Parallel Computing into Programming Languages Courses

Prof Wilson Rivera
University of Puerto Rico at Mayaguez

INTRODUCTION
The Department of Electrical and Computer Engineering at UPRM offers Bachelor of Science and Master of Science programs in Electrical Engineering and Computer Engineering. The Department has about 1,400 undergraduate students and near 100 graduated students. UPRM is a minority serving institution. This paper describes the adoption of the TCPP Curriculum Initiative into a Programming Languages (ICOM 4036) course which is a required course for Computer Engineering undergraduates. The annual enrollment of the course is around 80 students.

TOPICS TO DISCUSS

Parallelism and Memory Models
- Discuss Flynn’s Taxonomy
- Discuss memory access models (shared, distributed, hybrid)
- Show examples of parallel computing environments running on local resources (e.g. a GPU accelerated server; a cluster of servers; and multicore laptops)
- Show the Intel’s “Colony” demo to emphasize the importance of parallel programming

Parallel Programming Notations
- Show examples of OpenMP programming (emphasis on parallel constructs, synchronization mechanisms, and performance considerations)
- Show examples of MPI programming (emphasis on point-to-point communication, collective communication and synchronization)
- Show examples of CUDA programming (matrix-vector multiplication)
- Introduce Erlang programming and show examples compared to MPI
- Introduce TBB and discuss “colony” demo source code

Semantics and Correctness Issues
- Assignment: Implement a multi-access threaded queue with multiple threads inserting and multiple threads extracting from the queue. Use locks to synchronize access to the queue.
- Assignment: Implement a simple loop that calls a function dummy containing a programmable delay. Partition this loop across threads using static, dynamic and guided scheduling.

Graph Algorithms
- Introduce embarrassing parallelism (Ax-y)
- Discuss parallel performance consideration using an OpenMP implementation of mergesort
- Assignment: Implement the Dijkstra’s Single source shortest path algorithm using OpenMP and TBB and discuss related performance issues.

Performance Issues
- Define speedup and scalability concepts
- Introduce Amdahl’s law when discussing OpenMP examples
- Introduce ompP when discussing openMP examples
- Introduce communication overhead when discussing MPI examples
- Introduce IPM when discussing MPI examples
- Discuss Moore’s Law
- Discuss power density considerations

ASSESSMENT

Before this class students had NOT developed parallel codes

After this course students understand programming models

REFERENCES

[1] NSF/TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates

ACKNOWLEDGEMENT

This material is based upon work partially supported by the NSF-TCPP Early Adopter Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.