Runtime Correctness Check for Co-working Parallel Programs

Miwako Tsuji
Hitoshi Murai
Mitsuhisa Sato
miwako.tsuji,h-murai,msato@riken.jp
RIKEN R-CCS

Thomas Dufaud
Nahid Emad
thomas.dufaud@uvsq.fr
nahid.emad@uvsq.fr
Li-Parad, UVSQ

Joachim Protze
Christian Terboven
Matthias S. Müller
protze,terboven,mueller@itc.rwth-aachen.de
RWTH Aachen University

Taisuke Boku
taisuke@ccs.tsukuba.ac.jp
CCS, University of Tsukuba

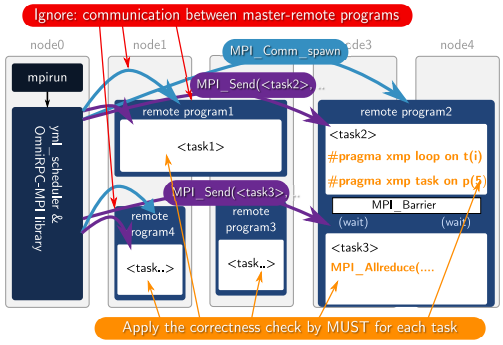
Serge G. Petiton
serge.petiton@univ-lille.fr
Université de Lille

1 INTRODUCTION

The multi SPMD (mSPMD) programming model, where several SPMD programs work together conducted by a workflow programming, had been proposed to realize a scalable programming environment [2]. To help to debug parallel programs in the mSPMD programming model, we have extended the mSPMD programming environment to support MUST [1], which is a runtime tool that provides a scalable solution for efficient runtime MPI error checking. The MUST has supported not only MPI but also a directive based PGAS language called XcalableMP (XMP).

In this paper, we describe the implementation and performance evaluation of the MUST in the mSPMD programming environment.

2 CORRECTNESS CHECK FOR THE mSPMD PROGRAMMING MODEL

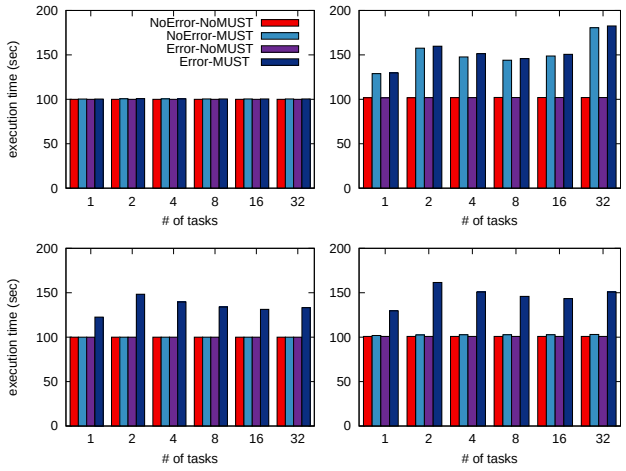


The above figure shows the overview of the mSPMD programming model and the correctness check for the mSPMD programming model. In the mSPMD programming model, our workflow scheduler invokes remote programs and requests a certain task for each of the remote program. There are 2 kinds of communication activities: (1) communication for workflow control (written in white/black font in the figure), and (2) communication within a task (written in orange font in the figure). In this work, we focus on the latter and only check errors in user-defined communication functions in a task.

MPI_ functions are wrapper of PMPI_ functions which actually perform communication. The MUST overwrites the MPI_ functions to analyze their correctness, then calls the corresponding PMPI_ functions. In our implementation of the MUST for mSPMD program, we replace MPI functions with PMPI functions for the workflow control to avoid to be checked. On the other hand, user-defined MPI functions and XMP directives are checked by MUST.

3 EXPERIMENTS AND SUMMARY

We perform some experiments on Oakforest-PACS supercomputer to investigate the overhead of the MUST in the mSPMD programming model. Each of workflow applications includes between 1 and 32 tasks, and a task uses 30 processes. The bottom figures shows the execution time of tasks of 100 × (Allreduce+sleep(1sec)), 10000 × (Allreduce+sleep(0.01sec)), 100 × (SendRecv+sleep(1sec)), 10000 × (SendRecv+sleep(0.01sec)) (From top-left to bottom-right). Tasks including invalid (but non-critical) MPI arguments are compared with those without any error.



Our experiments show that the MUST correctness checking in the mSPMD programming environment can detect syntax errors. The overhead to apply the MUST library is caused by (1) high frequently use of the collective communications and (2) syntax errors within a dependent communication pair (or triplet, quadruplet, ...) such as send/recv. Otherwise, the overhead is very small.

REFERENCES

[1] The MUST Project. https://www.itc.rwth-aachen.de/must.
[2] Miwako Tsuji, Mitsuhisa Sato, Maxime Hugues, and Serge Petiton. Multiple-spmd programming environment based on PGAS and workflow toward post-petascale computing. In *Proceedings of the 2013 International Conference on Parallel Processing (ICPP-2013)*, pages 480–485. IEEE, 2013.