Teaching Parallel Programming Using an Interactive Parallelization Tool

Ritu Arora
Texas Advanced Computing Center
University of Texas at Austin
Austin, USA
rauta@tacc.utexas.edu

Lars Koesterke
Texas Advanced Computing Center
University of Texas at Austin
Austin, USA
lars@tacc.utexas.edu

***Abstract*— We are iteratively developing an Interactive Parallelization Tool (IPT) for lowering the adoption barriers to parallel programming (MPI, OpenMP, CUDA). While IPT is still under active development, we have already put it to use in our trainings on MPI and OpenMP. An overview of IPT and some early results from its evaluation will be presented in the poster.**

Keywords-interactive parallelization tool; MPI; OpenMP; CUDA; cloud

I. Introduction

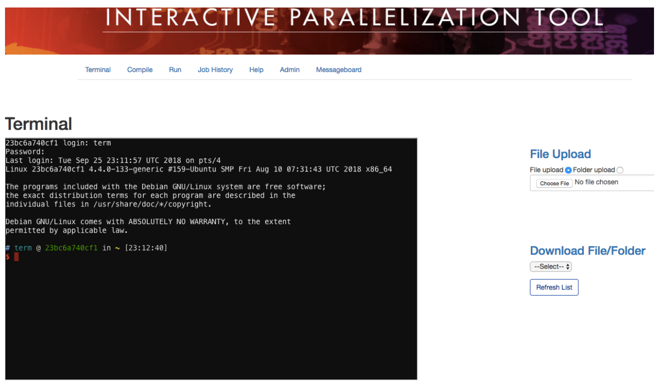
Teaching and learning parallel programming can be a challenging task due to several factors, such as, the low-level nature of the popular parallel programming models and the difficulty in troubleshooting errors like race-conditions and deadlocks. Additionally, even before students can test the performance and scalability of parallel programs on production systems, it becomes important that they spend time in familiarizing themselves with the user environment of those systems. To surmount such challenges, we have been developing a high-productivity Interactive Parallelization Tool (IPT) [1] and have deployed it in the cloud, so that, those interested in using it for generating parallel programs and testing those programs on the computational resources of the national CyberInfrastructure (CI), can do so from the convenience of their web browser [2]. A screenshot of IPT in action, when accessed through the web-browser, is shown in Figure 1.

IPT can assist in parallelizing specific types of serial C/C++ applications (e.g., those having regular meshes, and nested for-loops) using MPI, OpenMP, and CUDA. It works in the command-line mode and accepts the existing serial programs as input. An IPT user interactively specifies the parallel programming model and the hotspots for parallelization in their code. On the basis of the user-specifications and its in-built heuristics, IPT analyzes the input serial code and generates its parallel version. The parallel code generated using IPT is readable and is adequately commented to provide insights into the changes made to the serial code for parallelization purposes.

IPT can help instructors in teaching parallel programming through demonstration and in blended learning environments. The instructors can focus on teaching parallel programming concepts (such as, data distribution and loop-dependency analysis) and can then directly engage the students in doing parallel programming exercises without spending too much time in explaining the low-level details related to the syntax of the different parallel programming models, or the user environment of the production systems. In this manner, they can free up the face-to-face time in their classes for developing strong algorithmic foundations for developing parallel programs and increasing the number of hands-on exercises that can be done in the class.

IPT also supports parallelization of I/O done from MPI and OpenMP programs. Hence, it can be used for teaching the process of writing code that can optimally do the communication, computation, and I/O.

Some of the test cases and the steps to parallelize them using IPT have been made available to the community through a GitHub repository [3]. We have also started covering the topic of code correctness in our trainings on parallel programming using IPT and have prepared quizzes that can help students assess their knowledge on parallel programming [4]. Doing this is especially important, as there are some applications (e.g., irregular meshes) that cannot yet be parallelized using IPT. Therefore, teaching the pitfalls or common errors observed in parallel programs contributes towards making the IPT users feel self-sufficient in their manual parallelization efforts.



```
23bc6a740cf1 login: term
Password:
Last login: Tue Sep 25 23:11:57 UTC 2018 on pts/4
Linux 23bc6a740cf1 4.4.0-133-generic #159-Ubuntu SMP Fri Aug 10 07:31:43 UTC 2018 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

# term @ 23bc6a740cf1 in ~ [23:12:40]
$
```

Figure 1. Screenshot of IPT in Action.

II. IPT Evaluation

We have recently started using IPT for teaching parallel programming to students from a diverse range of disciplines. With IPT, we are able to quickly demonstrate the (1) differences in the structure and performance of the parallel code generated for different parallel programming paradigms, and (2) the impact of selecting different parallelization strategies.

Various students who have seen the IPT demo or have participated in the IPT training sessions are interested in using it for learning parallel programming. Some data related the evaluation of IPT is shown in Figure 2. As can be noticed from the data in Figure 2, 84.2% of the students in a class in which IPT was demoed, would like to continue using it for learning parallel programming, and 15.8% students were not sure about their decision. All the students found IPT to be useful. Across all the surveys conducted during IPT trainings, more than 80% people were interested in continuing to use IPT for their parallelization efforts. About 50% of the participants in some of the IPT trainings/tutorials belonged to the underrepresented groups as identified by the National Science Foundation.

Through the feedback collected from the community, we have observed that there is a demand for extending IPT for supporting the parallelization of serial programs written in Fortran as well. As a next step, we will be extending IPT to support the parallelization of Fortran applications as well.

ACKNOWLEDGMENT

The work presented in this paper was made possible through National Science Foundation (NSF) award number 1642396. We are very grateful to NSF for the same.

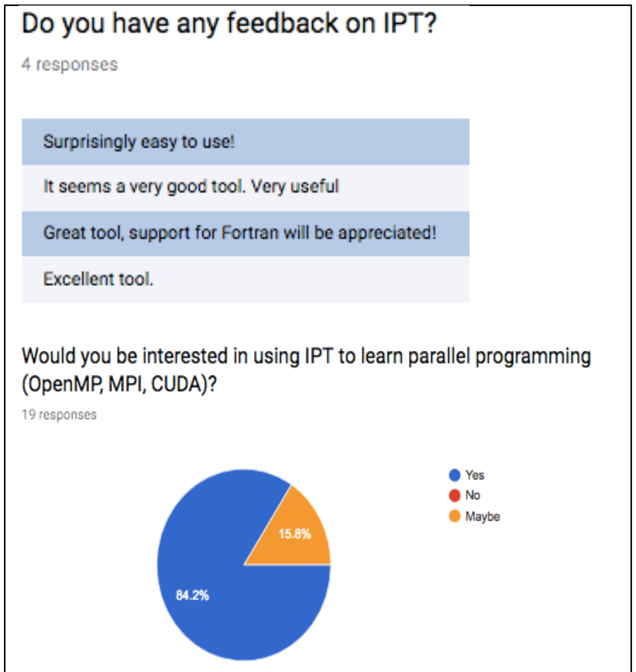
REFERENCES

[1] R. Arora, J. Olaya, and M. Gupta, "A Tool for Interactive Parallelization," In Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment (XSEDE '14). ACM, New York, NY, USA, Article 51, 8 pages. 2014. DOI: https://doi.org/10.1145/2616498.2616558

[2] IPT Gateway. Website accessed on November 19, 2018: https://ipt.tacc.cloud/

[3] IPT GitHub repository. Website accessed on November 19, 2018: https://github.com/ritua2/IPT

[4] T. Nguyen Ba, and R. Arora, "Towards Developing a Repository of Logical Errors Observed in Parallel Code for Teaching Code Correctness," In Proceddings of the EduHPC 2018 workshop @ SC18. https://grid.cs.gsu.edu/~tcpp/curriculum/sites/default/files/TrungNguyen_0.pdf



Do you have any feedback on IPT?

4 responses

Surprisingly easy to use!

It seems a very good tool. Very useful

Great tool, support for Fortran will be appreciated!

Excellent tool.

Would you be interested in using IPT to learn parallel programming (OpenMP, MPI, CUDA)?

19 responses

Figure 2. Student feedback on IPT.