:20-14:40 5. サンプルコード C++ (細野)

- Fortranパート Zoom ブレイクアウト 1

- 14:00-14:20 4. Fortranについて (牧野)
- 14:20-14:40 5. サンプルコード Fortran (牧野)

今日の予定(続き)

- 共通

- 14:40-14:50 休憩・ZOOMつなぎ直し
- 14:50-15:20 PIKG およびFDPSの新機能について(牧野)
- 15:20-16:55 6. 実習・質問(必要に応じてブレイクアウト利用)
- 16:55-17:00 7. 結び(牧野)

イントロダクション残り： FDPS は何をするか？

というよりむしろ、何がしたくない(やったことがない)人のためのものか：

- MPI でプログラム書いたことない
- キャッシュ再利用のためのわけのわからないループ分割とかしたくない
- 通信量減らすためにわけのわからない最適化とかするのも勘弁して欲しい
- SIMD 命令ができるようにコードをいじりまわすとか無理
- 機械毎にどういう最適化すればいいか全然違うとか、それ以前に言語から違うとかはもういやだ

そうはいっても—ではどうするか？

昔からある考え方はこんな感じ？

- 並列化コンパイラになんとかしてもらおう
- 共有メモリハードウェアになんとかしてもらおう
- 並列言語とコンパイラの組み合わせになんとかしてもらおう

しかし……

- 若い人はそういう考え方があったことも既に知らないような気がする。「スパコンとはそういうものだ」みたいな。
- つまり、こういうアプローチはほぼ死滅した。
- 理由は簡単：性能がでない。安価なハードウェアで高い性能がでるものがプログラミングが大変でも長期的には生き残る。

じゃあ本当のところどうするか？

1. 人生そういうものだとして諦めて MPI でプログラム書いて最適化もする
難点: 普通の人の場合性能がでない。難しいことをしようとすると無限に時間がかかって人生が終わってしまう。
 2. 他人 (学生、ポスドク、外注、ベンダ等) に MPI でプログラム書かせて最適化もさせる
難点: 他人が普通の人の場合、やはり性能でない。無限に人と時間とお金がかかる。あとでいじるにも無限に人と時間とお金がかかる。
- どちらも今一つというか今百くらいである。
 - もちろん、「普通でない人」を確保できればなんとかなるがこれは希少資源である。

原理的には、「普通でない人」を有効利用すればいい？

どうやって有効利用するか？

色々な考え方がありえるが、我々の考え方:

- 「普通でない人」がやった方法を一般化して、色々な問題に適用する。
- 例えば「粒子系一般」という程度
- DRY (Don't Repeat Yourself) の原則の徹底
- 理研 CCS 粒子系シミュレータ開発チームで詳細仕様策定及び実装

もうちょっと具体的には？

色々な粒子系計算

- 重力多体系
- 分子動力学
- 粒子法による流体 (SPH、MPS、MLS、その他)
- 構造解析等のメッシュフリー法

計算のほとんどは近傍粒子との相互作用 (遠距離力: Tree, FMM, PME その他)

もうちょっと具体的には？

なので、粒子と相互作用の定義を与えると

- 領域分割 (ロードバランスも考慮した)
- 粒子の移動
- ツリー法での相互作用の計算 (そのために必要な通信も)
- 相互作用計算の SIMD 化や GPGPU を利用するコード生成 (Ver 6.0 新機能)
- 円盤系向けの効率よい並列化コード生成 (Ver 7.0 新機能)

を高い効率 (実行効率・並列化効率) でやってくれるプログラムを「自動生成」できればいい。時間積分とかは自分で書く。

(独立時間刻み? P³T スキームで)

ということで、詳しくはこれからの説明をどうぞ。