Augmenting Massive Hands on Labs (MHOL) in Parallel Computing Course

Prathamesh Tugaonkar
Department of Computer Engineering,
Terna Engineering College,
Mumbai India
tprathamesh21@gmail.com

***Abstract*— Teaching parallel computing is always a challenge as it requires certainly a paradigm shift in thinking from sequential learning to parallel programming. Providing plenty of exercises would work as scaffoldings when monitored and instructed by an instructor. Let learners come across the errors and try out various ways of solving the problem. Also, computing requires reasonably good architecture. Investing in online HPC (High Performance Computing) platforms, MOOCs or clusters formation would be of good choice at an initial stage. This paper is the precursor to pedagogy, course outcomes (COs), constructivism, and available architectures that can be readily used for computation with a minimum investment for HPC or Parallel Computing course.**

Keywords- Constructivism; High Performance Computing; Course Design; Blooms' Taxonomy; Parallel Programming, OpenMP, Message Passing, Big Data.

I. INTRODUCTION

Computing challenges from array element addition, matrix multiplication to determining molecular model, body genome map, weather prediction, and fluid pressure dynamics signifies the empirical evolution in High Performance Computing (HPC). This evolution should be anticipated by motivated young minds to engage themselves in parallel computing/programming analysis. Notion of this paper is to design lucrative parallel programming content at graduate level in engineering so that students will start learning parallel codes at an early stage[1]. Teaching this subject in or before fourth semester will help the students to develop parallel belief system for programming despite sequential programming.

Understanding parallel processing is the important factor for advancing computation towards efficiency. To facilitate the learning, this paper talks about the combined approach of Vygotskian constructivism and principles of Bloom's Taxonomy. Constructivism affirms the learning through errors or experience by student in order to understand and remember the concepts. Conceptual understanding can be broaden by providing students with large number of programming exercises be in group or individually[5][6]. This also signifies the need of massive hands on labs where massive is related to providing large number of programming exercises. This massive approach of providing programming exercises will help learner to be open to the exhaustive numbers of errors to trial on[7].

For an effective course design, Blooms hierarchy is globally followed to pen down the course outcomes. Course outcomes for High Performance Computing course are as follows:

CO1: Identify approaches and platforms towards parallel processing for High Performance Computing. (Remember)

CO2: Interpret the analogy that anticipates the parallelism. (Understand)

CO3: Apply the various constructs, directives of the programming languages to infer the efficient parallel code. (Apply)

CO4: Compare the result of Sequential and Parallel code execution and analyze the parameter measures. (Analyze)

CO5: Justify the desired enhancement in the program and evaluate it across various parallel architectures. (Evaluate)

CO6: Modify any case study that encompasses current HPC technologies with respect to acquired knowledge throughout this course. (Creating)

II. LITERATURE REVIEW AND ANLOGIES

Pedagogical approach towards Parallel programming is beneficial when it is taught with human life analogies. Number of exercises with examples will help understanding the parallel concepts [1] e.g. Paper corrections by faculty members. Class rooms can also be made live by introducing games where learner is treated as a processor. Games can illustrate the various shared/parallel memory techniques, race conditions, local memory concepts etc[2].Similar method is found in [3], emphasizing the analogies based model to demonstrate the parallel programming. Another approach of teaching learning process is studied in [4] covering benefits of MOOCs for HPC courses. Authors have identified the various sources for providing Hands on services to the students by Open edX etc. It is also suggested by the authors to follow Backward Designs questionnaire explained by Grant Wiggins. It also talked about the scientific computing and paraphrased the measure metrics like speed up to judge the efficiency in their grid architecture.

III. ARCHITECTURES AT THE INSTITUTES

Certainly, due to involvement of industries such as IBM, Intel, Amazon, NVDIA, HPC is no more a hype but a reality. Students are keen towards understanding the concepts of parallel computing but the problem arises in implementing or in computing stage. It is hardly possible for private

TABLE I. BRIEF COURSE CONTENT

Concept	Relative Examples	Programming Language/ Compilers/Libraries	Identified Architectural Tools
Unlearning the code Serialization	**1. Pictorial View** **2. General aspect of Serialization vs. Parallelization**		• **AWS HPC Clusters** • **MATLAB Distributed Computing Server™** • **Parallel Computing Toolbox™(MathWorks)** • **Access to supercomputer at Colfax (Through Coursera course)**
Dependency	**1. SQL Query** **2. Fibonacci Series**	**C, C++ OpenMP, Intel OpenMP**	
Parallelism (Shared Memory)	**1. Jigsaw Puzzle** **2. Addition of Elements of Array (More than 1000000).**		
Parallelism (Distributed Memory)	**1. Stencil (Edge Detection in an Image)** **2. Simulate Vibrating String using FDM**	**Numactl, Memkind, MPI,mpi4py,FastMPJ, MPJ Express, Apache Hadoop**	

engineering colleges to setup the demanding architecture. I would prefer to demonstrate on the architecture of Colfax. I am also setting up the Beowulf Cluster at my institute. Syllabus designed by University of Mumbai is briefly correlated with the technologies in Table I. Examples mentioned in the table can be assigned and analyzed iteratively to make it massive programming exercise. Case studies related to Financial Computations such as market risk analysis, Computational Fluid Dynamics (CFD) to understand memory utilization, N-Body simulations, Creating DNA sequence by Penn State etc. will help to make convoluted concepts simple. To give deeper understanding of what do supercomputers mean and what is desired configuration for them, to be discussed in the class. SahasraT at IISc which is having Cray owned Aries interconnect is explained followed by terminologies like CPUs, GPUs, cores, hyper threading FLOPS etc. Similarly, Intel Knights Landing architecture at Colfax is explained which uses Omni-Path interconnect fabric. Continuous assessment through Quiz, Group activities in class will facilitate the learning [7].

IV. CONCLUSION

At a beginner's level, this course design would set up a strong foundation for learners for Parallel Computing. This design is primarily the combined approach of Constructivism and Blooms Taxonomy which is cognitive approach towards learning. Plenty of exercises, termed as massive, would keep learners engaged. Monitoring by instructor will help in keeping track of the progress and push the learners if stuck. Analogies will help in understanding the scope of the parallel computing. However, this design will be more fruitful if Quiz, Assignments, Group Projects are assigned to the learners, questions mapped to the COs.

REFERENCES

[1] Steven A. Bogaerts, "One Step at a Time: Parallelism in an Introductory Programming Course", Journal of Parallel and Distributed Computing, 2016

[2] Andrew T. Kitchen and Nan C . Schaller,Game Playing As A Technique For Teaching Parallel Computing Concepts , SIGCSE Bulletin, 1992.

[3] Henry Neeman, et.al. "Analogies for Teaching Parallel Computing to Inexperienced Programmers" SIGCSE Bulletin, 2006

[4] Julia Mullen, "Learning by Doing, High Performance Computing Education in the MOOC Era" Journal of Parallel and Distributed Computing, 2017.

[5] Julia Mullen, "Designing a New High Performance Computing Education Strategy for Professional Scientists and Engineers", IEEE 2016.

[6] B. Neelima, "High Performance Computing education in an Indian engineering institute", Journal of Parallel and Distributed Computing,2017.

[7] Donald Johnson, et.al. "Teaching Parallel Computing to Freshmen", Conference on Parallel Computing for Undergraduates, 1994