

Teaching CS2 in Parallel - A Multi-institution Approach to Introducing Parallelism in Data Structures

Akshaye Dhawan ¹ and John P. Dougherty ²

Category of Proposal: core, multi-institutional, liberal arts

List of courses impacted: Data Structures

This proposal examines the integration of the NSF/TCPP Core Curriculum Recommendations in a CS-2 Data Structures course in two different institutions and using two different languages. We seek funding for a teaching collaboration between the PI's to teach Data Structures using thread level parallelism concurrently at Ursinus College (Dhawan) and Haverford College (Dougherty) in the Fall of 2014.

1 Data Structures

We believe that it is important to continue offering an elective that is focused on parallel computation (both institutions currently do this). However, given recent developments in multicore processors, GPU's and clusters it is essential that *every* computer science major have an introduction to the key ideas behind parallel and distributed computing. This can be achieved by introducing these ideas earlier in the curriculum and it is with this intention that we pick Data Structures - a standard second course in the major.

Ursinus offers CS 174 Data Structures - a course covering basic data structures such as stacks, queues, lists, trees, BSTs and graphs. Students enter this course having taken CS 173 Introduction to Computer Science, a basic programming course in Java. Haverford offer CMSC 106 Introduction to Data Structures - a very similar course covering basic data structures but using Python. Students enter this course having taken CMSC 105 Introduction to Computer Science also offered in Python.

This course is a great candidate for introducing the shared memory model of parallel computation and for introducing thread level parallelism. We believe that by focusing on threading as the primary means to expose the student to parallelism, the instructor can teach basic definitions, introduce threads in code and show both the potential speedup and problems such as race conditions and concurrency control. This is essential because a typical CS-II course does not usually allow the time needed to introduce the novice programmer to CUDA, MPI, PVM or Open-MP. We believe that these platforms can be better dealt with in a Parallel Programming Course, however the key concepts behind parallelism can be introduced early in CS-II and built on in subsequent courses.

¹Assistant Professor, Department of Mathematics and Computer Science, Ursinus College

²Associate Professor, Department of Computer Science, Haverford College

2 Collaboration Plan, Evaluation and Dissemination

Collaboration Plan: We have a two semester plan for this collaboration. The Spring of 2014 will be used by both PI's to collaboratively plan and develop materials for the Data Structures course. Our goal is to closely develop assignments and materials that can be used in both courses but using POSIX threads in C++ at Ursinus and using the higher-level *threading* interface in Python at Haverford. These assignments will focus on basics of threading, speedup, independent sub-problems, race conditions and concurrency control. Both PI's are on leave in the Spring and this will allow us to have regular in-person meetings as we work on developing these materials.

In Fall'14 both PI's will concurrently teach Data Structures at their respective institutions. In addition to continuing our meetings, we will visit and guest lecture in each others classrooms. The goal is to create two classrooms where similar concepts are being introduced in two different language paradigms.

Evaluation Our evaluation goals are two-fold: we want to examine the impact of introducing parallelism early in upper-level courses and we want to examine the impact of language on the ease of introducing thread level parallelism in Data Structures. We will develop surveys and assignments that will be given to students in both courses at the beginning of the semester, mid-term and during the final week of class. These will be designed to measure understanding of key parallel programming concepts.

We will also collect data on the performance of these students in the parallel programming elective down the road and compare it to students who have had a sequential Data Structures course.

Finally, Ursinus College will provide a Teaching and Learning Initiative (TLI) consultant who will observe the class on a weekly basis and provide Dhawan with a detailed weekly evaluation as well as administer an independent mid-term evaluation.

Dissemination: In order to share the results of these changes amongst faculty in the liberal arts, we plan to present our specific course changes and evaluation results at the Eastern and North Eastern Conferences for the Consortium for Computing Sciences in Colleges (CCSC) and at the annual conference of the ACM Special Interest Group on Computer Science Education (SIGCSE). We are hoping to organize a workshop or Birds-of-a-Feather session at SIGCSE to open this up to the larger community. Also, Haverford is a part of consortium that includes Swarthmore College and Bryn Mawr College. This allows students at any of these three institutions to take courses at any institution. A similar Data Structures course is offered at the other institutions and we will share our materials and collaborate with the other instructors going forward.

Contributions: A number of early adopters have looked at modifying CS-II and we have already started to look at their approaches and materials in the CDER courseware website. Some of the things that set us apart are the following:

- This proposal is a multi-institutional partnership that will allow us to evaluate different languages in teaching parallelism early.
- Both institutions are small liberal arts colleges in the same geographical area allowing for two very similar classroom profiles. This will allow for a good control on the impact

of the language used in CS-2 on student understanding.

- The pairing an experienced faculty (Dougherty) with a pre-tenure faculty (Dhawan) will allow for an exciting exchange of ideas and mentorship.
- We will share all assignments, syllabi and homework developed for this course on the CDER courseware site.
- We are looking to organize a session at SIGCSE on teaching parallelism early in the curriculum.
- There will be significant regional dissemination of our approach to other liberal arts colleges in the area.