

HPC for novices through a course in parallel programming

Vaishali Shah

*Interdisciplinary School of Scientific Computing (ISSC)
Savitribai Phule Pune University
Pune, India
Email: vaishali@unipune.ac.in*

Venkatesh Sheno, Sandeep Joshi

*Centre for Development of Advanced Computing (C-DAC)
Savitribai Phule Pune University Campus
Pune, India
Email: {venkateshs,skjoshi}@cdac.in*

Abstract—We describe our experience with HPC training for masters students of M. Sc. (Scientific Computing) through a one semester long course in parallel computing at ISSC and internships for students at C-DAC, Pune.

Keywords-High Performance Computing; HPC; parallel programming; scientific computing; training course

I. INTRODUCTION

The Interdisciplinary School of Scientific Computing (ISSC), S. P. Pune University offers a two year M.Sc (Scientific Computing) [1] for the students with B. Sc. (CS) / B. C. A. background. This programme has a class strength of thirty students. The students joining this programme have a limited knowledge in Operating Systems, preliminary programming skills in C/C++ and a broad introduction to various computing courses. Although the students are taught Mathematics for Scientific Computing, Numerical Methods for Scientific Computing I & II along with appropriate hands-on component as a part of their M. Sc. curriculum, they require strong support to sharpen their mathematical skills and motivation for problem solving. The courses in computational Physics/Biology, Scientific Visualization and Parallel Computing lead to a broad training in scientific computing. Several computer science courses like Programming languages, Operating Systems, Networks, Database Systems and Software Engineering are in the course curriculum to serve the broader interests of students inclined towards jobs in industry. The last semester of this M.Sc programme is exclusively devoted for industrial training based on projects.

II. COURSE ON PARALLEL COMPUTING

The one semester course in parallel computing is comprised of four lectures of one hour duration and a hands-on session of three hours per week. Due to the limitations in the background of students, the course gets restricted to the basics of parallel computing and parallel programming using MPI to empower the students towards pursuing HPC.

A. Course contents

The course was based on the text, ‘Introduction to Parallel Computing’ by Ananth Grama and others. The topics covered include parallel computer architecture, parallel algorithm design, embarrassingly parallel computation, task

decomposition and mapping techniques, pipe-lined computation to reduce interaction overheads, MPI calls; point to point and collective communication calls, virtual topology (Cartesian) and group communicators. They are introduced to calculating the cost of computation for serial as well as parallel algorithm apart from the communication overhead for data transfer with different illustrative examples. The parallel performance metric analysis to realize the impact of parallel implementation for different examples forms the concluding part of the course.

B. Hands-on assignments

The typical exercises discussed in the lectures cater to the numerical algorithms in scientific computing field only, for example; trapezoidal rule, quick sort, matrix operations, iterative solver, direct solver for linear systems of equations and Cannon’s multiplication etc. The students are assisted by tutors to motivate the parallel algorithm implementation, debugging and help with the completion of the hands-on exercises. The emphasis on “think parallel” for the exercises are done through black board discussions and handouts. This hand holding methodology in hands-on sessions has ensured the learning outcomes of the course are met effectively to a good extent.

C. Evaluation

We followed a system of continuous evaluation through periodical tests/quizzes, and end semester exam for the theory part. The assessment for the hands-on was based on interaction with students in individual hands-on session and a viva based on a programming assignment.

D. Learnings/Observation

We record some of the observations gathered over several years of offering this course in conjunction with other courses at ISSC. The students who have done courses in Computational Physics/Biology prior to the parallel computing course could appreciate the spirit of the exercises discussed in the parallel computing course owing to a better exposure gained in mathematical modelling in sciences. The applications of linear system of equations, iterative solver and hence iteration based numerical algorithms are

appealing to these students, whereas, others get withdrawn. Further, the algorithmic approach to numerical methods intimidates some of the students who are diffident about their mathematical background, leading to lack of interest in parallel computing in spite of their having better programming skills. For example, we have encountered difficulty in motivating matrix operations despite their understanding of the two dimensional array as data structure for storing matrices. The implementation of the parallel linear iterative solver offered scope for learning data decomposition, task decomposition, inter-process communication (for obtaining local data), synchronization, and data collection inherent to any parallel algorithm, thus an appealing example for motivating the notion of “think parallel”.

In our experience, a composite class of postgraduates, research students, students with industry experience enriched the nature of interactions among peers with different levels of training, which, effectively raises the average reception level of the course material. Concurrent computing paradigms based on OpenMP and CUDA, Hybrid [MPI+X] programming, Python in HPC, cloud computing etc require an additional follow-up course.

E. Challenges

The most challenging part in parallel computing pedagogy has been to motivate the need for parallel computing and the notion of parallel algorithm. By and large the algorithms which the students come across in their prior training are inherently sequential algorithms. The subtlety in realizing that the explicit message passing is needed to access data local to different processes in the single program multiple data (SPMD) approach had to be conveyed through several examples during the course of parallel algorithm design. We face difficulty in emphasizing the importance of floating point arithmetic for numerical computation for students with a computer science background. In our experience, the most difficult task faced by students and teachers is the unravelling of the hidden parallelism in many sequential algorithms and limited realization of parallel computing due to inexperience with real world need of solving larger problems.

III. INTERNSHIPS AT C-DAC, PUNE

A few of the M.Sc (Scientific Computing) students take up one semester long dissertation projects at C-DAC, as part of the industrial training requirement of the programme. They are introduced to the C-DAC’s computing facility, PARAM Yuva II [2] and using the different computing resources, software/ tools, batch queuing system etc on it. The topics related to computer architecture and compiler optimization required for parallel scientific computing are covered through reading assignments to broaden their knowledge base to pursue their project work. Some projects by students are listed below:

- 1) Performance analysis and optimization of parallel Gauss Jacobi method
- 2) Performance analysis of parallel spectral PDE solver
- 3) Performance optimization of stencil computation on multi-core architectures

These students would require further training on parallel numerical algorithms to work with legacy codes in order to contribute to parallel application development.

The students in the pre-final year of B.Tech programme with interest in HPC and background in computing through various courses; DSA/ CO/ Programming/ Scientific Computing/ Compilers etc. spend two months at C-DAC for their summer internship. These students get mentoring in scientific computing and shared memory programming using OpenMP through web tutorials. The primary goal of this programme which is based on mentor-mentee interaction is to orient the students to read research papers, understand, implement and perform some experiments. Some projects by summer interns are listed below:

- 1) Performance optimization of numerical kernels
- 2) High performance matrix-matrix multiplication
- 3) Performance of stencil code on Intel Xeon Phi

This programme has its limitation due to short period of interns’ stay at C-DAC.

IV. CONCLUSION

We believe that the training in parallel computing enables the students enhance their understanding of parallel computer architecture, introduces them to parallel algorithm design, and aids to implement the codes to be run on parallel computing platform. Over the years, we have observed that the hand holding methodology in the formative stages of parallel programming has resulted in better participation of the students leading to the effective delivery of the course. Among other factors that poses serious challenge in motivating the students towards involved activity of parallel application development or parallel computing in general, is the lack of reward owing to extremely limited job opportunities for students with HPC skills in the Indian job market (until recently when data science opened up more avenues).

ACKNOWLEDGMENT

The authors thank the Interdisciplinary School of Scientific Computing for the computational resources for the hands-on assignments in the course and C-DAC for hosting the project students for their internships.

REFERENCES

- [1] ISSC - M.Sc. Syllabus. (2018) M.Sc (Scientific Computing), ISSC, Savitribai Phule Pune University. [Online]. Available: <http://issc.unipune.ac.in/syllabus.php>
- [2] Resources at National PARAM Supercomputing Facility. (2018) PARAM Yuva II at C-DAC, Pune. [Online]. Available: https://www.cdac.in/index.aspx?id=hpc_nsf_npsfres