

プログラミング言語 Erlang の評価

理工学部 数理情報学科
T070076 萩原 貴章
指導教員 佐野 彰

概要

本論文ではプログラムの並列化を容易に行うことができるプログラミング言語 Erlang について、その特徴や性能を C 言語と比較、実験することで Erlang のプログラミング言語として、同じ処理を並列化した場合のパフォーマンスの差がどれほどのものであるかを調べ、また並列化の際にはどのような条件が一番最適であるのかという観点からの評価を行なっていく。

Erlang は変数への代入が1度しかできないことが大きな特徴である。これは Erlang が純粋関数型言語で、一度束縛された変数が定数として扱われるためである。このように、C 言語とは大きく異なる Erlang のシングルコアで問題を扱った場合と並列化して問題を扱った場合のそれぞれのパフォーマンスが C 言語とどれほどの差を生むのか、またそれぞれのパフォーマンスが仕事の内容によってどのように変化をするのかを調べるためにクイックソートとマージソートをそれぞれの言語でなるべく同じ手順での処理、そして同一の行列を扱っていく条件で、4コアとハイパースレッディング機能によって各コア2スレッドまで並列化を行える CPU を搭載した、ターボブースト機能が搭載されていない純粋な CPU パワーを扱うことができる環境で実験を行なった。10個の異なるシードから作成したデータセットを読み込む処理については、それぞれの言語の読み込む時間が均一であるという証明がないことと、読み込みそのものに揺らぎが生じやすいことから、計測はアルゴリズムが最初に呼び出された段階からスタートするようにしている。

結果としては、まず実行方法によってパフォーマンスが劇的に変わるということが分かった。インタプリタ方式で実行した場合とコンパイルして実行した場合とでは10倍近くの差が開く結果となった。また、コンパイルした場合のソート時間と C 言語のソート時間を見比べてみると、Erlang がシングルコアソートではパフォーマンスが C に少し劣っているが並列化した場合、C 言語を上回る結果となった。しかし、並列化した Erlang は C 言語に比べてマージソートでは三倍近くのパフォーマンスを引き出していたが、クイックソートでは C 言語とあまり変わらなかった。ソート別に各ソート時間を見た場合、並列化した時のみマージソートのほうが速いという結果に変わっていた。

この原因を調べるために分割するプロセス数を深さ N を変化させて検証してみた。その結果、マージソートではシングルコアソートを行うプロセス数が計算機の同時処理可能な4コア×2スレッド=8プロセスと同じであった時、最大のパフォーマンスを引き出すことができた。しかし、クイックソートではプロセス分割の深さを大きくしていくほどパフォーマンスが向上することが分かった。ここから推測できることは仕事の振り分け方によって CPU のパワーを最大に引き出せる条件が変化するということだ。

Erlang は従来の言語では難関であった並列化などの処理を簡単に行えるようにするというコンセプトから設計されている。この事からも、元々開発、管理が行いやすい言語であるということが分かる。また並列化を行う際に CPU のパワーを最大限まで引き出すには、マージソートのように仕事を均等に分ける場合には深さ N のとき、計算機の同時処理可能なプロセス数と同じだけプロセス数を生成してやれば良い。逆にクイックソートのように仕事の振り分けが不均等であるアルゴリズムの場合は最低でも同時処理可能なプロセス数以上のプロセスを生成することで、CPU のパワーを引き出すことができるようになる。