Erlang VM を用いた Rust の並列処理ライブラリの開発

1220310　岡本 怜士　【 ソフトウェア検証・解析学研究室 】

1　はじめに

Erlang VM を用いたシステムは CPU バウンドな処理の遅さが目立つ場合がある．それを解決するために，Erlang VM の NIF(Native Implemented Function) という機能が利用されている．事例としては，この機能に対する性能評価を行った研究 [1] や，有名ネットワークサービスである Discord が，Go 言語を用いた開発から Rust による NIF を用いた開発へ移行した例 [2] などがある．これらは Erlang VM の並列性能とネイティブコードで実行される Rust を組み合わせることで，高速性とメモリ安全性を両立している．

しかし，NIF を用いた開発にはいくつか問題点がある．まず，Rust は静的型付け言語だが，Erlang は動的型付け言語であるため，NIF 呼び出しの際，引数の型の整合性はプログラマに責任がある．また，Rust のデータ型やトレイトに対するメソッドは直接 NIF としてエクスポートできないため，これらを用いる場合は適切なラッパ関数が必要となる．使用するデータ型が増えるたびに，ラッパ関数を作成するのは手間がかかる．

本研究では，これらの問題を解決するためのライブラリを開発した．

2　ライブラリ概要

提案するライブラリは，NIF を記述する言語 (Rust) のみを用いて Erlang VM を用いたシステム全体を記述できるようにするものである．このようにすることで，型の整合性に関する責任をコンパイラに任せることが可能となる上に，メソッド呼び出しなどに対する記述の制約もなくなる．これを実現するためには，あるエントリポイントから Erlang VM 上で実行を開始するような仕組みが必要となる．また，NIF の API には軽量プロセスを新しく作成するものが含まれていないため，これを実現する仕組みも必要となる．

前者は，Rust 標準のビルドツールである Cargo のサブコマンドを新たに実装することで実現した．

後者の実現は，条件として NIF から Erlang VM に値を渡す手段は NIF の返り値かメッセージを送ることのみであり，さらに，渡される値の型は NIF のコンパイル時に宣言されている必要があるということを踏まえる必要がある．軽量プロセスを作成するたびに NIF からリターンするのは現実的ではないため，本ライブラリではメッセージを送る方法を採用した．メッセージには新しく作成される軽量プロセスが評価する関数と実引数を含める．メッセージに含まれる値の型を事前に宣言する必要があるという問題については，マクロによって Rust の列挙型を自動生成するという手段で解決した．

3　性能評価実験

本ライブラリによって並列性能が悪化しないかを評価するために，行列積を並列に求めるプログラムと，ロジスティック写像 $x_{n+1} = 4.0x_n(1 - x_n), n = 10^9$ を求めるプロセスをいくつか生成するプログラムをベンチマークとして作成し，実行時間を調べた．それぞれの結果を下のグラフに示す．

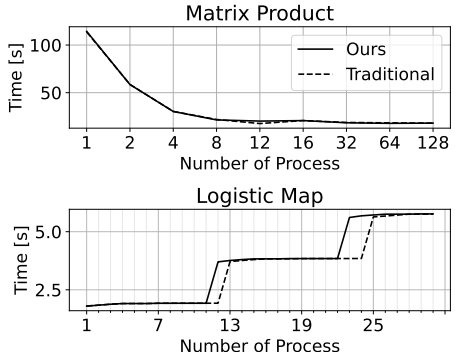

図 1　ベンチマーク評価結果

行列積ベンチマークでは性能の差はほとんど見られない．一方ロジスティック写像ベンチマークでは，提案手法のほうが同じタスク量のプロセスが増えるにつれて並列性能が悪化しやすくなっているのがわかる．

4　考察

本ライブラリを用いることで，プログラマが扱わなければならないプログラミング言語が一つ少なくなっているため，学習コストの面で負担軽減に貢献できていると考えられる．ただし，提案手法と従来の手法を比較すると，どちらのベンチマークもコード量に大きな差は生じなかった．これは，Erlang と Rust のコードを比べたとき，変数の型や参照の扱いなど Rust では明示的に書かなければならないことが Erlang では省略できることが影響していると考えられる．

5　まとめ

Erlang，および NIF を用いた開発において，型の整合性などの問題点を解決するために，Rust を用いて Erlang VM の並列処理を表現するライブラリを開発した．

参考文献

[1] 高橋, 上野, 山崎,「関数型言語 Elixir の IoT システムへの導入に向けた基礎評価」, 情処研報 Vol.2018-EMB-48 No.5, 2018/6/29

[2] Why Discord is switching from Go to Rust, [online] https://discord.com/blog/why-discord-is-switching-from-go-to-rust (参照 2022-02-02)