

生体シミュレーションサービス Flint K3 における高性能計算に向けて

永井 美穂† 置田 真生‡ 安部 武志§ 浅井 義之§ 北野 宏明§ 萩原 兼一‡

†大阪大学 基礎工学部

‡大阪大学 大学院情報科学研究科

§沖縄科学技術大学院大学 オープンバイオロジーユニット

1 はじめに

Flint K3[1] は、クラウド環境上で生体シミュレータ Flint[2] を実行できる SaaS[3] である。主な利用者は生体工学分野の研究者であり、生体モデル (BM) の開発に利用する。BM の開発においては、BM の変更とシミュレーション (SIM) 結果の確認を繰り返すため、高速な SIM が要求される。我々は、BM を並列実行可能なシミュレーションコード (SC) へと自動変換することで高性能計算を行うトランスレータ方式 [4] を開発し、Flint K3 に導入した。

効率的な並列 SIM のためには、並列に動作するプロセス数 (並列度) を BM の規模や構造に応じて適切に選択する必要がある。最適値を越える並列度を選択した場合、過剰に資源を消費するにも関わらず処理時間は増大する。Flint K3 はマルチテナント方式 [3] であるため、過剰な資源消費は他の利用者に不利益を与える。しかし、並列処理に不案内な利用者にとって、最適な並列度の選択は容易ではない。

そこで、我々は Flint K3 への並列度推薦機能の導入を検討している。同一環境で過去に実行された BM と処理時間の情報をもとに、任意の BM に対して最適な並列度を推定しユーザに提示する。本稿では、処理全体のうち BM から SC への変換処理 (2 節参照) を対象に、最適な並列度の推定について報告する。

2 Flint K3 におけるシミュレーションの流れ

Flint K3 は以下の手順で SIM を実行する。

- (P0) 対象 BM と必要なパラメータを利用者から受取
- (P1) BM を SC へ変換
- (P2) コンパイルした SC を実行
- (P3) 実行結果をチャート画像に変換し利用者に提示

SIM のパラメータは、並列度 p および SIM の期間 l を含む。(P1) および (P2) は自動並列化されており、 p 個のプロセスで並列動作する。

Towards High Performance Computing in Cloud-based Biophysical Simulation Service Flint K3

†Miho NAGAI ‡Masao OKITA §Takeshi ABE §Yoshiyuki ASAI §Hiroaki KITANO ‡Kenichi HAGIHARA

†School of Engineering Science, Osaka University

‡Graduate school of Information Science, Osaka University

§Open Biology Unit, OIST Graduate University

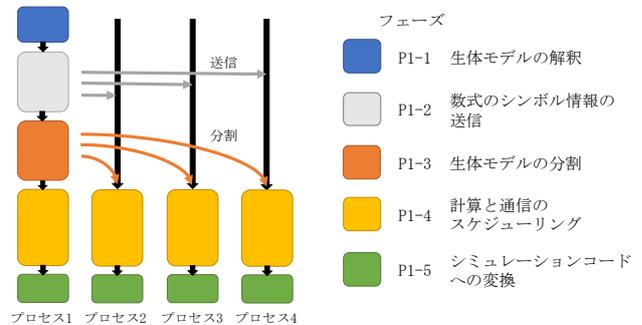


図 1: (P1) のフェーズと並列処理の流れ

全処理時間の大半を (P1) および (P2) が占める。例えば大規模な心筋細胞モデル (数式数: 約 4 千万) の場合、(P1) に約 1 時間を要し、(P2) では SIM の期間 1 秒毎の計算に約 1.7 時間を要する (AMD Opteron 2.0 GHz 48 コアを使用)。

BM 開発の初期段階においては、(P1) が性能ボトルネックとなる。この理由は、BM の誤りを発見する可能性が高く、比較的小さな l を用いて SIM を繰り返すためである。一方、後期段階においては長大な SIM の結果を確認するため、(P2) が性能ボトルネックとなる。

3 最適な並列度の推定

我々はまず BM 開発の初期段階を想定し、(P1) の並列実行における最適な並列度の推定に取り組んだ。推定手法は、(P1) の処理時間 T_s の予測モデル (PM) の作成、および予測モデルを用いた並列度の決定に分けられる。紙面の都合上、本稿では前者について述べる。

PM の作成手順は以下の通りである。経験的に、 T_s は BM に含まれる数式の数 n および並列度 p とそれぞれ独立に相関がある。そこで、(P1) の処理を主要な 5 つのフェーズに分解し、まずアルゴリズムを解析してフェーズごとに n および p を変数とするモデル関数を作成する。次に、同一環境で過去に実行した複数 SIM の T_s の集合を学習データとし、最小二乗法を用いて係数を求める。

(P1) の主要なフェーズと並列処理の流れを図 1 に示す。(P1-1) および (P1-3) は逐次処理を行うため、処理時間は p に依存しない。提案手法では両者を統合して扱う。モデルの分割に用いたアルゴリズムの計算量

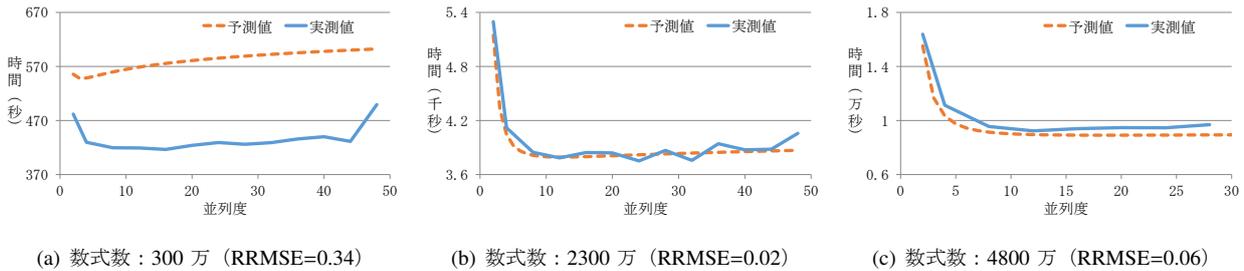


図 2: (P1) の処理時間 (予測値と実測値)

は $O(n^2)$ であるため、両フェーズの合計処理時間のモデル関数 $T_{1,3}(n)$ を次式とする。

$$T_{1,3}(n) = c_1 n^2 + c_2 n + c_3$$

(P1-2) では、あるプロセスから残りのプロセスに全数式のシンボル情報を送信する。二分木状に通信するため、その処理時間のモデル関数 $T_2(n, p)$ を次式とする。

$$T_2(n, p) = (c_4 n + c_5) \log_2 p$$

(P1-4) で用いたアルゴリズムの計算量は $O(n^2)$ であるが、数式は全プロセスに均等に分配されているため、(P1-4) の処理時間のモデル関数 $T_4(n, p)$ を次式とする。

$$T_4(n, p) = c_6 \left(\frac{n}{p}\right)^2 + c_7 \frac{n}{p} + c_8$$

(P1-5) は数式をプログラム行に変換する処理を p 並列で実行するため、その処理時間のモデル関数 $T_5(n, p)$ を次式とする。

$$T_5(n, p) = \frac{c_9 n + c_{10}}{p}$$

全処理時間の予測値 $T(n, p)$ は、これら 4 つのモデル関数の総和となる。

4 実験

PM から求めた T_s の精度を評価し、並列度自動推薦の有用性について検討する。本実験における最適並列度の推定方法として、 T_s の削減を優先し、 T_s の予測値が最小になる並列度 p_p を推薦するとした。実験環境は、CPU が AMD Opteron 6174 2.2 GHz 12-core \times 4、主記憶が DDR3-SDRAM ECC 256 GB である。PM の係数は、同一環境で実行した 8 種類の BM (数式数 : 15 万 ~ 4800 万) を学習データとして求める。 T_s を予測する BM として、3 種類の心筋細胞モデル (数式数 : (a)300 万, (b)2300 万, (c)4800 万) を選んだ。いずれも、学習データに類似構造の BM を含むため、 T_s の精度はより高まると考える。

T_s の予測値と実測値、相対平均二乗誤差 (RRMSE) を図 2 に示す。(a) の誤差は、(b) および (c) の誤差と比

表 1: 推薦並列度で実行した場合の T_s の増加率 R

	数式数	推薦 p_p	最速 p_m	$R = T_m(p_p)/T_m(p_m)$
(a)	300 万	4	16	1.031
(b)	2300 万	12	24	1.008
(c)	4800 万	20	12	1.025

較して大きい。この理由は、BM の構造に依存する処理時間の変動を PM が無視するためと推測する。

表 1 に推薦並列度を用いた場合の T_s の増加率 R を示す。ここで、 T_s の予測値および実測値が最小となる p をそれぞれ p_p および p_m と表し、並列度 p における T_s の実測値を $T_m(p)$ と表す。(a) および (b) について、 p_m より資源消費の小さい p を、性能低下がそれぞれ 3%、0.8% 以内に収まる範囲で推薦している。一方で、(c) については、 p_m より資源消費の大きい p を推薦しており、望ましくない。処理時間の増大を一定以内に抑えるために必要十分な p の推薦手法は、今後の課題である。

謝辞

本研究の一部は科学研究費補助金 (26730035, 25136711) および文部科学省「医・工・情報連携によるハイブリッド医工学産学連携拠点整備事業」の支援による。

参考文献

- [1] Flint K3, <https://flintk3.unit.oist.jp/> (2013). Accessed Jan. 09, 2015.
- [2] Flint, <http://physiodesigner.org/simulation/flint/> (2012). Accessed Jan. 09, 2015.
- [3] Mell, P. et al.: SP 800-145. The NIST Definition of Cloud Computing, Technical report (2011).
- [4] Okuyama, T. et al.: Accelerating ODE-based Simulation of General and Heterogeneous Biophysical Models using a GPU, *IEEE Trans. Parallel and Distributed Systems*, Vol. 25, No. 8, pp. 1966–1975 (2014).