
発表概要

並列スクリプト言語 (Perl)+の実装と設計

外崎 由里子[†] 大野 和彦[†] 中島 浩[†]

PC クラスタやデュアル CPU マシンなどの普及により近い将来、研究者だけでなく一般のエンドユーザが並列環境を利用できるのが当たり前になると予想される。逐次計算機上では、C などの高性能なコンパイラ型言語処理系とともに、Perl などの手軽なスクリプト言語処理系が使用されてきた。並列計算機上では前者に対して HPC++ などが開発されているが、後者に相当するものの研究は進んでいない。そこで我々は、エンドユーザが容易に並列計算機資源を有効利用できる環境を実現するため、並列性を簡易に記述できるように Perl を拡張した、並列スクリプト言語 (Perl)+ の設計・開発を行っている。(Perl)+ では指定した計算機を並列環境に追加し、RPC により任意のサブルーチンを実行することができる。その返り値は遅延評価されるため、ユーザは複数のサブルーチン呼び出しを容易に並列タスクとして実行できる。また、並列タスク間では通信用に擬似的なファイルストリームを開くことができ、Perl の入出力関数を使ってデータを送受信できる。これらの機能は C で実装し、Perl と C を組み合わせるためのツールである XS を使って Perl モジュール Perlplus.pm を構築している。このため、ユーザは本モジュールを取り込むだけで (Perl)+ の機能を利用できる。本モジュールは起動時に各ホスト上に Perl プロセスを生成し、PVM により RPC や通信を実現している。また、各プロセス上で実行スレッド/受信スレッドを生成することで、RPC の実行と並行して受信処理を行えるようにしている。

Design and Implementation of a Parallel Script Language (Perl)+

YURIKO TONOSAKI,[†] KAZUHIKO OHNO[†] and HIROSHI NAKASHIMA[†]

The spread of PC clusters and multi-CPU machines makes multiprocessors environment available not only for the researchers but also for the end users. On the uniprocessor machines, we can use both efficient languages such as C and simple script languages such as Perl. On the multiprocessors, the languages of the former type such as HPC++ have been developed. However, the latter type is not researched enough. Thus, we designed and implemented a parallel script language named (Perl)+ as an extension of Perl. (Perl)+ supports parallel task generation using RPC. Since the return value of a subroutine is lazily evaluated, the subroutine is executed in parallel to its caller. In addition to the communication through input arguments and return value, a user may open quasi file streams for the communication between parallel subroutines. Through this stream, any type of Perl data may be transferred using input/output functions of Perl. We implemented these functions in C. The user-interface is built as a Perl module Perlplus.pm, using XS for the linkage of C and Perl. This module adds specified hosts to the PVM virtual machines and creates Perl processes. We also introduced multi-threads for concurrent execution of user's Perl code and PVM message receiving.

(平成 13 年 7 月 27 日発表)

[†] 豊橋技術科学大学情報工学系

Department of Information and Computer Sciences,
Toyohashi University of Technology